IntechOpen

Multi-Agent Systems
Modeling, Control, Programming, Simulations and Applications

Edited by Faisal Alkhateeb, Eslam Al Maghayreh and Iyad Abu Doush

MULTI-AGENT SYSTEMS - MODELING, CONTROL, PROGRAMMING, SIMULATIONS AND APPLICATIONS

Edited by **Faisal Alkhateeb, Eslam Al Maghayreh** and **Iyad Abu Doush**

Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications
http://dx.doi.org/10.5772/611
Edited by Faisal Alkhateeb, Eslam Al Maghayreh and Iyad Abu Doush

Contributors

Eduardo Aranda-Bricaire, Eduardo Gamaliel Hernandez-Martinez, André Filipe De Moraes Batista, Maria Das Graças Bruno Marietto, Wagner Tanaka Botelho, Terry Lima Ruas, Guiou Kobayashi, Brunno Dos Passos Alves, Sidney De Castro, Norihiro Kamide, María Guijarro, Rubén Fuentes-Fernandez, Gonzalo Pajares, Robson Dos Santos França, Emerson Aguiar Noronha, Alexandre Heideker, Fábio Aragão Da Silva, Takao Tsuji, Takuhei Hashiguchi, Tadahiro Goda, Tsutomu Oyama, Kenji Horiuchi, Takao Shinji, Shinsuke Tsujita, Yasuhiro Kojima, Alan Mutka, Andelko Katalenic, Ivica Draganjac, Stjepan Bogdan, Iskra Antonova, Idilia Batchkova, Emilio Insfran, Luca Cernuzzi, Lorena Rodriguez, Petr Skobelev, Victoriya Repka, Vyacheslav Grebenyuk, Katheryna Kliushnyk, Gottfried Koppensteiner, Munir Merdan, Wilfried Lepuschitz, Thomas Moser, Constantin Reinprecht, Faisal Alkhateeb, Zain Al-Abdeen Al-Fakhry, Eslam Al Maghayreh, Ahmad T. Al-Taani, Shadi Aljawarneh, Michael Kieser, Danielle J. Marceau, Qingshan Li, Lili Guo, Xiangqian Zhai, Baoye Xue, Adil Sayouti, Hicham Medromi, Fouad Moutaouakil, Lin Wang, Zhixin Liu, Masatoshi Sakawa, Hideki Katagiri, Takeshi Matsui, Richard Moussa, Marie Beurton-Aimar, Pascal Desbarats, Dorin Militaru, Mohamad Khaled Allouche, Jean Berger, Atalla F. Sayda, Ricardo Bagnasco, Joan Serrat, Almir dos Santos Albuquerque, Adroaldo GuimarÃ£es Rossetti, Vasco Pinto Da Silva Filho, Rogério Cid Bastos

Notice

First published in Croatia, 2011 by INTECH d.o.o.
eBook (PDF) Published by IN TECH d.o.o.
Place and year of publication of eBook (PDF): Rijeka, 2019.
IntechOpen is the global imprint of IN TECH d.o.o.
Printed in Croatia

Legal deposit, Croatia: National and University Library in Zagreb

Additional hard and PDF copies can be obtained from orders@intechopen.com

Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications
Edited by Faisal Alkhateeb, Eslam Al Maghayreh and Iyad Abu Doush

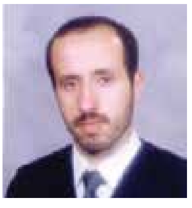
p. cm.

ISBN 978-953-307-174-9

eBook (PDF) ISBN 978-953-51-5991-9

Meet the editors

Faisal Alkhateeb is an assistant professor at the department of computer science at Yarmouk University. He holds a Ph.D. from Grenoble 1 university (2008), M.Sc from Grenoble 1 university (2004), M.Sc from Yarmouk University (2003), and a B.Sc. from Yarmouk University (1999). He published several journal articles and conference papers. He became the chairman of computer science department at Yarmouk University in September 2010.



Dr. Eslam Al Maghayreh is currently an assistant professor at the Computer Science department at Yarmouk University. Dr. Al Maghayreh received a PhD degree in Computer Science from Concordia University (Canada) in 2008, a Master degree in Computer Science from Yarmouk University (Jordan) in 2003, and a Bachelor degree in Computer Science from Yarmouk University in 2001.

Iyad Abu Doush is an assistant professor at the department of computer science at Yarmouk University (YU) in Jordan. He obtained his Ph.D degree in Artificial Intilligence from NMSU (USA). His master degree is in Computer Science from YU. He had B.sc degree in computer science from YU. Dr. Abu Doush has published several journal articles and conference papers and he is a reviewer for several international conferences and journals.

Contents

Preface

A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents. Multi-agent systems can be used to solve problems which are difficult or impossible for an individual agent or monolithic system to solve. Agent systems are open and extensible systems that allow for the deployment of autonomous and proactive software components. Multi-agent systems have been brought up and used in several application domains. This book is a collection 24 excellent works on multi-agent systems divided into four sections: Multi-Agent Systems Modeling; Control, Negotiation, Reasoning, Tracking and Networking on Agents Environments; Multi-Agent Systems Programming and Multi-Agent Systems Simulation and Applications.

Faisal Alkhateeb, Eslam Al Maghayreh and Iyad Abu Doush,
Yarmouk University,
Jordan

Part 1

Multi-Agent Systems Modeling

1

Requirements Modeling for Multi-Agent Systems

Lorena Rodriguez[1], Emilio Insfran[1] and Luca Cernuzzi[2]
[1]ISSI Research Group, Department of Information Systems and Computation, Universidad Politécnica de Valencia, Camino de Vera s/n 46022, Valencia,
[2]DEI, Universidad Católica "Nuestra Señora de la Asunción", Campus Universitario, C.C. 1683, Asunción
[1]Spain
[2]Paraguay

1. Introduction

The inadequate management of system requirements is a major cause of problems in software development (Anton, 2006). Requirements Engineering is a branch of Software Engineering and includes a set of activities concerning the identification, specification, validation, and management of system requirements. However, a traditional approach to requirements engineering may not be fully effective for certain complex systems that require high levels or specific kinds of abstractions, and the need to define more specific requirements engineering processes for these types of systems thus arises.

A Multi-Agent System (MAS) is a specific type of system that is composed of multiple intelligent agents that interact with each other to achieve certain objectives. These systems can be used to solve problems that it is difficult or impossible for a monolithic or a single agent system to resolve. In recent years, various methodologies have been proposed to guide the development of multi-agent systems, such as Tropos (Giorgini et al., 2005), Ingenias (Gómez-Sanz & Pavón, 2003), Gaia (Zambonelli et al., 2003), etc. However, despite the importance of the requirements phase in the development of software systems, many of the proposed methodologies for the development of MAS do not adequately cover the requirements engineering phase (Cernuzzi et al., 2005), focusing mainly on the design and implementation phases. Moreover, a recent study on the application of requirements engineering techniques in the development of a multi-agent system (Blanes et al. a, 2009) found that 79% of the current methodologies for MAS development use requirements engineering techniques which have been adapted from other paradigms (object orientation, knowledge engineering, etc.) (Anton, 2006). However, these techniques and notations may not be sufficient to cover the nature of MAS, since these systems, along with their organizational, structural, or functional properties, characteristics that are not normally necessary in conventional software systems such as *pro-activity, adaptability, collaboration, truth, or disposition* (Blanes et al. a, 2009). These characteristics are denominated as *social behavior* (Hector & Lakshmi, 2005). Therefore, there is a need for new methods and techniques that enable the appropriate acquisition and treatment of MAS requirements.

(Blanes et al. b, 2009) and (Rodriguez et al., 2009) present two proposals for the acquisition and modeling of requirements for the Gaia (Zambonelli et al., 2003) methodology which covers the analysis and design phase of MAS. Based on the experience using these approaches, this chapter presents an evolution of these earlier proposals to support the acquisition and modeling of requirements regardless of the analysis and design methodology used, and covers four essential perspectives of a MAS: *organizational*, *structural*, *functional*, and *social behavior*.

It is also worth mentioning that agent technology is useful in many complex domains: e-commerce, health, stock market, manufacturing, games, etc. In particular, we are interested in the game development domain since it comprises a set of characteristics such as collaboration, negotiation, trust, reputation, etc., which specially can be dealed with a MAS. According to Google Trends and the ESA annual report (ESA, 2010), games development is one of the business markets that has undergone most growth in the last few years. In addition, the agent-oriented paradigm is one of the most promising for modeling such business market due to the social behavior characteristics (negotiation, cooperation, etc.) of the agents and the complexity that MASs can support. For this reason, in the next chapter, we illustrate the feasibility of our approach by applying the requirements modeling process to the development of the strategic board *Diplomacy Game* (Fabregues et al., 2010).

The structure of this chapter is as follows. Section 2, discusses the related works. Section 3, describes the organizational, structural, functional and social behavior perspectives that were considered when modeling MAS. Section 4, presents the requirements modeling proposal. Section 5, focuses on the case study of the *Diplomacy Game*. Finally, section 6, concludes and introduces future research work.

2. Related work

The importance of the requirements phase in software development is widely known. However, capturing and modeling the requirements of a system is not a trivial task. In particular, MAS require abstractions, techniques, and notations which have been specifically tailored to this domain. We propose four basic perspectives for the modeling of MAS requirements: *organizational*, *structural*, *functional,* and *social behavior.* This section, presents some proposals for the acquisition and modeling of requirements that cover these four perspectives of a MAS.

The organizational perspective is supported in proposals such as GBRAM (Anton, 2006). GBRAM is a relevant traditional goal-oriented requirements engineering proposal. It provides a procedural guide for the identification and development of goals and introduces techniques that assist in their methodical and systematic analysis. GBRAM has a great deficiency in terms of formality. This includes the lack of models, formal notations and tools that support the modeling that the method uses. Nevertheless, the guidelines and the level of clarity it offers are very good. Moreover, GBRAM also emphasize the verification of the requirements through its refinement stage which specifies certain guidelines to follow, thus making this process more reliable. Therefore it is possible to track the requirements captured, and this is reflected in the traceability offered by the method.

Another proposal for requirements modeling that supports the organizational perspective is the i * framework (Yu, 1997). This framework has been established as the basis for the Tropos methodology (Giorgini et al., 2005). Tropos has been appropriately adapted to the acquisition and modeling of the actors in the system and its environment, i.e., the actors, goals, tasks, interactions, dependencies, resources needed, etc. However, it does not permit a full representation of constraints nor does it propose a modeling environment. Since we consider goal orientation to be of particular interest in the capturing of requirements for MAS, we believe that it is necessary to analyze other methods which are complementary to this approach.

The structural perspective is supported by proposals such as AUML (Odell et al., 2000). AUML tends to be asserted as a notational standard in various methodologies; one of the most common proposals for the requirements phase is the adoption of Use Case diagrams. This formalism has shown good results for the representation of functional requirements and is also a good tool for communication with stakeholders. Nevertheless, Use Cases have limitations in capturing qualitative aspects of the environment and interactions with it. In addition, a interesting contribution of AUML is the Agents Interaction Protocol (AIP), which constitutes a central aspect for MAS, specified by means of protocol diagrams.

Another proposal that covers the structural perspective is KAOS (Van Lamsweerde et al., 1998), a proposal for modeling requirements through goals. KAOS consists of a specification language, a method of elaboration, and the meta-level knowledge which is used as a guide. A KAOS model contains goals, information system requirements, expectations about the system environment, conflicts between goals, obstacles, entities, agents, etc. One of the strengths of the proposal is that of its use of formality to achieve correction. Moreover, the idea of constraint is useful in identifying some of the external problems of integrity, and this contributes to the robustness of the system. However, the successful implementation of the method depends heavily on the developer's experience in the domain and how well defined the problem to be solved is (Huzam & Maibaum, 2006).

Other proposals do not support the organizational and structural perspective. This is the case of CREWS (Maiden, 1998), which focuses on the perspectives of functional and social behavior. CREWS is based on system object models that are abstractions of the key features of the different qualities of the problem domain. CREWS uses these models to generate normal course scenarios, and it then uses the theoretical and empirical research into cognitive science, human-computer interaction, collaboration systems and software engineering as a basis to generate alternative courses for these scenarios. A potential weakness of the CREWS approach is that the generation of scenarios is domain-oriented, in contrast with the goal-oriented scenario analysis and the task-oriented Use Case modeling. If the scenarios are intended to validate the requirements, these requirements should be oriented towards the generation of scenarios.

In summary, the organizational perspective is covered by proposals such as GBRAM and i *, and the structural perspective is covered in proposals such as KAOS and AUML. Most of the proposals presented in some way cover, either totally or partially, the functional and social behavior perspective, as in the case of CREWS. However, to the best of our knowledge no methods that completely cover all four perspectives needed for the development of a MAS exist.

3. Different perspectives for multi-agent systems

This work aims to provide a solution to the lack of RE modeling approaches that appropriately cover the four perspectives of MASs: *organizational*, *structural*, *functional*, and *social behavior*. In order to contextualize these perspectives, an overview of them is presented which emphasizes both social behavior and organizational aspects, since these are key aspects for the development of MASs.

3.1 Organizational perspective

In the organizational perspective, the organization is represented as a system that has certain goals. The organization attains these goals through consistent actions, which use system resources and alter the desired system state (Falkenberg, 1998). Some authors such as (Zambonelli et al., 2003) consider that the human organizational metaphor is very adequate for systems which are situated in open and changing environments. They define a MAS as a software system that is conceived as the computational instantiation of a group of interacting and autonomous individuals (agents). Each agent can be seen as playing one or more specific roles: it has a set of responsibilities or goals in the context of the overall system and is responsible for pursuing these autonomously. Furthermore, interactions are clearly identified and localized in the definition of the role itself, and they help to characterize the overall structure of the organization and the agent's position in it. The evolution of the activities in the organization, which is derived from the agents' autonomous execution and from their interactions, determines the achievement of the application goal.

3.2 Structural perspective

The structural perspective shows the system architecture in terms of entities and the static relationship between them. The modeling of these entities and relationships provides an abstract structural perspective of the system. We believe that this perspective is necessary to identify the entities that will be needed to build the future MAS. If the static and structural relationships are to be captured accurately, the development method must include formalisms and techniques to specify relationships of hierarchy (inheritance), semantic dependency (association) and part-of relations (aggregation).

3.3 Functional perspective

The functional perspective shows the semantics associated with the organizational roles' services that are motivated by the occurrence of events. In this context, we understand an organizational role to be the representation of an abstract entity that provides (multiple) system methods or services. An event is something that occurs in the environment and to which the organizational role reacts by running a method or service. This perspective focuses to model the functional requirements to be met by the roles in the future MAS.

3.4 Social behavior perspective

The social behavior perspective shows the possible sequences of events or services to which an agent can respond or that occur in its lifetime, along with interaction aspects such as communication between agents, and this is often represented as state or activity diagrams. As is discussed above, in addition to organizational, structural, and functional properties, a MAS also requires characteristics that are not normally required in conventional software systems, such as *pro-activity, adaptability, collaboration, truth, or disposition*. These characteristics are denominated as *social behavior*. We therefore believe that covering this perspective in a proposal for modeling requirements for MAS is an important contribution towards the development of such systems, since the essence of these systems is the performance of complex tasks that other types of systems are not capable of solving.

3.4.1 Classification

In order to properly structure and organize the features of social behavior requirements, we briefly present the classification scheme of agent characteristics defined in (Hector & Lakshmi, 2005). According to the authors, three main attributes of an agent are defined: (a) *autonomy*, which refers to the fact that an agent should run independently, with little or no human intervention, (b) *temporal continuity*, which signifies that an agent should run continuously rather than simply perform a task and finish, and (c) *social skills*, which signifies that an agent should possess some form of social skills, since the agent's advantages lie in its interactive communication with other agents. In addition to these core attributes, an agent can also be classified according to the following *social behavior* characteristics:

a. *Pro-activeness*: this refers to how the agent reacts to -and reasons about - its environment, and how it pursues its goals. The agent can directly react to stimuli in its environment by mapping an input from its sensors directly to an action, or it can take a purely planning, or goal-oriented, approach to achieve its goals. This last approach relies upon utilizing planning techniques.
b. *Adaptability*: this describes an agent's ability to modify its behavior over time. In fact, the term "agent" is often taken to implicitly mean "intelligent agents", which combine traditional artificial intelligence techniques to assist in the process of autonomously performing tasks. This feature includes other sub-features such as learning and sub-submission.
c. *Mobility*: this refers to the agents' capability of transporting their execution between machines on a network. This form of moving can be physical, where the agent travels between machines on a network, or logical, where an agent which is running on a single machine is remotely accessed from other locations on the Internet.
d. *Collaboration*: collaboration among agents underpins the success of an operation or action in a timely manner. This can be achieved by being able to coordinate with other agents by sending and receiving messages using some form of agent communication language, and permits a high degree of collaboration, thus making social activities such as distributed problem solving and negotiation possible. Moreover, it is possible for agents to collaborate without actual communication taking place. The interaction of agents with resources and their environment may lead to the emergence of collaborative or competitive behavior.
e. *Veracity*: this refers to the agent's ability to deceive other agents via their messages or behavior. An agent can thus be truthful in failing to intentionally deceive other players. Moreover, an agent that is untruthful may try to deceive other agents, either by providing false information or by acting in a misleading way.
f. *Disposition*: this refers to the agent's "attitude" towards other agents, and its willingness to cooperate with them. An agent may always attempt to perform a task when asked to do so (benevolent), or may act in its own interests to collaborate with other agents only when it is convenient to do (self-interested), or it might try to harm other agents or destroy them in some way (malevolent).

The above characteristics in the classification represent to some extent abstraction of human social behavior, and are those that differentiate agent paradigms from traditional software development. In this work, we use this classification to study the characteristics of social behavior and to propose mechanisms for the definition and specification of requirements of these types. In particular, and as a starting point, in this work we will focus on the following characteristics: proactiveness, collaboration, veracity, and disposition. Other characteristics such as adaptability or mobility will be considered in future work.

Social behavior is a skill that must have an agent in a MAS. Moreover, if we consider the organizational metaphor, an agent can, at different times in its life-cycle, play one or more specific roles, which in turn have a set of responsibilities and goals. We therefore propose to identify these features of social behavior in the requirements modeling process at role level, through an analysis of the goals that need to be attained. Therefore, in the later phases of the software development, when an agent has to be defined, the corresponding roles of which a given agent will be composed will determine the agent's complete social behavior.

4. Modeling requirements for multi-agent systems

To support the *organizational*, *structural*, *functional* and *social behavior* perspectives, we propose a requirements modeling process which is decomposed into two main activities: *Requirements Definition* and *Requirements Specification*. The user's specific needs are identified in the *Requirements Definition* activity. In particular it is identified: the organizational structure of the system; the roles that are required in each sub-organization; the roles goals; the social behavior needed for the roles to carry out their goals; and relevant entities of the environment. The detailed requirements for developers are specified in the *Requirements Specification* activity. The specifications extracted from the *Requirements Definition* activity are refined, and the level of detail increased, in order to identify artifacts which are closer to the analysis and development of the system: activities and interactions, resources of the system, the permissions that roles have in those resources and organizational rules.

Moreover, the process is based on the definition of models needed to describe the problem in more concrete aspects that form the different perspectives of the system. In particular, in the *Requirements Definition* activity it is possible to identify: (a) a *Refinement Tree,* which identify and represent the goals of the system and their hierarchy, the roles that will carry out these goals in an organizational context, and the organizational structure of the system, and (b) a *Domain Model* with which to represent the entities that could be used as the organization's resources. In turn, the *Refinement Tree,* as is specified by the above description, represents: (i) *Mission Statement*, which is the main objective that the system under development provides the environment with in order to identify the overall goal within the organization as a whole; (ii) *Organizational Model*, to represent the sub-organizations of which the global organization is composed; (iii) *Role Model*, to represent the roles involved in each sub-organization, the inheritance relationships between them, and the social behavior needed between roles to accomplish their goals; and (iv) *Goal Model*, to represent the goals associated with each role.

Moreover, in the *Requirements Specification* activity it is possible to identify: (c) a *Behavior Model*, to represent the decomposition of goals into tasks and protocols in order to understand the internal flow of a role to determine its responsibilities, (d) an *Environment Model*, to represent the permissions of the roles identified in the *Role Model* with regard to the resources of the *Domain Model*; and (e) an *Organizational Rules Model*, to represent the constraints of the organization's behavior.

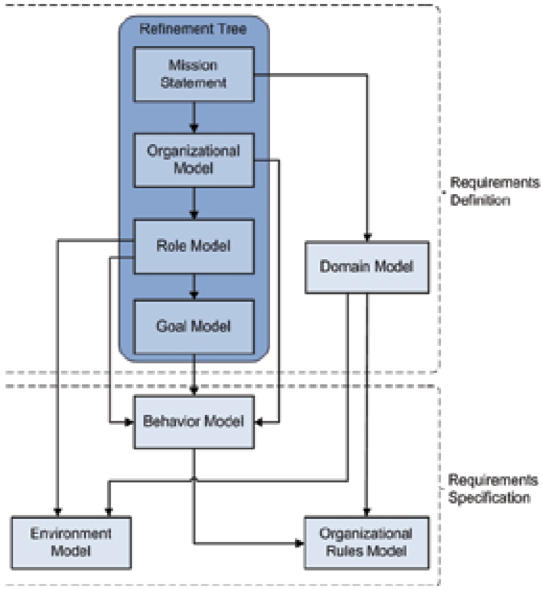


Fig. 1. Models of the proposal and its relationship

In order to obtain a clear view of the models used, each of them is presented as follows. The *Mission Statement* is defined in natural language, with a recommended extension of one or two paragraphs. Since the *Mission Statement* identifies the overall goal within the organization as a whole, it provides us with information about the organizational and functional perspectives. The *Mission Statement* is the root of the *Refinement Tree*. It is successively refined to identify the goals of the system to be represented as leaf nodes in the tree. It is possible to distinguish three general levels in this process: (i) we first define the decomposition of the system in a hierarchy of sub-organizations, thus representing the *Organizational Model*. A sub-organization is a part of the system that aims to achieve a goal in the system and weakly interacts with other parts of the system (low coupling); (ii) we then decompose the sub-organizations into roles that partially represent the *Role Model*. A role is the representation of an abstract entity that has (multiple) system goals; (iii) and finally, we identify the goals of the system and a hierarchy of them, thus representing the *Goal Model*. The goals are tasks which are carried out by a role in the sub-organization. The structure of the *Refinement Tree* allows us to identify elements of the organizational perspective through the decomposition of the system into sub-organizations; elements of the structural perspective by identifying the roles that make up the sub-organizations and finally aspects of the functional perspective by identifying the goals that each role has to perform. As was previously mentioned, the *Role Model* describes the roles that belong to the sub-organizations of the *Refinement Tree*. The purpose of this model is to represent the different roles found in each sub-organization and to reason about their special relationships. The special relationships between roles can serve to identify the common properties between the roles in order to create a hierarchy of roles using inheritance relationships and the identification of the social behavior relationships between roles in different sub-organizations. The resulting *Role Model* comprises the information represented in the *Refinement Tree*: one diagram for the inheritance relations between roles and one or more diagrams as needed for each sub-organization for the social behavior relations between roles. The UML Use Case Diagram is used to represent this information and complement the *Refinement Tree* representation of roles. The roles are represented as actors which are labeled with the stereotype <<role>>.In addition, the inheritance relations are represented with the corresponding diagram relation, and the social behavior relations are represented as relations labeled with the stereotypes <<collaboration>>, <<disposition>> and <<veracity>>. We propose naming the relations with the corresponding property (i.e. for the social behavior relation *collaboration* the relation is named as "communicative", "non-communicative" or both, for the social behavior relation *disposition* the relation is named as "benevolent", "self-interested", "malevolent" or the combinations, and finally for the social behavior relation *veracity* the relation is named as "truthful", "untruthful" or both). A social behavior relation between two roles could be of one or more property, since the relation is dynamic, i.e. it may alter depending on the agent that will eventually play the role. This information allows us to express elements of the structural, organizational and social behavior perspectives.

The *Domain Model* represents the entities identified in the problem domain. The purpose of this is to identify key concepts and relationships, thus representing a first structural view. These entities are seen from the point of view of the application domain, and implementation details are therefore avoided at this level. Associations and inheritance relationships between domain entities are also represented. The identification of these domain entities and their relationships allows us to extract information for the structural perspective and to partially extract information for the organizational perspective. The UML Class Diagram will be used to represent this information.

The *Behavior Model* shows a sequence of steps that represent the flow of activities needed to achieve the goals identified in the system. A representation of the flow of tasks could be useful: to understand the logical flow of a role; to complement the information regarding social behavior identified in the *Role Model;* and to help to identify new information when one role needs to work with others in order to accomplish a task. The identification of the flow of activities allows us to extract information for the functional perspective. Furthermore, the identification of interactions between different roles allows us to identify information for the social behavior perspective. The UML Activity Diagram will be used to represent this information.

The *Environment Model* represents the permissions of the roles with regard to the entities identified in the *Domain Model*. For each role identified in the *Role Model*, resources are established for those who can legitimately access them. Finally the permissions (perceive or modify) are established. The identification of these permissions offers information of the structural and functional perspectives of the system. The UML Use Case Diagram is used to represent this information, and the roles are represented as actors which are labeled with the stereotype <<role>>, the resources are represented as classes and the permissions are represented as relations between the role and the entity, which are labeled with the stereotypes <<perceive>>, and <<modify>.

The *Organizational Rules Model* identifies and represents the general rules concerning the organization's behavior. These rules can be viewed as general rules, responsibilities, restrictions, the desired behavior, and the sequence or order in such conduct. These rules will be represented by building on GBRAM, in which two types of dependency relationships between goals are distinguished: precedence and restriction, which are represented by the symbols **<** and **→** respectively, and by adding a relationship to the proposal to represent general rules of the system, which is represented with only natural language. This information contributes to extract information for organizational, structural and functional perspectives of the system. We suggest that the set of rules should be represented with a table schema in which each rule is defined by a natural language description of the relationship, the type and the corresponding formula if necessary.

Each of these models provides the information which is necessary to cover the different perspectives of a MAS: (i) structural (*Domain Model, Role Model,* and *Environment Model*); (ii) organizational (*Mission Statement*, *Organizational Model*, *Roles Model, Domain Model,* and *Organizational Rules Model*); (iii) functional (*Mission Statement, Goal Model, Behavior Model, Environment Model* and *organizational Rules Model*); and (iv) social behavior (*Role Model* and *Behavior Model*). Figure 1 shows an overview of these models and their respective relations.

The process for defining and specifying the requirements is described in the following subsection.

4.1 Requirements modeling process

As was mentioned earlier, the requirements modeling process proposed involves two phases: *Requirements Definition* and *Requirements Specification*. Figure 2 shows an overview of this process, using the SPEM graphical notation (OMG, 2010). Each activity of the process produces a document that is composed of the sum of all the models and documents of the working definition that is included in each activity. The *Requirements Definition* activity tasks are performed first, thus producing the requirements specification. The *Requirements Specification* activity tasks are then performed, using the requirements specification produced in the previous activity as input and resulting in the production of the refined requirements specification. At this point the *Requirements Definition* activity can again be performed in case some kind of inconsistency or incompleteness is encountered in the specification, or the process may end.

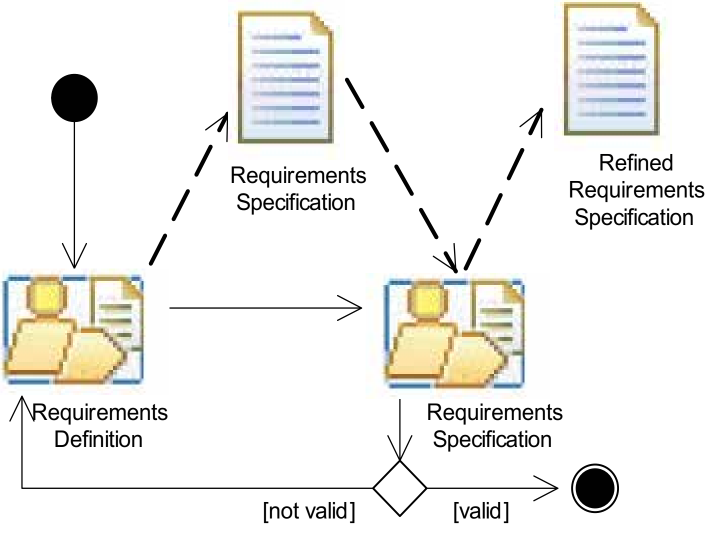


Fig. 2. Requirements modeling process overview

4.1.1 Requirements definition

The *Requirements Definition* activity consists of three tasks whose aim is to identify the models of the phase, as is shown in Figure 3. The first task is to *Create Refinement Tree*, beginning with the definition of the *Mission Statement* which is then broken into sub-organizations, roles and goals. This information is part of the *Mission Statement*, *Organizational Model*, *Role Model* and *Goal Model*. The list of roles identified in the previous task will be used as input for the next task: *Refine Roles*. Here we discuss possible structural similarities in order to identify inheritance relationships, and we analyze the goals to be attained by each role in each sub-organization in order to identify the social behavior relationships between them. If deemed appropriate, it is possible to return to the previous task in order to update the *Refinement Tree*, or the next task can be performed. In the last task, *Identified Entities*, the *Domain Model* is constructed from the identified entities, and association and inheritance relationships among them are defined.

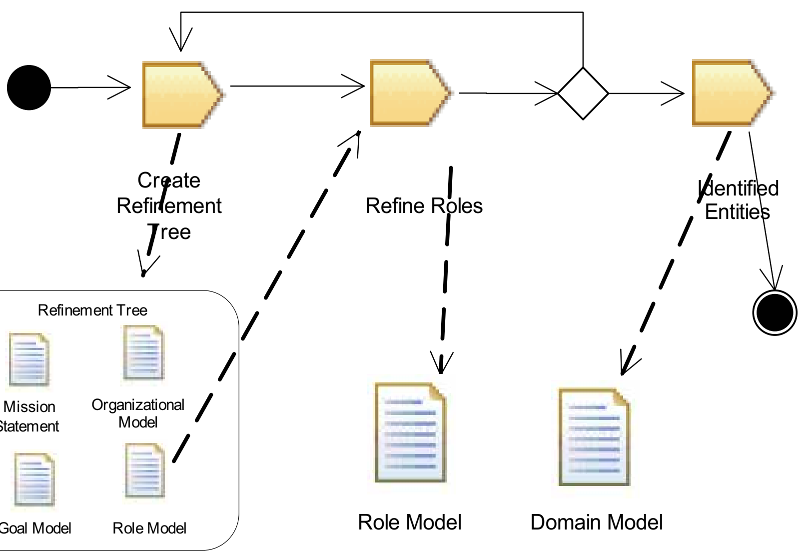


Fig. 3. Requirements Definition activity decomposed into tasks and artifacts

4.1.2 Requirements specification

The *Requirements Specification* activity involves the creation of three models: *Behavior Model*, *Environment Model* and *Organizational Rules Model*, and therefore consists of three tasks for the creation of the models, as is shown in Figure 4. The first task in this activity is *Create Activity Diagrams*. The *Organizational Model*, *Role Model*, *Goal Model* of the *Requirements Definition* activity are used as input. The necessary Activity diagrams are created as a result of this. When this has been completed, the next task is performed: *Develop Environment Model*. The *Role Model* and the *Domain Model* of the *Requirements Definition* phase are taken as input. Then, the *Define Organizational Rules* task is performed, taking as input the *Role Model* of the *Requirements Definition* activity and the *Environment Model* of the current activity. The *Organizational Rules Model* is produced as a result of this.

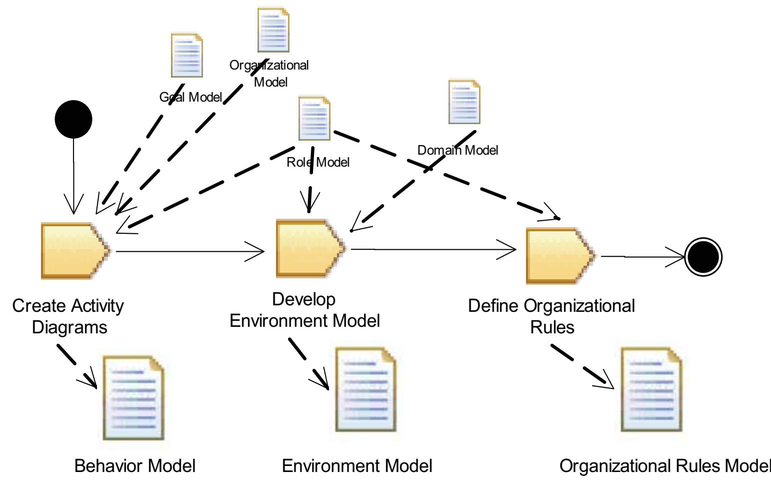


Fig. 4. Requirements Specification activity decomposed into tasks and artifacts

Finally, the artifacts generated during the process can relate to analysis and design artifacts from other methodologies by establishing a traceability framework. This will increase the overall quality of the system to be.

5. Case study: diplomacy game with agents

We have used the *Diplomacy Game* to verify the feasibility of our approach in areas such as negotiation, argumentation, trust and reputation (Fabregues et al., 2010) in the game development domain. Many interesting features make the *Diplomacy Game* compelling for the applying the agent technology: the absence of random movements, all players move their units simultaneously, all units are equally strong so when one attacks another the winner of the battle is decided by considering solely the number of units helping one another, etc. Accordingly, from a player's point of view, the most important feature of the game is the negotiation process: deciding allies, selecting who to ask for help, arguing with other players to obtain information about their objectives or to discover what they know, and so on. We have used the rulebook of the *Diplomacy Game* (Wizards, 2010) as a description of the system to be modeled with the process proposed in this work. The most relevant aspects of the game are provided as follows.

The *Diplomacy Game* is played by seven players and a Game Master. Each player represents one of the seven "Great Powers of Europe" in the years prior to World War I. These Great Powers consist of England, Germany, Russia, Turkey, Italy, France, and Austria. At the start of the game, the players randomly decide which Great Power each will represent. This is the only element of chance in the game. As soon as one Great Power controls 18 supply centers, it is considered to have gained control of Europe. The player representing that Great Power is the winner.

Diplomacy is a game of negotiations, alliances, promises kept, and promises broken. In order to survive, a player needs help from others. In order to win the game, a player must eventually stand alone. Knowing who to trust, when to trust them, what to promise, and when to promise it is the heart of the game.

At the beginning of each turn, the players meet together in small groups to discuss their plans and suggest strategies. Alliances between players are made openly or secretly, and orders are

coordinated. Immediately following this period of "diplomacy," each player secretly writes an order for each of his or her units on a slip of paper. When all the players have written their orders, the orders are simultaneously revealed, and are then all resolved. Some units are moved, some have to retreat, and some are removed. Resolving orders is the most challenging part of the rules and requires complete knowledge of the rules. Each turn represents six months of time. The first turn is called a Spring turn and the next a Fall turn. After each Fall turn, each Great Power must reconcile the number of units it controls with the number of supply centers it controls. At this time some units are removed and new ones are built. The purpose of the Game Master is to keep time for the negotiation sessions, collect and read orders, resolve issues, and make rulings when necessary. This role should be strictly neutral.
Each turn has a series of phases: (a) Spring four-phase turn: Diplomatic phase, Order Writing phase, Order Resolution phase, Retreat and Disbanding phase; (b) Fall five-phase turn: Diplomatic phase, Order Writing phase, Order Resolution phase, Retreat and Disbanding phase, Gaining and Losing Units phase. After a Fall turn, if one Great Power controls 18 or more supply centers, the game ends and that player is declared the winner.
Based on the tasks of the *Requirements Definition* and *Requirements Specification* activities proposed, and which were presented in the previous section, the development of the case study is presented below.

5.1 Requirements definition

The requirements modeling process starts with the *Requirements Definition* activity. This activity starts with the first task: *Create Refinement Tree*. First the *Mission Statement* of the system must be defined, which in this case is simple and is the *Management of the Diplomacy Game*. For the definition of the sub-organizations of the system we decided that the problem naturally leads to a conception of the whole system as a number of different MAS sub-organizations, one for each phase of the game, and one extra sub-organization representing the start of the game. The resulting sub-organizations are: *Initial phase*, *Diplomatic phase*, *Writing Order Phase*, *Order Resolution phase*, *Retreat and Disband phase* and *Gaining and Losing Units phase*. This concept of representing the sub-organizations of the system as phases was also used in (Zambonelli et al., 2003). The roles that are part of each sub-organization are then defined, resulting in three roles: *Great Power*, *Game Master* and *Unit* which, depending on which sub-organization they are, have different goals. Finally the roles are refined with the goals they need to attain in order to fulfill each sub-organization's objective. For example in the *Order Resolution Phase* sub-organization, the *Game Master* role has the goal of *Resolve Order Conflicts* and the *Unit* role has the goal of *Follow Orders*. Figure 5 shows the complete resulting R*efinement Tree*.
The second task, *Refine Role Model,* is performed to complete the *Role Model* based on the information defined in the *Refinement Tree*. Possible inheritance relationships between roles can be specified in this task. The goals of each role in each sub-organization are also reviewed in order to identify whether the role needs social behavior relationships in any sub-organization. Owing to space limitations we shall only illustrate the *Role Model* showing one of the most significant diagrams (see Figure 6). However, the same analysis should be performed for all of the sub-organizations, thus resulting in one or more Social behavior diagrams for each sub-organization. Upon analyzing the goals of the roles of the *Diplomatic Phase* sub-organization, we identified that the *Great Power* role needs to have the collaborative relation to attain all of its goals in the sub-organization analyzed, and more specifically, the role needs to be communicative with other instances of the *Great Power* role

and with the *Game Master* role. The same applies in the case of the *Game Master* role fulfilling its *Control Negotiation Session* goal: the collaborative relationship will be with the *Great Power* role. The collaborative relationship between *Great Power* and *Game Master* will therefore be on both sides, represented with a non-directional arrow. Moreover, if the *Great Power* role is to fulfill all of its goals in the sub-organization analyzed, it needs to be benevolent, self-interested or malevolent with regard to another instance of the *Great Power* role, depending on the agent's intentions. In this sub-organization, negotiation, persuasion and trust are keys to the *Great Power* role. On the other hand, the *Great Power* role in the sub-organization analyzed is in all cases benevolent with regard to the *Game Master* role, and vice versa. Finally, we believe that it is necessary for the veracity relation between the *Great Power* role and other instance of the same role to be truthful or untruthful, again depending on the intentions of the agent playing the role. We also believe that it is necessary for the veracity relation between the *Great Power* role and the *Game Master* role to be truthful in both directions.

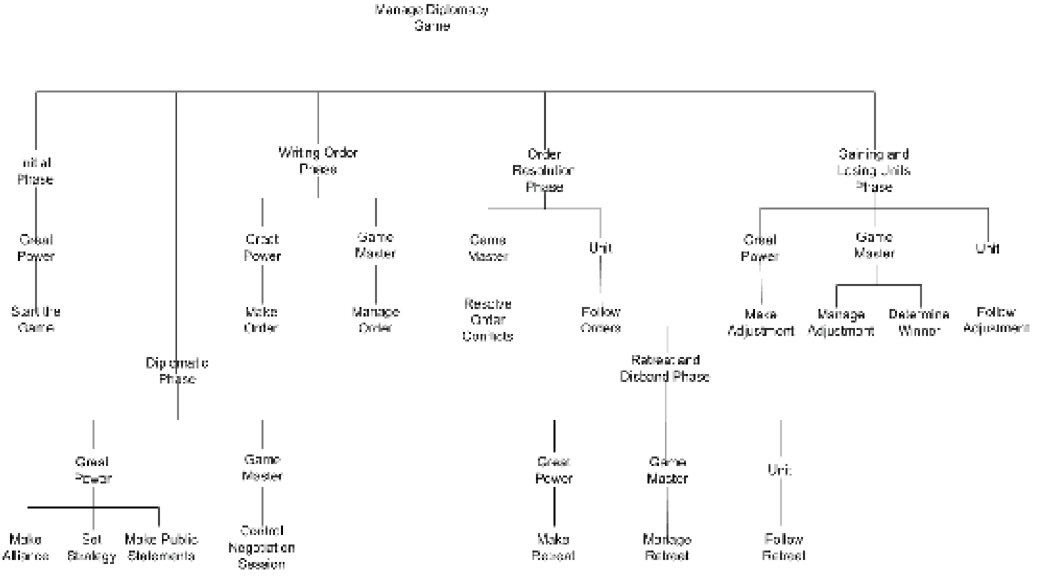


Fig. 5. Diplomacy game Refinement Tree

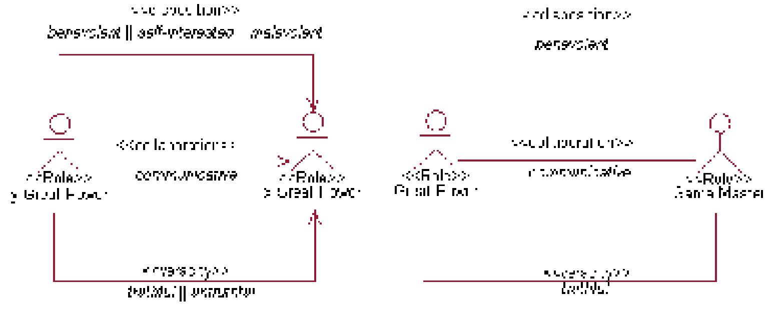


Fig. 6. Social behavior diagram showing relations between roles in the *Diplomatic phase* sub-organization (collaboration, veracity, and disposition)

The third task, *Identify entities,* is performed to define the *Domain Model.* Figure 7 shows the *Domain Model* generated. Briefly, the domain consists of a *Map* that is composed of many *Countries* which in turn have *Boundaries* and *Provinces*. A *Province* can be an *Inland*, *Coastal* or *Water* province. A *Supply Center* is in a *Province*, but a *Province* may or may not have a *Supply Center.* Furthermore, a *Unit* is in a *Province*, but a *Province* may or may not have a *Unit.* A *Unit* can be an *Army* or a *Fleet*. Both a *Province* and a *Unit* belong to a *Great Power* which in turn is a *Country*, but not all *Countries* are *Great Powers.* A *Great Power* has many *Documents*, *Orders*, *Retreats* and *Adjustments*, and they all belong to only one *Great Power*. *Orders*, *Retreats* and *Adjustments* are all for one *Unit* and a *Unit* follows many *Orders*, *Retreats* and *Adjustments.*

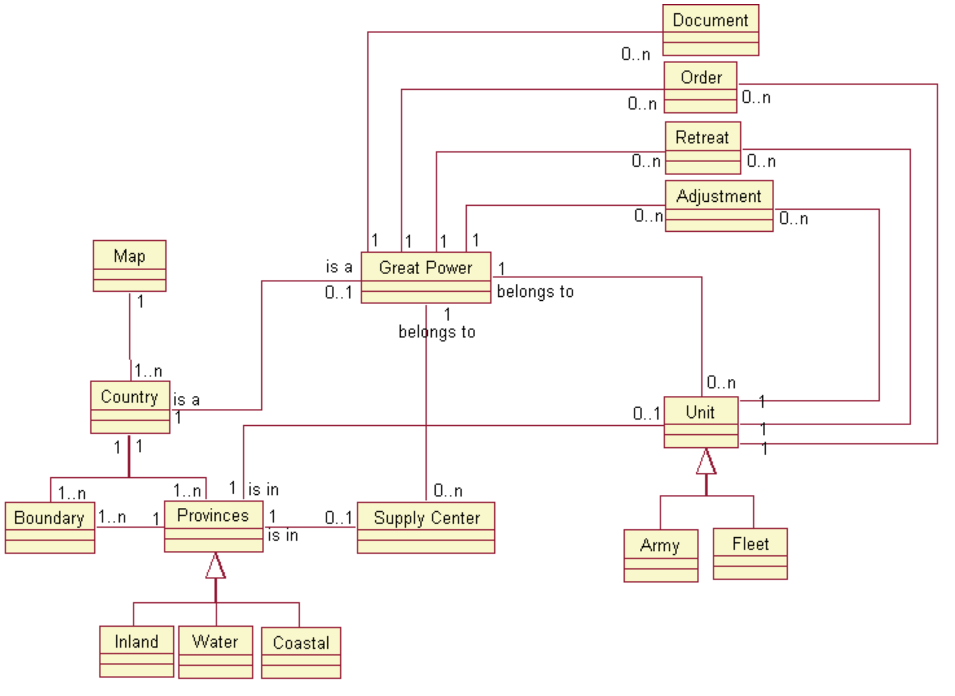


Fig. 7. Diplomacy game Domain Model

As a result of performing the *Requirements Definition* activity we obtain the *Refinements Tree* shown in Figure 5, which represents the *Mission Statement*, *Organizational Model*, partially the *Role Model* and *Goal Model*. One or more Social behavior diagrams is needed for each sub-organization, thus complementing the information of the roles in the *Refinement Tree* to complete the *Role Model*. An example of this is shown in Figure 6. Finally, the UML Class Diagram which relates the entities identified in the domain to represent the *Domain Model* are shown in Figure 7.

5.2 Requirements specification

The second activity is to perform the *Requirements Specification,* which starts with the first task, *Create Activity Diagrams*, in order to specify the *Behavior Model* using the information from the *Organizational Model*, *Role Model,* and *Goal Model* generated in the *Requirements Definition* activity as input. Once again, owing to space limitations we shall only illustrate one Activity diagram from the *Behavior Model* (see Figure 8). However, the same analysis must be carried out for each goal identified in the *Goal Model,* resulting in one or more Activity diagrams for each goal. The presented activity diagram specifies the activities and protocols performed by the *Great Power* role to attain the *Make Alliance* goal and the activities and protocols performed by the *Game Master* role to attain the *Control Negotiation Sessions* goal, both of which are roles of the *Diplomatic phase* sub-organization. As the goals of these two roles are related, we decide to specify their activity diagrams in just one diagram with tree swim lines, two for the interaction between the two instances of the *Great Power* role (active and passive) to attain the *Make Alliance* goal, and the third for the interaction between the *Game Master* role and the instances of the *Great Power* role to attain the *Control Negotiation Sessions* goal.

As is shown in Figure 8, the flow of actions performed by the *Great Power* active role to attain the *Make Alliance* goal begins with a fork that gives the control to one initiator protocol: *Meet in private groups,* and to one reactive protocol: *Interrupt negotiation session (Game Master)*. The first protocol is initialized by the *Great Power* active role and result in the reactive protocol *Meet in private groups (Active:Great Power)* of the *Great Power* passive role, while the other is a reaction of the *Great Power* role to the *Interrupt negotiation session* protocol initialized by the *Game Master* role if the negotiation time has ended. If this protocol is performed, the *Great Power* active role must terminate the flow of action.

After the *Meet in private groups* protocol has been performed, the *Great Power* active role must perform the *Decide who to trust* activity in order to attain the *Make Alliance* goal. The *Great Power* passive role has the same flow of actions as the *Great Power* active role, with the difference that its *Meet in private groups (Active:Great Power)* protocol is a reaction to the *Meet in private groups* protocol initialized by the *Great Power* active role, and since this is a passive instance of the *Great Power* role, it does not end the flow of actions.

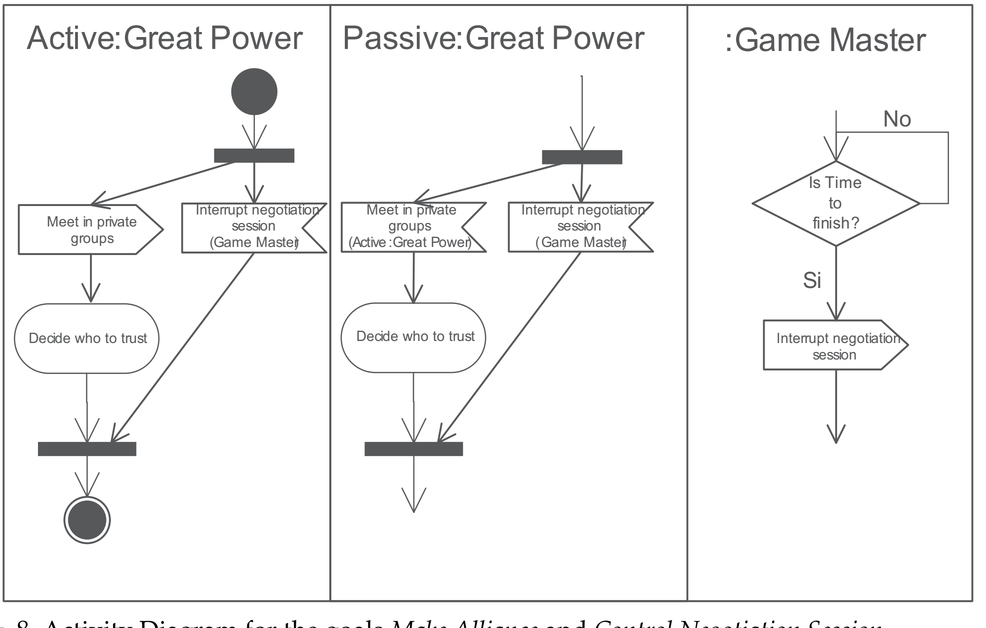


Fig. 8. Activity Diagram for the goals *Make Alliance* and *Control Negotiation Session*

The second task that must be performed in the *Requirements Specification* activity is *Develop Environment Model* using the information from the *Role Model* and *Domain Model* generated in the *Requirements Definition* activity as input. Figure 9 shows the permissions of the *Great Power*, *Game Master* and *Unit* roles with regard to the *Domain Model* resources that each role needs to perceive or modify in order to attain its goals. The *Great Power* role perceives the following entities in the system: other instances of *Great Power*, *Units*, *Map*, *Provinces*, B*oundary* and *Country*; and can modify: *Supply Center*, *Document*, *Order*, *Retreat* and *Adjustment*. The *Game Master* role perceives the following entities in the system: *Great Power*, *Units*, *Map*, *Provinces*, *Supply Center*, *Boundary* and *Country*; and can modify: *Order*, *Retreat* and *Adjustment*; but cannot perceive or modify the *Document* entity. Finally the *Unit* role perceives the following entities in the system: *Great Power*, *Map*, *Provinces*, *Boundary*, *Country*, *Document*, *Order*, *Retreat* and *Adjustment*; but cannot perceive or modify the following entities: other instances of *Unit* and *Supply Center.*

The third task that must be performed in the *Requirements Specification* activity is to *Define Organizational Rules,* using the information from the *Domain Model* generated in the *Requirements Definition* activity and the *Behavior Model* of the current activity as input. In the current domain, the important rules to identify are the general rules of the game, the number of players, the rules concerning the movement of the units depending on the type of unit and on the type of provinces the move take place in, etc. Table 1 shows an extract from the *Organizational Rules Model*.

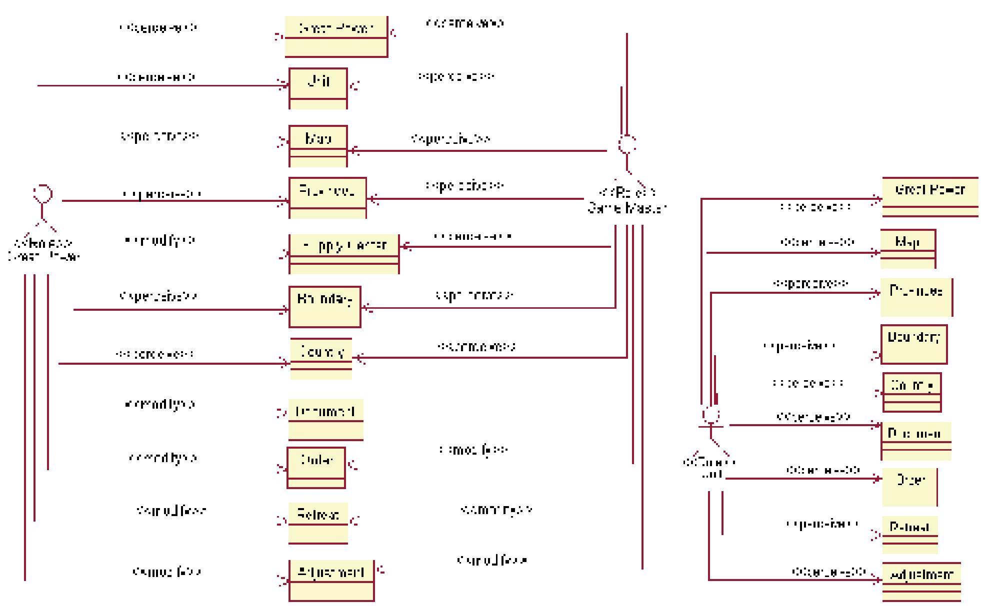


Fig. 9. Diplomacy game Environment Model

Finally, as a result of performing the *Requirements Specification* activity we obtain the *Behavior Model which* is composed with all the Activity diagrams (see example in Figure 8). We also obtain the *Environment Model* (see Figure 9). And finally, a table representing the *Organizational Rules Model* is obtained (see Table 1).

Description
The game is divided into a two year tour: Spring four-phase turn and Fall five-phase turn
Spring four-phase turn has phases: Diplomatic, Order Writing, Order Resolution and Retreat and Disbanding
Fall five-phase turn has phases: Diplomatic, Order Writing, Order Resolution and Retreat and Disbanding, Gaining an Losing
Only seven players may perform the role of "Great Power"
When 18 supply centers belongs to a "Great Power" the game ends and the winner is that "Great Power"
At the start of the game each "Great Power", except Russia, controls 3 supply centers
At the start of the game, the "Great Power" Russia controls 4 supply centers
Maximum time in the first diplomatic phase is 30 minutes

Table 1. Extract of Organizational Rules Model

5.3 Discussion

With the definition of the *Diplomacy Game Refinement Tree* (see Figure 5), the requirements engineer is able to identify the overall goal of the system, the decomposition of the system in a hierarchy of sub-organizations, roles involved in each sub-organization, and the goals which are carried out by each role in the corresponding sub-organization. The *Diplomacy Game Refinement Tree* provides information for the *organizational*, *functional*, and *structural* perspectives of the case study system. In addition, the social behavior needed for each role to carry out their goals is specified by mean of one Social behavior diagram for each sub-organization (see Figure 6). The case study presents a variety of social characteristics that allow to fully evaluating the proposed Social behavior diagram. In particular, we identified relationships of collaboration, disposition and veracity. The Social behavior diagrams provide information for the *social behavior* perspective. Moreover, the relevant entities of the environment of the game are identified in the *Diplomacy Game Domain Model* (see Figure 7), providing information for the *organizational* and *structural* perspective.

With the construction of one Activity diagram for each goal, the requirements engineer is able to refine each goal in activities and protocols, and also to refine the social behavior identified in the previous activity. It is proper to mention that the collaboration relationships identified in the Social behavior diagrams is refined in the Activity diagrams. As an example, the initiator protocols and reactive protocols in Figure 8 show the specification of the collaboration relation identified in the Social behavior diagram of Figure 6. The Activity diagrams also provide information for the *functional* and *social behavior* perspectives.

Furthermore, the *Diplomacy Game Environment Model* (see Figure 9), identifies the resources of the system, and defines the permissions that roles have in those resources, providing information for the structural and functional perspectives. The organizational rules of the game are specified (see Table 1), providing information for the *organizational*, *structural* and *functional* perspectives.

Finally, due to its characteristics, the *Diplomacy Game* case study offers a good example to validate the feasibility of our approach to model the requirements of a MAS covering its *organizational*, *structural*, *functional*, and *social behavior* properties.

6. Conclusions and future work

In this work we proposed a requirements modeling process for MAS. The approach is organized into two main activities: *Requirements Definition* and *Requirements Specification*. In the *Requirements Definition* activity the following is modeled: (a) the organizational structure and structural properties of the system; (b) the functional behavior of the system; and (c) the domain entities and their relationships. In the *Requirements Specification* activity the requirements specifications are refined, identifying: (a) the interactions on which the social behavior of the system is based; (b) the mains activities which conform the functional behavior of each role; (c) the permissions of the roles in the domain entities; and (d) the structural and functional behavior. This process supports the four perspectives that characterize a MAS: *organizational*, *structural*, *functional* and *social behavior*. We believe that this proposal addresses the need for a requirements modeling process for MAS because it incorporates specific abstractions needed to capture and specify these four perspectives. In particular, the definition and specification of features of social behavior at the requirements level will increase the quality of specifications, thus providing the expressiveness needed by the MAS in an early stage of the software development process.

As a case study, we have presented the requirements modeling of the *Diplomacy Game*. The game development domain, given its characteristics, particularly allows us to observe and reason about different ways in which to identify, define, and specify requirements of social behavior, in addition to the organizational (because of the various phases of which a game is composed), the structural (owing to the different types of elements used), and the functional (because of the different actions to be performed). Moreover, according to Google Trends and the ESA annual report (ESA, 2010), game development is one of the business markets that has undergone most growth in the last few years. The social behavior characteristics (negotiation, cooperation, pro-activity, etc.) and complexity of games make them appropriate subjects for resolution with the agent-oriented paradigm.

Currently, we are working on the definition and specification of other social behavior characteristics, such as adaptability and mobility. In addition, plan to empirically validate our approach through a series of experiments using game development experts as subjects.

We are also working to extend this agent proposal to a model-driven development approach in the context of a project named Multi-modeling Approach for Quality-Aware Software Product Lines (MULTIPLE). MULTIPLE focuses on the definition and implementation of a technology framework for developing software product lines of high-quality software in the context of model-driven development. This extension would facilitate the integration and traceability among the artifacts generated during the requirements modeling process and the analysis and design artifacts used in the MAS development. This will increase the overall quality of the MAS to be developed. Finally, we plan to build a tool that will support the overall process defined, using the Eclipse Development Environment (The Eclipse Fundation, 2010).

7. Acknowledgments

This research is funded by the Ministry of Education and Science, under the National Research Program, Development and Innovation, MULTIPLE project (TIN2009-13838).

Lorena Rodriguez has a scholarship under the College Scholarship Program and Support of the Scientific and Technological Production, in the context of the Social Responsibility Program of the Itaipu Binacional/Parque Tecnológico Itaipu-Py.

8. References

Anton, A. (1996). Goal-based requirements analysis. *Proceedings of the 2nd International Conference on Requirements Engineering (ICRE '96)*, pp.136–144, ISBN: 0-8186-7252-8, Colorado Springs, April 1996, IEEE Computer Society, Colorado

Blanes, D.; Insfrán, E.; Abrahão, S. (2009) a. Requirements Engineering in the Development of Multi-Agent Systems: A Systematic Review. *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, pp. 510 – 517, ISBN: 0302-9743, Burgos, Spain, September 2009, Springer-Verlag, Berlin, Heidelberg

Blanes, D.; Insfrán, E.; Abrahão, S. (2009) b. RE4Gaia: A Requirements Modeling Approach for the Development of Multi-Agent Systems. *International Conference on Advanced Software Engineering and Its Applications (ASEA'09)*, pp. 245 – 252, ISBN: 978-3-642-10618-7, Jeju Island, Korea, December 2009, Springer, Berlin

Cernuzzi, L.; Cossentino, M.; Zambonelli, F. (2005). Process models for agent-based development. *Engineering Applications of Artificial Intelligence*, 18, 2, (March 2005) page numbers (205 – 222), ISSN: 0952-1976

ESA (2010). Entertainment Software Association, Industry Facts. Last accessed on September 14, 2010, en http://www.theesa.com/facts/index.asp

Fabregues, A.; Navarro D.; Serrano A.; Sierra C. (2010). dipGame: a Testbed for Multiagent Systems, *Proceeding of the ninth International Conference* of *Autonomous Agents and Multi-Agent Systems*, Toronto, Canada, May 2010

Falkenberg, E.D.; Hesse, W.; Lindgreen, P.; Nilsson, B. E.; Oei, J.L.H.; Rolland, C.; Stamper, R. K.; Van Assche, F.J.M.; Verrijn-Stuart, A. A.; Voss., K. (1998). *FRISCO - A framework of information system concepts – The FRISCO Report.* Task Group FRISCO, ISBN: 3-901882-01-4

Giorgini, P.; Kolp, M.; Mylopoulos, J.; Castro, J. (2005). Tropos: A Requirements-Driven Methodology for Agent-Oriented Software. In: *Agent-Oriented Methodologies*, Brian Henderson-Sellers; Paolo Giorgini, page numbers (20-45), Idea Group , ISBN: 1591405815, USA

Gómez-Sanz, J.; Pavón, J. (2003). Agent Oriented Software Engineering with INGENIAS, *Proceedings of the 3rd Central and Eastern Europe Conference on Multiagent Systems (CEEMAS '03)*, pp. 394 – 403, ISBN: 3-540-40450-3, Prague, june 2003, Springer-Verlag, Prague

Hector, A.; Lakshmi Narasimhan, V. (2005). A New Classification Scheme for Software Agents. *Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05)*, pp. 191 – 196, ISBN: 0-7695-2316-1, Sydney, Australia, July 2005, IEEE Computer Society, Washington, DC

Huzam S. F. Al-Subaie; Maibaum Tom S. E. (2006). Evaluating the Effectiveness of a Goal-Oriented Requirements Engineering Method. *Proceedings of the Fourth International Workshop on Comparative Evaluation in Requirements Engineering*, pp. 8 - 19, ISBN: 0-7695-2712-4, Minneapolis/St. Paul, Minnesota, September 2006, IEEE Computer Society Washington, DC

Maiden, N. (1998). CREWS-SAVRE: Scenarios for Acquiring and Validating Requirements. *Automated Software Engineering*, 5, 4, (October 1998) page numbers (419 - 446), ISSN: 0928-8910

Odell, J.; Parunak, H. V. D.; Bauer, B. (2000). Extending UML for agents. *Proceeding of the 2nd Int. Workshop on Agent-Oriented Information Systems*, pp. 3 – 17, Berlin, iCue Publishing

OMG (Object Management Group). Software Process Engineering Meta-Model (SPEM), version 1.1. Last accessed on March 11, 2010, en http://www.omg.org/cgi-bin/doc?formal/05-01-06.pdf

Rodriguez L.; Hume, A.; Cernuzzi, L.; Insfrán, E. (2009). Improving the Quality of Agent-Based Systems: Integration of Requirements Modeling into Gaia. *Proceedings of the Ninth International Conference on Quality Software (QSIC '09)*, pp. 278 – 283, ISBN: 978-0-7695-3828-0, Jeju, Korea, August 2009, IEEE Computer Society Washington, DC

The Eclipse Foundation. Last accessed on September 2010, from http://www.eclipse.org

Van Lamsweerde, A.; Darimont, R.; Letier, E. (1998). Managing conflicts in goal-driven requirements engineering. *IEEE Transactions on Software Engineering*, 24, 11, (November 1998) page numbers (908 – 926), ISSN: 0098-5589

Wizards (2010). The rules of Diplomacy. The game of international intrigue. . Last accessed on September 14, 2010 en http://www.wizards.com/avalonhill/rules/diplomacy_rulebook.pdf

Yu, E. (1997). Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. *Proceedings of the 3rd IEEE Int. Symp. on Requirements Engineering (RE '97)*, pp. 226 – 235, ISBN: 0-8186-7740-6, Washington D.C., USA, January 1997

Zambonelli, F.; Jennings, N.; Wooldridge, M. (2003). Developing Multiagent Systems: The Gaia Methodology. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 12, 3, (July 2003) page numbers (317 – 370), ISSN: 1049-331X

2

A Multi-Agent System Architecture for Sensor Networks

María Guijarro, Rubén Fuentes-Fernández and Gonzalo Pajares
University Complutense Of Madrid, Madrid, Spain

1. Introduction

Today it is increasingly important a good design architectures that support sensors. This chapter shows how the design of the control systems for sensor networks presents important challenges because of using sensor networks has design problems. Besides the traditional aspects about how to process the data to get the target information, engineers need to consider additional aspects such as the heterogeneity and high number of sensors, and the flexibility of these networks regarding topologies and sensors in them.

The increasing availability of sensors plugable in networks at low costs is rapidly increasing their use for different applications like smart spaces or surveillance systems (Tubaishat, M. & Madria, S. 2003).These networks pose important challenges for engineers working in the development of the related control systems. Some of the most relevant are:

- *Potential high number of nodes*. The current trend is to set up networks densely populated with sensors and a minor number of controllers (Yick et al., 2008). These magnitudes imply that engineers must consider issues such as the organization of the communications and local pre-processing of data to save bandwidth and get suitable response times.
- *Sensor heterogeneity*. These networks include a wide variety of types of devices (e.g. cameras, motion sensors or microphones) whose management and usage differs (Hill et al., 2000). These sensors are usually specialized in specific applications, so they do not offer the same services. The combination of different types of sensors in a network and the use of its data requires a high-level of modularity and adaptability in the architecture.
- *Changing network topology*. Sensor networks are less stable than traditional computer networks (Yick et al., 2008). Their sensors are more prone to fail than conventional computational devices: they frequently operate unattended in environments that can lead them to malfunction, and with very limited resources. A common way to overcome sensor failure is redeploying new sensors, which further changes the network topology. These changes make that the control of the network must deal with ad-hoc topologies to attend the communications needs of a given moment with the available resources.
- *Several levels of data processing*. Processing of data happens at both local and global levels (Tubaishat, M. & Madria, S. 2003). Since sensors can be deployed over quite wide areas, the management of data may need to be contextualized, for instance to determine what

signals are relevant in a situation. Nevertheless, centralized processing is also necessary, mainly for the transformations and integration of data. Thus, the architecture of the control must deal with groups at different levels that need to coordinate.

- *Unreliable networks of reduced bandwidth*. The network established in these cases is highly unreliable when compared with wired networks (Yick et al., 2008). It is usually a wireless network where the hostile environment produces intermittent connectivity with a high variability in the conditions for communication. This issue requires solutions similar to those of the changing topology.

These challenges have been addressed in several works, though usually focusing only on the solutions for some specific issues. For instance, literature (Akyildiz et al., 2007; Baronti et al., 2007; Tubaishat, M. & Madria, S. 2003; Yick et al., 2008) reports works on routing in ad-hoc networks to minimize energy consumption, optimal data processing to reduce computation time in sensors or data integration in specific domains. However, the integration of the different solutions is not a trivial problem and research in architectures for these networks pays attention to it.

The architectures proposed for these networks usually consider some basic infrastructure and a component model. The infrastructure provides basic services for all the components, and the components model specify the interfaces and behaviour that components must provide to be integrated in the network. Examples of these works are the architectural principles in (Estrin et al., 1999) and MADSN (Qi et al., 2001), TinyOS (Hill et al., 2000) and Tenet (Gnawali et al., 2006). (Estrin et al., 1999) is probably one of the first works discussing the specific problematic of sensor networks. It advocates for architectures where data processing is performed as close as possible to the sources of those data, probably in the sensors themselves. Sensors are also responsible of communication, sometimes supported by communication specific devices. MADSN (Qi et al., 2001) proposes changing the paradigm for data integration from one where all the data are transmitted to a central processing node, to another in which mobile agents travel to the nodes that collect data and make there the processing, propagating only ellaborated data. TinyOS (Hill et al., 2000) is focused on infrastructure. It is an operating system based on micro-threading, events and a simple component model. A component has tasks, commands and handlers that are executed in the context of an encapsulated fixed-size frame, and they operate on the state of the component. Tenet (Gnawali et al., 2006) is a model of components and its supporting libraries built on top of TinyOS. It proposes a two-layered architecture with simple sensors and masters. Sensors get data and only make basic signal processing. Masters perform the integration of data using more powerful computational devices. These examples are also illustrative of the main limitations of these architectures:

- *Constraining architecture.* Most of times, the architecture includes a restricted component model. Systems need to adhere strictly to its principles, which imply developing specific interfaces and conforming to certain rules of behaviour. This is the case of TinyOS (Hill et al., 2000) and Tenet (Gnawali et al., 2006).
- *Lack of a complete vertical solution.* Although these architectures are conceived to integrate the solutions of several aspects and to give complete models on how to build a sensor network, they usually do not cover the whole design. For instance, most of the proposed examples (Estrin et al., 1999; Gnawali et al., 2006; Hill et al., 2000) are mainly related with communication and sensor management issues, but they do not say anything about the design of the sensor controllers. Even if as proposed in (Estrin et al., 1999) controllers are in sensors, the problematic of their design is focused more on

communications and data integration than on gathering data and the processing of the raw signals.

- *Lack of a supporting modelling language and development process.* The proposed architectures focus on design principles (Estrin et al., 1999; Qi et al., 2001) or infrastructure (Gnawali et al., 2006; Hill et al., 2000). Nevertheless, this is not enough to build a sensor network. Engineers need a development process that indicates them what the relevant requirements to consider are, the design alternatives, and the steps to follow in the development. The industrial use of such process demands customizable modelling languages and automated support tools.

In order to address the previous limitations, some works have proposed multi-agent systems (MAS) (Weiss, G., 2000) as the basis for the development of sensor networks. A MAS is composed of a large number of agents and other computational artefacts. These agents are social entities, which need to interact with other agents to achieve the satisfaction of system goals. Agents are goal-oriented components, i.e. they rationally choose for execution those actions that will potentially contribute to satisfy their objectives. These choices depend on the information they have in their mental states about their environment, past experiences and themselves. The works in this approach usually see sensors as devices controlled by agents. This choice meets some of the aforementioned requirements of sensor networks. Sensors are only responsible of data gathering and basic processing, while computationally expensive processes are assigned to agents. This organization gives freedom of choice to put the execution of data processing either mainly in the sensors or in the controllers. Despite of this common feature, differences between approaches are important. From the point of view of the goal of this work, they do not achieve a complete architecture and process to design it either because they are too focused on some specific issues (e.g. optimization of communications (Qi et al., 2001), only provide an architecture or just a development process which is usually a general-purpose agent-oriented software engineering (AOSE) methodology.

This work addresses these issues with a twofold solution: a standard architecture for sensor networks able to deal with different design choices; a design language oriented to the kind of abstractions appearing in it, and a development process for such systems. This chapter focuses only on the first two elements, though it gives a brief introduction to the process. To provide these elements, this work adopts as its basis a well-known general purpose AOSE methodology, INGENIAS. INGENIAS (Pavón et al., 2005) covers the full-development cycle from analysis to coding with a model-driven engineering (MDE) approach (Schmidt, D. C. 2006) supported by tools. The definition of its modelling language and tools is based on metamodels. Metamodels are a common way of defining formally modelling languages in MDE. The fact of having a formal definition of the language effectively constrains engineers during design to build proper models, and it also allows automated processes for code generation, both for tools and final systems. This allows its adaptation to new contexts by means of extensions of its metamodel.

The architecture proposed in this work considers sensor networks composed by sensors and controllers, therefore following common approaches with sensors and MAS (Pavón et al., 2007). However, it extends both definitions in several ways. A sensor is defined as an environment element with attached functioning parameters and an internal state, which can be modified using its methods. The sensor is also able to perceive events coming from its environment, and to raise events in order to notify changes in its state or that of its external environment. The architecture considers sensor networks composed by devices (which include sensors) and actors (i.e. agents). It extends common definitions for these concepts (Pavón et al., 2007) in several ways.

A device is defined as an environment element with attached functioning parameters, an internal state, and methods to work on that state. The device is also able to raise events in order to notify changes in its state. A sensor is a device that perceives events coming from its environment. Actors are similar to controllers in other approaches, but the architecture introduces for them a neat separation of concerns with roles. A role is defined by its goals, which are related with its responsibilities, and the capabilities (i.e. tasks) and resources (i.e. devices and applications) it has to achieve them. Different role types have exclusive skills. For instance, only device controllers can communicate with devices, and the group leaders have the power to impose certain goals to the members of their groups. The current version of the architecture includes several predefined role types, but this list can be extended to address new needs of sensor networks, such as secure communications or resource assignment (Tubaishat, M. & Madria, S. 2003). These roles are played by actors, which are agents with common inherited capabilities about goal management and task execution. Their specification focuses on how they implement the specific tasks related with their roles. The architecture defines teams of roles and their interactions to perform certain tasks, for instance, the setup or the dynamic addition of sensors.

The previous definitions of sensor, roles and agent partially match those of INGENIAS external applications, roles, and agents. Nevertheless there are relevant differences. For instance, INGENIAS external applications and sensors are both environment elements characterized in terms of offered methods and produced events, but sensors extend applications considering its internal state, and how this changes as a consequence of external events and method execution. Thus, this research has modified the INGENIAS metamodel to accommodate the new concepts and being able to use the INGENIAS tools.

2. Important

Although there are partial solutions for the design of sensor networks, their integration relies on ad-hoc solutions requiring important development efforts. In order to provide an effective way for their integration, this chapter proposes an architecture based on the multi-agent system paradigm with a clear separation of concerns. The architecture considers sensors as devices used by an upper layer of controller agents. Agents are organized according to roles related to the different aspects to integrate, mainly sensor management, communication and data processing. This organization largely isolates and decouples the data management from the changing network, while encouraging reuse of solutions. The use of the architecture is facilitated by a specific modelling language developed through meamodelling.

3. Agent development with INGENIAS

INGENIAS (Pavón et al., 2005) is a MDE methodology for the development of MAS. It comprehends a specific modelling language, a software process and a support tool. Following MDE principles, it defines its design modelling language with a metamodel. This metamodel is the basis for the semi-automated development of its tool, and also guides the definition of the activities of its software process.

MAS in INGENIAS are organizations of agents, which are intentional and social entities. Agents use applications, which represent the environment and system facilities. The models

to specify these MAS describe their environment, agents and interactions, both from the static and dynamic perspectives. The modelling language also includes a simple extension mechanism for agents through inheritance relationships: a new sub-agent type inherits all the features of its super-agent type, but it can also extend or constrain them. Table 1 shows the main INGENIAS concepts used by our approach.
The support tool of the methodology is the INGENIAS Development Kit (IDK). It provides a graphical environment for the specification of MAS design models. The tool can be extended with modules. The standard distribution includes modules for documentation and code generation based on templates. A template is a text file annotated with tags. These tags indicate the places where information from models has to be injected to get a proper instantiation. The instantiated template can describe, for instance, the code for an agent in a framework, the documentation of its goals, or the configuration files for its deployment. Engineers can use code components in models to attach specific code to entities. For instance, if engineers want to generate nesC (Gay et al., 2003) code, they first need to develop a template with the general description of an agent in that language; then the code generation module reads the design models of the sensor network, and for each agent appearing in them, it generates its specific code for nesC instantiating the template (i.e. replacing the tags with information from the models that includes the code components).

Concept	Meaning
Agent	An active element with explicit goals able to initiate actions involving other elements.
Role	A group of related goals and tasks. An agent playing a role adopts its goals and must be able to perform its tasks.
Environment application	An element of the environment. Agents/roles act on the environment using its actions and receive information from the environment through its events.
Internal application	A non-intentional component of the MAS. Agents/roles use it through its actions and receive information from it through its events.
Goal	An objective of an agent/role. Agents/roles try to satisfy their goals executing tasks. The satisfaction or failure of a goal depends on the presence or absence of some elements (i.e. frame facts and events) in the society or the environment.
Task	A capability of an agent/role. In order to execute a task, certain elements (i.e. frame facts and events) must be available. The execution produces/consumes some elements.
Interaction	A basic communication action. Agents/roles send with them information to other agents/roles.
Method	A basic imperative operation of an application described by its parameters and result.
Frame fact	An information element produced by a task, and therefore by the agents/roles.
Event	An information element spontaneously produced by an application.
Interaction	Any kind of social activity involving several agents/roles.
Group	A set of agents/roles that share some common goals and the applications they have access to. The behaviour of groups is specified with workflows involving its components. These workflows indicate their tasks, the elements these produce/consume and the agents/roles that carry them out. The workflows must fulfil the group goals through the achievement of the individual agent/role goals.
Society	A set of agents, roles, applications and groups, along with general rules that govern their behaviour.
Environment	The set of external applications with which the components of a MAS interact.

Table 1. Main concepts of the modelling language of the INGENIAS methodology.

There are two main reasons for the choice of INGENIAS in this work considering available alternatives with agents (Vinyals et al., 2008). First, its modelling language is a suitable basis for the extensions required for sensor networks. It considers concepts such as agents, roles and environment applications that are required in our architecture, and covers the interactions between system components with a high-level of detail. Second, INGENIAS strictly adheres to MDE principles. It defines its modelling language with a metamodel that is also the basis of the IDK development. This facilitates the modification of the language to house additional concepts and propagating these changes to the tool. Given the complexity of the development of sensor networks (Tubaishat, M. & Madria, S. 2003; Yick et al., 2008), the availability of support tools (e.g. for coding, debugging or reporting) is mandatory to get a high productivity. Nevertheless, the IDK has the shortcoming of using an ad-hoc approach for transformations based on modules and templates, although there are ongoing efforts to support more standard approaches (García Magariño et al., 2009). The development process proposed in our work adopts standard transformation languages (Sendall et al., 2003) to manipulate models and code. This has two key advantages. First, the tools to develop and run these transformations are already externally available, so there is no need of new developments. Second, these languages focus on the description of the transformations, which facilitates their understanding as this is not blurred with low-level details about processing design models (e.g. reading the input file, managing syntax errors or generating the output file).

3.1 An agent-based modelling language

The design of MAS to manage sensor networks in the presented approach uses specializations of general agent concepts. The purpose of these specializations is acting as a guide for engineers: they indicate the concepts that should appear in the specifications and how they are related. The main extensions of our approach to the INGENIAS (Pavón et al., 2005) conceptual framework appear in Fig. 1. with their main relationships. The mechanism used for the metamodel extension is its direct modification (Cook, S., 2000). Note that profiles cannot be used since this is not an UML extension, and INGENIAS limits inheritance to agents.

Fig. 1. represents elements of the metamodel of the modelling language in our approach. Nodes and links respectively represent meta-entities and meta-relationships. Meta-relationships with triangles and diamonds are standard (i.e. non specific of INGENIAS) representations of inheritance and aggregation (i.e. whole-part link) relationships. Numbers in the ends of relationships are cardinality indications. The stereotypes of nodes (represented between angle brackets) are the names of the INGENIAS meta-entities that our meta-entities extend. The meta-entities have the features (e.g. attributes and relationships) of the extended meta-entities and add new features and constraints. For instance, at the meta-modelling level, there are meta-entities device and controller that are modifications of the INGENIAS meta-entities environment application and role respectively. These meta-entities are related with a meta-relationship WFUses, which is restricted to connect this pair of meta-entities. These meta-elements are instantiated in models. For example, a model can contain instances of the device meta-entity, which can only be related with instances of the controller meta-entity through instances of the WFUses meta-relationship. The rest of the section discusses the concepts present in Fig. 1.

A sensor network in the proposed architecture distinguishes between reactive and active components. Reactive components receive requests or notifications of events, and generate

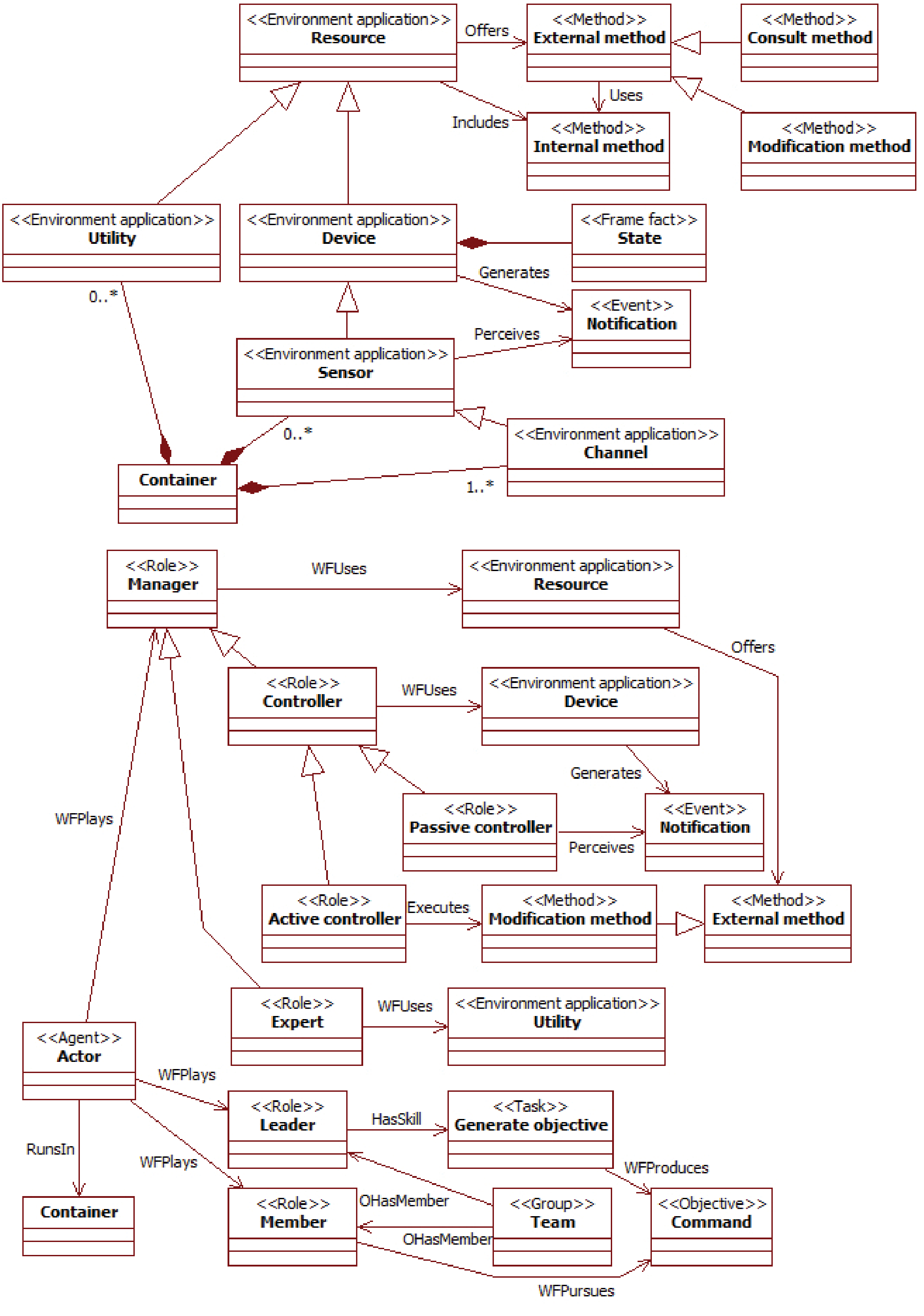


Fig. 1. Partial metamodel of agent-based concepts for sensor networks.

answers for them that only depend on the input and some internal state if this exists. Active components take initiatives on their own that contribute to satisfy the system goals. The basic type of reactive component is the resource, and the actor of active component. Actors are a specific type of agents that use resources. Their work is organized through the roles they played. Roles represent prototypical aspects of the activities in the network. A role indicates the goals it pursues and the available elements to achieve them, which can be information, capabilities and resources.

A *resource* is an external application. Its specification is known, but neither its behaviour nor its interfaces can be modified. The only way to interact with it is what their external/public interfaces allow. The actions available for this external manipulation of resources are represented by *external methods*. These methods can change the internal state of the resource, i.e. *modification method*, or just consult it, i.e. *consult method. Internal methods* can be used to provide information about the internal behaviour of the resource with specification purposes, but other components of the MAS cannot invoke them. A resource may have functioning parameters that influence its behaviour. These parameters can determine for instance, the threshold of certain operations or the initially available energy. Resources represent different elements appearing in sensor networks. A *utility* is a stateless resource. It corresponds to a computational facility available for the network, such as data normalization, combination of different signals or information conversions. *Devices* are stateful resources able to generate events called *notifications*. The state is characterized in terms of frame facts, which are the units of information in INGENIAS. Devices offer specific methods to manage the subscription of other components to their notifications. A subclass of devices is *sensors*, which generate events but are also able to perceive them in the surrounding environment. Thus, the behaviour of a sensor is characterized in terms of a state machine that changes its state according to the execution of methods and the appearance of events from the environment. A *channel* is a particular type of sensor intended for communication. It is able to send and receive information over a medium and perform basic tests on it.

These resources are used by *manager* roles to provide services in sensor networks. The language distinguishes two types of managers depending on if they work with devices or utilities.

The *controller* is the role with access to devices. According to the rights it has over it, there are two types of controllers. A *passive controller* can only consult the device state with consult methods and perceive those events to which it subscribes. The *active controller* is able to make requests to change the device state using its modification methods. In this way, several access levels can be granted to controllers of the same devices.

The *expert* is the role in charge of utilities. This role specifies the knowledge and skills required to manipulate an utility, as well as how to obtain additional information that can be extracted from sequences of data manipulations over time. For instance, an expert can store information about temporal series of signals to draw conclusions about trends.

Another concept central in the proposed solution is that of *team*. A *team* is a hierarchical INGENIAS group that comprehends a *leader* role and several *member* roles. The leader has the right of posing new commands to the *members* of its team, where a *command* is a kind of objective. Roles receiving the commands must include them in their agenda, but their management of them depends on their design. The leader can be also the provider of a given service for all the members of its team. Teams facilitate setting up basic groups of collaborating roles. For instance, there are groups for the initialization of the network, solving issues of quality of service, communications or data processing. These teams constitute design blocks that can be reused in different specifications.

The previous roles are played by roles and actors. When a role plays another role, it adds the features of that role to its own ones. The actors are agents with common skills for the management of goals (e.g. decomposition, checking their state or removing when satisfied), planning for their achievement (in terms of the available information, resources and capabilities) and basic communications (both with agents and resources). When an actor plays a role, it fulfils the standard behaviours specified by the role, that is, it implements its capabilities, has actual access to its resources, and manipulates the related goals and information. The actor can have additional elements beyond those of its roles. Note that an actor manipulates all these elements globally. For instance, the satisfaction of a goal linked to a certain role can be the result of the information produced with a capability related with another role.

The previous elements run in *containers*, which represent deployable computational devices. A container has basic processing capabilities that allow the execution of agents, and at least one channel for communication. Additionally, it can include an arbitrary number of resources. Note that, given the relationships and constraints in the metamodel, a device and its managers run in the same container.

In order to provide a simple extension mechanism for the language, this approach also generalizes the INGENIAS inheritance relationship. It is not constrained to agents but can be applied to any concept with an equivalent meaning: a sub-concept inheriting from a super-concept has all its features but it can extend or constrain them with additional models.

4. Architecture for sensor networks

The metamodel for sensor networks just defines the modelling primitives that can be used when specifying these networks as MAS. However, it cannot specify how these elements should interact to provide the expected functionality. The architecture provides this information. This section focuses on its description through its main teams. The list is not exhaustive, as more teams can be specified to address new needs. The description of teams includes their purpose, and the characterization of their leader and member roles regarding their responsibilities. Note that when talking about roles performing actions, it is really the actors playing those roles that perform the actions, as roles are just functional abstractions.

The *initialization* team is aimed at setting up all the components of a container and providing them with the initial information required for their proper functioning. Its team leader is the initializer and its members play the role of targets of the initialization. The *initializer* creates all the actors in its container and sends them the information about the managers they play. Then, each manager receives a list of the assigned resources, and the notifications and external methods it can use. If required, it can also obtain information to initialize the resources. Besides, each manager receives information about all the teams it belongs to, including the type of team, its leader and the role of that manager in it. Note that these teams can involve roles whose actors are not running in the same container.

An *information process* team focuses on the generation of information from the data of devices. Its team leader is a consumer for that information. It organizes the gathering and processing of data. Team members play the role of *providers* of information and can be passive controllers or experts. The activity of these teams can begin either with a request from the customer or with a notification from a device. In the second case, a passive controller provider captures an event raised by a device and notifies it to its consumer. From this point forward, both scenarios are the same. The consumer may send additional requests to its manager providers: to passive controllers in order to collect additional data; to experts to further manipulate these data before their use. Note that with this approach, the consumer itself can be regarded as a manager that provides services of a higher-level, as it encapsulates the interactions with a group of resources and its managers.

Communication teams refine the INGENIAS communication schema, as they give further details about how interactions are transmitted between different actors and roles. They manage communication through *channels*. The *communicator* is both the team leader and the active controller of the channel. The rest of the members of the team play the role *customers* in the communication. All the customers in a communication team are able of direct communication between them, but they need to ask its communicator for external communication outside the team. These teams encapsulate the use of the communication infrastructure and related algorithms, which makes transparent the communication capabilities of other network elements from the design point of view. Engineers only need to guarantee that each role or actor that needs to communicate belongs to a communication team in order to have access to a communicator. In order to optimize communications (e.g. latency or energy consumption) and perform message routing, communicators need to build a rough map of the nearby communicators. Containers have a limited range of communication, so some messages may need several hops to reach its final destination. To build the map, a communicator broadcasts a request of information amongst other communicators in range. Available communicators answer this request with information about the features of their service, and take note of the sending communicator.

Load balancing teams are intended to keep the quality of service in the network. Sensor networks face to several situations that can require their dynamic reconfiguration. Some of them were outlined in the introduction, such as failure of sensors or communications, but also sensors overloaded with requests or replacement of the failing customer for some data. Although different, all these situations are solved through the collaboration of two sub-teams. First, there is *a failure notification* team where a team leader *referee* controls a group of team member *watchers* that can warn of potential failures in the behaviour of some observed elements of the network. For instance, a controller can be the watcher of a sensor: when this sensor depletes its energy, it does not longer answer the requests of its watcher, which raises to its referee the information about the failure. The referee evaluates that information and if it determines that there is need of acting, a repairer team begins working. A *repairer* team has as leader a *dispatcher* governing a set of *referees* and *initializers*. When a dispatcher receives the notification of a failure, it looks for some replacement. The replacement can be obtained either asking other referees in the team for a component with similar features or asking an initializer to create a new one if possible. For instance, in the case of failure of a sensor, the replacement could be another sensor in a container near the location of the original one, but if an expert is failing, a new one can be created and assigned to the utilities of the original one. The dispatcher informs of the replacement to the involved referees, which send to the initializers in their containers the information to update. For instance, adjustments need to be made in the state of the replacement or the teams depending on it.

Note that any container must have running at least two teams. The initial setup requires one initialization team, and integration with other elements of the network a communication team. Executing these teams requires at least one actor which plays the initializer and communicator roles.

The architecture involving these teams pursues satisfying three main objectives. First, it facilitates the design of sensor networks decoupling the different responsibilities in roles and teams. Second, it looks for networks that can semi-autonomously reconfigure themselves to address new situations, a concept present in current research in autonomic computing (Kephart, J. O. & Chess, D. M., 2000). Third, it achieves the extensibility of the design of systems to control sensor networks through new teams.

5. Development process

In this chapter a simple model-driven development process customized is included to develop the control system of sensor networks following the architecture in section 4. A model-driven process focuses the development on design models. Engineers refine these models from abstract representations to those models closer to the intended target platform, and finally to code. The refinements are partially supported by automated transformations. The process proposed in this approach is based on the software process of the INGENIAS methodology (Pavón et al., 2005). It adds to the INGENIAS process several specific activities aimed at identifying the elements required in a sensor network. These elements are those defined in the modelling language and organized in the teams of the architecture. Fig. 2. shows the resulting process. Activities 1-7 are specific of the current approach, while activities 8-13 summarize INGENIAS activities. The process takes as input a previous analysis of the data required as output of the network and the sensors able to provide them, and produces as output the code of the control system for the sensor network.

The design of the network begins with activity 1. Engineers determine the containers of the network, i.e. the computational devices able to execute code and transmit information. These are usually the sensors, but also additional devices such as computers or communication facilities can be considered here. This activity also identifies the resources: the sensors that gather data from the environment; the utilities that represent services that actors use to process data. The activity distinguishes the two aspects of the sensor, as resource and container. Note that the modelling language provides different concepts for these aspects, and therefore assign to them particular features that must be considered in the design models. When these elements have been identified, engineers assign them initialization and communication teams. As discussed in section 4, these teams are mandatory for every container.

Decision 2 and activities 3-4 are intended to organize complex processing and integration of data. According to the architecture, information process teams are responsible of these activities. Engineers identify in decision 2 and activity 3 specific data that must be generated in the network. For each group of data, activity 4 designs the corresponding team. First, engineers discover the sensors that provide the source data. For each sensor, they must assign at least one active controller and a passive one. The first one is required in the initialization, and the second one to provide access to sensor data. Next, engineers must identify the data transformations required to get the final information. Some of them are achieved using utilities of the network. For these utilities, engineers assign an expert. Finally, the team is composed by the passive controllers of the sensors and the experts of the utilities playing the role of providers, and a customer to integrate and consume the information. The identification of this kind of teams finishes when all the complex calculation of data has a team assigned.

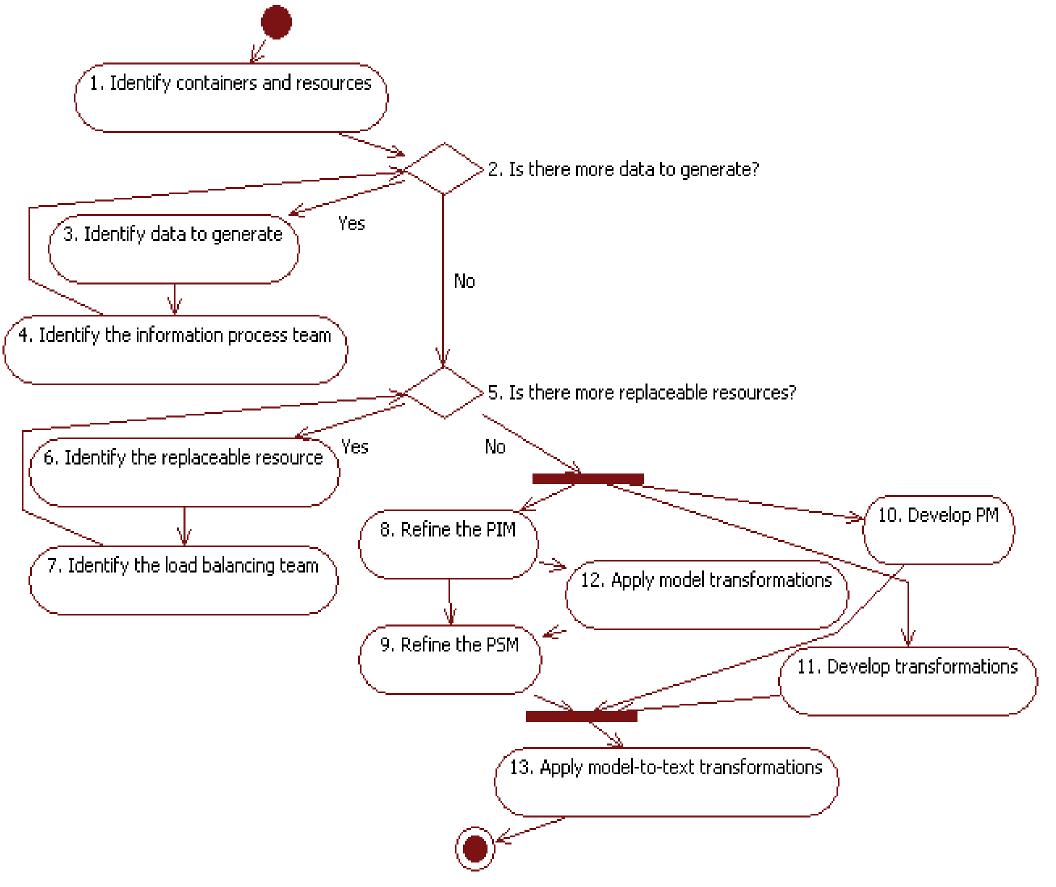


Fig. 2. Process for the development of sensor networks.

Decision 5 and activities 6-7 are intended to specify the teams that manage the dynamic adaptation of the network. Engineers begin this design with decision 5, where they find out what the elements are that can fail or be incrementally set up or deployed during the working of the network. This identification considers resources, roles and actors. For every element identified in this decision, activity 6 carries out an analysis regarding its potential replacements, and how they can be located and evaluated to find the best suited if several are available. Activity 7 designs the specific team related with this replacement. It includes a watcher that monitors the element. In the case of a device it is a passive controller, for a utility it is an expert, and for a role or agent it can be a customer that communicates with it. The team also needs a referee that evaluates when the failure needs to be notified for a potential replacement. The repairer team includes referees related with the same type of elements and similar features. For instance, for sensors they can be referees of nearby sensors and for roles other agents in the same container able to work with the same resources. As an alternative, initializers can be used to set up new roles or agents in these teams. Each of these teams must also identify its dispatcher, which selects the best alternative for a required replacement.

After these activities, engineers have available PIM of the resulting system according to the architecture. These PIM describe the devices, agents and roles, the information they exchange, and their interactions; they do not contain details on the final target platform, for instance about energy levels or low-level control commands for the sensors. Activities 8-13 follow the INGENIAS process to refine these models and generate the final code of the control system.

Activity 8 adds several INGENIAS PIM to the MAS specifications. Organization models define agents and groups outside the architecture, and assign to the groups workflows that describe their work. This allows refining the teams when complex processing of data needs further specification. Agent models refine actors and roles with additional goals, capabilities and information. These models also establish the pieces of information whose appearance determine when a goal is satisfied or failed. Tasks/Goals models map tasks with the goals that satisfy them, and hierarchically decompose goals and tasks into sub-elements. Interaction models describe actor interactions in terms of goals pursued, information exchanged and tasks performed. These models provide the details of the previous architectural design, though they are not always required. For instance, if engineers do not need to refine teams beyond what is said in the architecture, they do not use organization models.

Activity 9 and 10 develop the models required for the final target platform. Activity 10 develops the PM corresponding to the target platform. These PM include information about how to translate general concepts to specific elements in the platform. As explained in section 3, INGENIAS uses templates to represents PM. In case that these PM are available from previous projects, activity 10 can be omitted. Activity 9 develops the PSM of a specific design for the target platform. The PSM provide two main types of information. First, resources include their functioning parameters for the target platform, which can describe their limits about energy, memory or computational power. Second, engineers provide with code components attached to modelling entities the code specific for them. That is, part of the code required for the final system cannot be extracted from models, as models abstract the specific low-level details of the behaviour of systems. For instance, there are not modelling primitives to describe complex algorithms, and templates only contain general code for concept types in a platform. Engineers can include the remaining information attaching INGENIAS code components to the elements in models.

Activity 11 considers the development of the transformations that support the semi-automated refinement of PIM to PSM in activity 12, and the generation of code from PSM in activity 13. In the case of an INGENIAS development, transformations are implemented as IDK modules. These modules support model transformations and model-to-text transformations. Model transformations are useful to represents standard refinements of model concepts. For instance, each actor needs several goals to manage its planning cycles (e.g. collect information, discard non-achievable goals, look for achievable goals), but these are standard and engineers do not need to write them for each actor; a transformation can automatically generate these goals for the available actors. The best-known example of model-to-text transformation is code generation. In this case, the IDK includes a module for this purpose in its standard distribution. For a given specification and target platform, this module operates as follows: (1) it identifies the templates for the concepts present in the specifications and the target platform; (2) it traverses the templates looking for their tags; (3) when it found a tag, it replaces the tag with information from the models, which can be the content of a code template; (4) it returns the instantiated template as its output, which is the code of the concepts. In this way, changing the target platform for a given design only

requires using different PM (i.e. code templates) and changing the attached code components.
Note that though Fig. 2. shows a sequence of activities, a true development needs to carry out several iterations of these activities. For instance, engineers can discover when they are developing their PSM in activity 9 that some teams are missed, and they will need to return to activities 2-7. Moreover, activities 1-7 need further refinement to provide more guidance depending on specific application contexts.

6. Related work

This section compares the proposed approach with related work in sensor networks and MAS. The introduction already discussed different perspectives on the design of sensor networks. This section follows this classification and distinguishes between integral solutions with architectures and partial solutions for specific aspects. Among architectures, there are examples focused on the infrastructure and others on the high-level design of the network. Transversal to these approaches, some researchers have proposed the use of MAS for the development of the related control systems.
Architectures for sensor networks focused on infrastructure provide a platform with basic services for the sensor network. This platform has a component model that those elements to integrate in the network must fulfil. In this group can be included operating systems (e.g. TinyOS (Hill et al., 2000) and Contiki (Dunkels et al., 2000)), programming languages (e.g. nesC (Gay et al., 2003)) and middleware (e.g. MORE (Wolff et al., 2007), RUNES (Costa et al., 2005), SMEPP (Caro-Benito et al., 2009) and Tenet (Gnawali et al., 2006)). These works and ours appear at different levels of abstraction when considering the development of sensor networks and their control systems. According to MDA (Kleppe et al., 2003), the models based on the architecture proposed in this work are PIM that use highly abstract primitives to model sensor networks. These abstract elements are mapped to the constructions available in these implementation platforms. For instance, the concept of team in our architecture can be partially supported in SMEPP (Caro-Benito et al., 2009) with the concept of group, which provides mechanisms for authentication and authorization, communication between agents can be implemented with μSOA messages of MORE (Wolff et al., 2007). The information of these platforms would appear in our approach as PM. Engineers would refine the PIM of our architecture in that provides the information for that specific implementation. This refinement would be partly implemented with automated transformations from PIM to PSM, for instance to create the structure of μSOA messages, and partly manual, for instance the actual content of messages. If required, abstract components of these architectures could appear in the architecture of this work as additional roles and teams. The code generation module of the IDK would generate the code for the control system from the final PSM and PM.
Architectures considering the high-level design of the network have adopted usually the form of guidelines. Either they just give some abstract design principles or they consider also a development process. From the point of view of the design principles, the flexibility of the proposed architecture allows it adopting the principles underlying a variety of these approaches. For instance, carrying out the processing of data as close as possible to their sources (as (Qi et al., 2001) recommends) means that the actors playing the roles of information process teams should run in the same container, and moving that processing to more powerful computational devices (as proposed in (Gnawali et al., 2006)) splits these

actors in different containers. In both cases the design of roles and teams is the same, and only the initialization information actually changes. The proposed architecture is not intended however for mobile agents as those in (Qi et al., 2001; Tong et al., 2003). Actors in the proposed architecture are not able of redeploying in a container different from that where the initializers create them. However, the initializers could be modified to allow this kind of behaviour. It would be enough to allow initializers to collect information about the actor that wants to migrate (e.g. current state, teams or available resources), and send it to the target container where another initializer would use it to create another actor with the same data. Of course, this migration would also demand checking that the resources and managers that the actor needs are available in the target container.

This section has already mentioned works based on agents (Botti et al., 1999; Hla et al., 2008; Jamont, J. P. & Occello, M., 2007; Pavón et al., 2007; Qi et al., 2001), but some of them deserve further discussion given the similarities with our work. (Hla et al., 2008) establishes some guides for the design of MAS for sensor networks and uses some concepts common with our approach, as controllers, sensors and providers. They also consider concepts that our approach can incorporate, such as directory facilitators to refine the location of sensors with certain features. However, these roles are informally defined in terms of their responsibilities and the set is closed. In this sense, our approach with a specific modelling language and the possibility of defining teams facilitates customization. Besides, (Hla et al., 2008) does not consider a development process for control systems. (Botti et al., 1999; Jamont, J. P. & Occello, M., 2007; Pavón et al., 2007) present development process for control systems. ARTIS (Botti et al., 1999) is a methodology for holonic manufacturing systems that includes the use of sensors. It considers aspects of real time, but ignores issues such as limited resources. (Jamont, J. P. & Occello, M., 2007) is tailored for sensor networks and has been validated with real projects. Though it considers automated generation of code, it does not offer a standard process for it, as our approach does with MDE. This makes more difficult reusing available infrastructure for development and reusing the design models of previous projects. (Pavón et al., 2007) deserves special mention as it also considers INGENIAS for the design of sensor networks. As a matter of fact, both approaches represent complementary points of view. The approach proposed in this work extends the modelling language of INGENIAS with new concepts, and establishes patterns and guidelines to address the design of these networks with its architecture. These tasks correspond to activities 1-7 in Fig. 2. Since these models are INGENIAS models, their refinement to the running code can follow any suitable INGENIAS development process. These tasks correspond to activities 8-13 in Fig. 2. In particular, this refinement can follow (Pavón et al., 2007), which is targeted for sensor networks. Thus, these works can be seen as part of an ongoing effort to provide engineers with a tailored methodology and development process for sensor networks.

7. Conclusions

The presented approach is intended to facilitate the high-level design of sensor networks based on MAS. It includes an agent-oriented modelling language with specific extensions and an architecture describing how these elements interact to achieve the standard functionalities of these networks.

The modelling language is built around three main concepts. Resources are the passive elements in the network. They are modelled in terms of their available methods. Their subtypes include sensors and data processing utilities. Sensors add to resources a state and work with events, both perceived from their environment and raised to inform to their controllers. These abstractions cover the most common uses of sensors in previous works. The active elements of the network are designed as roles. Roles are common abstraction in MAS defined in terms of their goals, capabilities to achieve them, and their resources and information. Managers are in our approach the roles governing resources. They can have different access rights in order to organize the use of the resources. The final element of the language is teams, which are hierarchical groups of roles aimed at performing some collaborative activities in the network. Actors running in containers implement the roles.

The architecture works with these concepts to specify teams that define standard aspects of behaviour in these networks. It identifies teams for the initialization or redeployment of containers, the management of data (including collection, processing and integration), communications and load balancing (or adaptation of the network to changes in the environment or its elements).

The proposed solution is intended to be flexible in several ways. First, it allows accommodating new or modified concepts for specific needs through changes in its metamodel. Using model-driven techniques, engineers propagate these changes to the supporting tools. Second, the specialization of concepts with inheritance relationships and the organization of systems around teams cover a variety of approaches, so it allows incorporating existing research in the area. Third, the use of a MDE approach facilitates reusing the knowledge present in the definition of teams. These teams can become the basic building blocks for sensor networks with MAS, as their models can incorporate information for the final code generation. Only models and transformations related with the control of specific sensors and particular manipulations of data need to be replaced in the system. If new teams were required, they could be modelled as extensions of concepts presents in the architecture as done with standard teams.

The main concern in the application of the proposed approach is the difficulty to model the low-level details of sensor networks, such as energy consumption of routing algorithms for messages. At the moment, the only mean to do that is attaching code snippets to entities in design models for the code generation. There are plans to extend the modelling language with additional primitives to describe some low level issues. For instance, methods can be modelled with additional state machines, and certain standard data transformations can be added as instances of the methods of utilities. Moreover, this work has applied the standard INGENIAS development process for part of its process.

8. References

Akyildiz, I. F., Melodia, T., Chowdhury, K. R. (2007) A survey on wireless multimedia sensor networks. Computer Networks 51(4), pp. 921-960.

Baronti, P., Pillai, P., Chook, V. W. C., Chessa, S., Gotta, A., Hu, Y. F. (2007) Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards. Computer Communications 30(7), pp. 1655-1695.

Botti, V., Carrascosa, C., Julián, V., Soler, J. (1999) Modelling agents in hard real-time environments. In Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'99), Lecture Notes in Computer Science 1647, pp. 63-76.

Budinsky, F. (2003) Eclipse Modelling Framework: Developer's Guide. Addison Wesley.

Caro-Benito, R. J., Garrido-Márquez, D., Plaza-Tron, P., Román-Castro, R., Sanz-Martín, N., Serrano-Martín, J. L. (2009) SMEPP: A Secure Middleware For Embedded P2P. In Proceedings of the 2009 ICT Mobile Summit, pp. 1-8.

Casbeer, D. W., Kingston, D. B., Beard, R. W., McLain, T. W. (2006) Cooperative forest fire surveillance using a team of small unmanned air vehicles. International Journal of Systems Science 37(6), pp. 351-360.

Cook, S. (2000) The UML family: Profiles, prefaces and packages. In Proceedings of the 3rd International Conference «UML» – The Unified Modelling Language (UML 2000), Lecture Notes in Computer Science 1939, pp. 255-264.

Costa, P., Coulson, G., Mascolo, C., Picco, G. P., Zachariadis, S. (2005) The RUNES middleware: A reconfigurable component-based approach to networked embedded systems. In Proceedings of the 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'05), vol. 2, pp. 806-810.

Dunkels, A., Gronvall, B., Voigt, T. (2004) Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors. In Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN 2004), pp. 452-462.

Estrin, D., Govindan, R., Heidemann, J., Kumar, S. (1999) Next century challenges: Scalable coordination in sensor networks. In Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 1999), pp. 263-270.

Gay, D., Levis, P., Von Behren, R., Welsh, M., Brewer, E., Culler, D. (2003) The nesC language: A holistic approach to networked embedded systems. In Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation (PLDI 2003), pp. 1-11.

Gnawali, O., Jang, K. Y., Paek, J., Vieira, M., Govindan, R., Greenstein, B., Joki, A., Estrin, D., Kohler, E. (2006) The Tenet architecture for tiered sensor networks. In Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys 2006), ACM Press, pp. 153-166.

Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., Pister, K. (2000) System architecture directions for networked sensors. ACM Sigplan Notices 35(11), pp. 93-104.

Hla, K. H. S., Choi, Y., Park, J. S. (2008) The multi agent system solutions for wireless sensor network applications. In Proceedings of the 2nd KES Symposium on Agent and Multi-Agent Systems – Technologies and Applications (KES-AMSTA 2008), Lecture Notes in Computer Science 4953, pp. 454-463.

Jamont, J. P., Occello, M. (2007) Designing embedded collective systems: The DIAMOND multiagent method. In Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007), pp. 91-94.

Kephart, J. O., Chess, D. M. (2003) The Vision of Autonomic Computing. Computer 36(1), pp. 41-50.

Kleppe, A. G., Warmer, J., Bast, B. (2003) MDA Explained: The Model Driven Architecture - Practice and Promise. Addison-Wesley.

García-Magariño, I., Rougemaille, S., Fuentes-Fernández, R., Migeon, F., Gleizes, M. P., Gómez-Sanz, J. J. (2009) A Tool for Generating Model Transformations By-Example in Multi-Agent Systems. In Proceedings of the 7th International Conference on

Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009), Advances in Soft Computing 55, pp. 70-79.

Lorincz, K., Malan, D. J., Fulford-Jones, T. R. F., Nawoj, A., Clavel, A., Shnayder, V., Mainland, G., Welsh, M., Moulton, S. (2004) Sensor Networks for Emergency Response: Challenges and Opportunities. IEEE Pervasive Computing 3(4), pp. 16-23.

Object Management Group (OMG) (2006) Meta Object Facility (MOF), Core Specification, Version 2.0. January 2006, http://www.omg.org

Object Management Group (OMG) (2009) OMG Unified Modeling Language (OMG UML), Superstructure, V2.2. February 2009, http://www.omg.org

Pavón, J., Gómez-Sanz, J., Fernández-Caballero, A., Valencia-Jiménez, J. J. (2007) Development of intelligent multisensor surveillance systems with agents. Robotics and Autonomous Systems 55(12), pp. 892-903.

Pavón, J., Gómez-Sanz, J. J., Fuentes, R. (2005). The INGENIAS Methodology and Tools. In Henderson-Sellers, B., Giorgini, P. (eds.). Agent-Oriented Methodologies, Idea Group Publishing, chapter IX, pp. 236–276.

Qi, H., Iyengar, S. S., Chakrabarty, K. (2001) Multiresolution data integration using mobile agents in distributed sensor networks. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 31(3), pp. 383-391.

Schmidt, D. C. (2006) Model-Driven Engineering. IEEE Computer 39(2), pp. 25-31.

Sendall, S., Kozaczynski, W. (2003) Model Transformation: The Heart and Soul of Model-Driven Software Development. IEEE Software 20(5), pp. 42-45.

Son, B., Her, Y., Kim, J. G. (2006) A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea mountains. International Journal of Computer Science and Network Security 6(9B), pp. 124-130.

Tong, L., Zhao, Q., Adireddy, S. (2003) Sensor networks with mobile agents. In Proceedings of the IEEE Military Communications Conference (MILCOM 2003), vol. 1, pp. 688-693.

Tubaishat, M., Madria, S. (2003) Sensor networks: an overview. IEEE Potentials 22(2), pp. 20-23.

Vinyals, M., Rodriguez-Aguilar, J. A., Cerquides, J. (2008) A survey on sensor networks from a multi-agent perspective. In Proceedings of the 2nd International Workshop on Agent Technology for Sensor Networks (ATSN 2008), pp. 1-8.

Weiss, G. (ed.) (2000) Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press.

Wolff, A., Michaelis, S., Schmutzler, J., Wietfeld, C. (2007) Network-centric Middleware for Service Oriented Architectures across Heterogeneous Embedded Systems. In Proceedings of the 11th International IEEE Enterprise Distributed Object Computing Conference (EDOC 2007), Middleware for Web Services Workshop, pp. 105-108.

Yick, J., Mukherjee, B., Ghosal, D. (2008) Wireless sensor network survey. Computer Networks 52(12), pp. 2292-2330.

3

Modeling Artificial Life Through Multi-Agent Based Simulation

Terry Lima Ruas, Maria das Graças Bruno Marietto, André Filipe de Moraes Batista, Robson dos Santos França, Alexandre Heideker, Emerson Aguiar Noronha and Fábio Aragão da Silva
Center of Mathematics, Computation and Cognition, Federal University of ABC
Brazil

1. Introduction

The Multi-Agent Based Simulation (MABS) area is placed at the intersection of two distinct areas: Distributed Artificial Intelligence (DAI) and Computational Simulation. This field of research provides a proper infrastructure for modeling and understanding the processes related to social interactions such as coordination, cooperation, training and coalition of groups and resolutions of conflicts, among others. Such understanding is made possible because of the relationship established between local and global behavior, which leads to leading to explicit chains of cause and effect of how internal agent components affect the agents behavior, how this behavior affects the agency and, dialectically, how the agency affect its agents components. Multi-agent simulation models are based on the concept of the individual-program relationship, which allows the simulation of artificial worlds where each entity (interactive computing entity) is represented as an agent that maps a single entity (or a group of them) in the target system. Since the infrastructure of technical and theoretical areas of simulation allows researchers to mimic the essential elements of a target system without having to work directly with the target system itself, it becomes handful when dealing with phenomena such as the spread of fire without hazarding the integrity of the environment and its living beings.

Artificial Life in Computers yields the creation of a laboratory capable of providing the necessary means to study, research, reproduce and maximize the simulations on a specific subject. As stated previously, a simulation model is a particular type of model that aims to represent a given system. However, it differs from classical models in the sense that it facilitates a) the study of how the modeled systems behave under certain conditions, and b) the examination, in varying degrees of detail, the consequences of changing internal behaviors of the system, and vice versa.

The results obtained in a simulation might be of great help in the decision-making process, in the evaluation of systems and in reducing implementation time and costs.

In (Ferber, 1996; Gilbert & Troitzsch, 1999) some simulation goals are presented, namely:

- Discover and formalize new theories and models;
- Develop a better understanding of some features of the real system;
- Test hypotheses of the modeled system;

- Predict future actions and behaviors.

More specifically, (Ferber, 1996) defines that an agent-based simulation model relates to the idea that a system is comprised of all relationships of its inner parts, and in that sense, it is possible to simulate an artificial world based on the relationships of its entities.
The simulation occurs when there is a transposition of the population of a target system [1] to a *conceptual model* equivalent, followed by the encoding of this model to a *computational model*. In this case, an agent (or actor) equates to a real world entity or a group of them. Such actors can be of different natures and with various granularities, such as humans, robots, computer programs, inanimate objects and organizations.
After the establishment of the multi-agent paradigm in the computer science, the role of multi-agent based simulation has been acquiring relevance in a variety of scientific disciplines. In particular, the sources of analogy between agent-based technologies and models of actual social systems, and the efforts towards dealing with such complex systems through simulation models have created this intense interdisciplinary effort that provided ground for the advent of a new scientific field, named Multi-Agent Based Simulation (MABS). As a result, research interfaces were created across various disciplines under the umbrella of a multidisciplinary area that involves researchers from as diverse fields of study as Psychology, Sociology, Economy and Computer Science.
Considering the relatively recent advent of MABS, its multidisciplinary aspect might also pose as one of the biggest challenges to be overcome by researchers, since it requires cutting across traditional boundaries of school of thoughts, mixing different theories, methodologies, techniques and point of views. In this chapter, the principles of multi-agent based simulation are presented, as well as some simulations that exemplify the integration of MABS and artificial life. To accomplish that, the chapter is divided in three main parts: the first part focus on the presentation of MABS concepts and techniques. The second part presents some of the main simulation platforms and frameworks available today and also analyses and compares two of them. The third and final part displays a set of models that aim to simulate artificial life though the use of MABS techniques.

2. Principles of Multi-Agent Based Simulation

The main goal of the Multi-Agent Based Simulation (MABS) researchers is to develop and study simulation models taking into consideration a theoretical-technical framework based on the Distributed Artificial Intelligence field. The general relevance of simulation, and more specifically agent-based simulation becomes so clear that some authors have gone far enough to consider it as a third way of doing science, along with traditional deduction and induction reasoning (Axelrod, 1998). It could be stated that simulation distinguishes from standard deduction and induction in its implementation and also in its goals. A simulation starts with a set of explicit assumptions (as in deduction) but not generally providing any theorems, producing data which are suitable for analysis by induction that come from a strictly set of assumptions.
Following that perspective, a simulation model is a kind of model that represents a specific target system. What makes this model distinct from the others is (i) the chance of studying the global behavior of the modeled system in certain conditions and (ii) the possibility to inspect the consequences of changes in the internal components of the system. An important aspect to be considered in simulation systems is the assurance that both conceptual and

[1] The target system is equivalent to the simulation domain and can be real or theoretical.

computational models accurately represent the target system, and that can be achieved by using two processes: validation and verification.

The validation process aims to certify that the conceptual model represents the target system in an acceptable degree of adherence. Thus, the validation processes fundamentally addresses a specific question: Does the simulation outcomes correspond to those from the target system? On the other hand, the verification process' main purpose is to assure that the conceptual model was correctly translated to the computational environment. Specifically, a multi-agent simulation model is based on the concept that it is feasible to simulate an artificial world inhabited by interactive computational entities. Such simulation can be achieved by transposing the population from a target system to its artificial counterpart. In that sense, an agent is similar to an entity or a group of entities of the target system. Moreover, agents can be of distinct natures and granularities, such as human beings, robots, computer algorithms, inanimate objects and organizations.

In a multi-agent based simulation, the relationship between agents can be specified in several ways, ranging from reactive to deliberative attitude approaches. In both cases, agents must be able to decide and perform their actions autonomously. Nevertheless, to ensure a proper execution of the simulation, agents actions must be synchronized through the scheduling of a minimum set of events, and their behavior can be either time-stepped scheduled - performed within each discrete time step - or event-driven scheduled, in which agent actions are scheduled by other agents' actions and/or events.

The MABS area provides a suitable infrastructure to model, study and understand the processes related to complex social interactions such as coordination, collaboration, group formation, evolution dynamics of norms and conventions, free will and conflict resolution, among others. That can be achieved by relating local and global behavior and analyzing how agents can affect the environment and other agents (and vice-versa, leading to explicit chains of cause and effect), how internal agent components affect the agent's behavior, how this behavior affects the agency and, dialectically, how the agency affects its agents components (Gilbert & Troitzsch, 1999).

2.1 Multi-agent modeling

Multi-agent simulations require the development of multi-agent models, which aims to model complex real-world systems as dynamical systems comprised of interacting autonomous decision-making entities called agents. Traditional analytical methods might not be suitable to deal with complex phenomena that are simply too complicated to be analytically tractable, especially when involving non-linear relationships. Multi-agent models have then emerged as an alternative for these types of problems. In recent scientific literature, many denominations for agent-based modeling can be found, such as: Individual-Based Modeling (IBM), Agent-Based Systems (ABS), among others.

An agent-based model is essentially a population of heterogeneous agents, which represents autonomous entities that interacts between themselves and with their environment, allowing the formation of a social system where aggregated structures (patterns) emerge from those interactions. The fundamental principle of an agent-based model is the emergence of social structures and groups of behaviors from the interactions of individual agents. These agents operate in artificial environments and under specific rules that are valid only when taking into account the limitations of each agent regarding their own computational and memory capabilities.

In Table 1 a comparison between a traditional and agent-based modeling is presented (each of the aspects is explored in the subsequent sections of this chapter).

Traditional	Agent-Based
Focus on continuous time	Focus on discrete time
Mathematical language (equations)	Descriptive model
Aggregate level granularity	Individual level granularity
Top-down (macro-to-micro) approach	Bottom-up (micro-to-macro) approach
Pre-defined behavior	Emergent behavior
Global control	Local control

Table 1. Comparison Between Traditional and Agent-Based Modeling.

2.2 Main aspects of multi-agent models

Simulation models based on multi-agents are comprised of a number of heterogeneous agents, relationships between these agents and an environment capable of simulating the behavior and interactions of such agents. Also, there is no central authority in charge, as agents are modeled to behave autonomously in a self-organized model based on simple local rules of interactions between agents and the environment. The ultimate goal of such model is to allow the emergence of system-level phenomena resulting from these local interactions between agents themselves and the environment.

A more specific definition of agent would be of a discreet entity with its own objectives and behaviors. Each agent contains internal states and behavior rules, allowing them to interact with other agents and the surrounding environment. Agents are also autonomous and display some degree of initiative, allowing them to behave as object-oriented entities. They are modeled to execute the vast majority of their actions without any direct interference from either humans or other computational agents. Examples of agents include people, groups, organizations, social insects, swarms, robots, and so forth.

2.2.1 Ascending (bottom-up) modeling

Agent-based models are built from agents that have very simple rules defined for their behavior. The interactions between these agents create collective structures in an ascending approach instead of a descending one, where the macro structures and behavior of a system would be modeled and then used to explain micro interactions of its components. Modeling a complex system using a top-down approach would prove much too complex and not appropriate as a complex system behavior is the result of a large number of interactions. An analytical/reductionist approach is also not adequate for modeling complex systems as it assumes that the system behavior can be understood by analyzing its parts separately.

So a bottom-up model is therefore more suitable for complex systems such as the ones applied to artificial life simulation, as the bottom-up approach focus instead on simple rules of behavior for small parts of a system - its agents - and how they interact with each other, making use of computational power to simulate a large number of those agents and their interactions, allowing emergent patterns to be observed and studied. The model can then be easily manipulated in terms of addition or removal of its micro-level individual properties and how these changes might affect the macro-level social phenomena. For instance, a bottom-up model for an ant colony would describe ants in a micro-level and in terms of their behavior as individuals in the colony and how they communicate to each other. A simulation tool would then be used to mimic the colony environment where several individual ants are put to communicate and perform tasks, allowing an observer to study the emergence of colony-level social phenomena.

2.2.2 Complex systems

The word "complexity" has roots in two Latin words: "complexus" which means "totality" and "completere" which means "to embrace". So complex system are formed by two or more interlinked components that creates a network of objects interacting with each other, displaying a dynamic and aggregated behavior. In this context, the complex adjective is not to be confused with complicated. Moreover, in a complex system the action of a single element might affect the actions of other objects in the network, making the famous paradigm 'The whole is more than the sum of its parts' even more true.

In fact, complex systems are made of several simple behavioral units that influence each other mutually in an intricate network of connections that ultimately generates a global complex behavior. As a result of such a systematic behavior, many properties of a complex system can only be observed during its collective behavior and cannot be identified in any of its fundamental units. One example of a complex system is the fire propagation phenomenon. An adequate (non-analytical) approach to treat complex systems, more specifically the fire propagation phenomenon, is to use simulation techniques based on Cellular Automata (CA). Simulating complex systems allows researchers to (a) propose new structures or alternatives to treat social systems, studying and understanding their existence and operation; (b) have a better understanding of the social, anthropological, psychological aspects, etc. used to describe and explain the analyzed phenomena and (c) to use existing theoretical models already proven effective when dealing with institutional and social processes.

2.2.3 Unpredictable systems

Unpredictable systems are complex systems with a high degree of instability and unpredictability in the decision-making process, and such aspects need to be treated in a dynamic manner. According to (Lempert, 2002), agent-based models are often useful under conditions of deep uncertainty where reliable prediction of the future is not possible by either in a best estimate or probabilistic approaches (such as the ones in traditional simulation models).

In his work, (Lempert, 2002) argues that agent based models are useful at describing the behavior of inherently unpredictable systems. According to him, the predictive policy analysis is an example of application of agent-based simulations, as police simulators may be effective in situations where the standard methods of predictive policy analysis are least effective. Also, in dynamic and unpredictable systems, agents must be modeled in a way that their deliberation and responsiveness are balanced so that they act appropriately. This must be done to avoid long deliberations that might impact the performance of the simulation but also to avoid agents to become too reactive to choose the best action to execute.

2.2.4 Emergent behavior

According to (Axelrod, 1998), 'emergent properties' of a system can be described as the large-scale effects of locally interacting agents, noticed as non self-evident, stable macroscopic patters arising from individual agent's local rules of interaction. Below is a non-exhaustive list of situations when agent-based models are useful for capturing emergent behavior:

1. The interactions between agents are discontinued, nonlinear. This can be particularly useful when describing complex individual behavioral. Discontinuity proves much too complex by using traditional analytical methods (for instance, differential equations);
2. There is a significant necessity of designing a heterogeneous population of agents. The heterogeneity allows agents with clearly distinct rationality and behavior to be modeled;
3. The topology of the agent's interactions is complex and heterogeneous. This can be particularly useful when modeling social processes, specially the inherent complexity of physical and social networks.

Emergent phenomena can also be formalized as requiring new categories to describe them, which are not necessary to describe the behavior of the model's underlying components (i.e. agents) (Gilbert & Terna, 2000). In some models, the emergent properties can be formally deduced, but they can also be unpredictable and unexpected, as anticipating the consequences of even simple local interactions sometimes proves to be a hard task. Also, according to (Axelrod, 1998), an example of emergent phenomenon can be seen in a model where agents represent consumers and have local behavior rules that allow them to choose and buy brands of video tapes according to the availability of machines on which to play it. Only by analyzing the agent's local rules, one would not intuitively notice that the simulation model is most likely to lead one format to completely overcome the other.

Moreover, mathematical analysis might be limited in its ability to derive the dynamic consequences in models where, for instance, agents have an adaptive behavior influenced by their past experience. For this type of situation, a simulation model is usually the only feasible method.

2.2.5 Open systems and self-organization

Self-organization is a process where the organization of a system is not guided or managed by an outside source. Self-organizing systems normally represent open systems and might typically display emergent properties. Open Systems in turn can be described as a system with high environmental adaptability through quick incorporation of new elements, information and ideas. On the other hand, a closed system resists the incorporation of new ideas and risks atrophy, ceasing to properly serve the environment it lives in.

Self-organization is considered an effective approach for modeling the complexity in modern systems, allowing the development of systems with complex dynamics and adaptable to environmental perturbations without complete knowledge of future conditions. According to (Gardelli et al., 2008), "*The self-organization approach promotes the development of simple entities that, by locally interacting with others sharing the same environment, collectively produce the target global patterns and dynamics by emergence. Many biological systems can be modeled using a self-organization approach*".

Some examples of self-organizing environments include food foraging in ant colonies, nest building in termite societies, the comb pattern in honeybees, brood sorting in ants, decentralized coordination for automated guided vehicles, congestion avoidance in circuit-switched telecommunication networks, manufacturing scheduling and control for vehicle painting and self-organizing peer-to-peer infrastructures, among others.

2.2.6 Local and global control

Governing laws of behavior for individual agents in a multi-agent simulation model can be implemented as either local, global or a combination of both. The decision depends on the type of simulation being modeled and the constraints imposed by the problem being solved.

According to (Parker, 1992), emergence might occur solely based on the interaction of the local control laws of the individual agents, which might not be aware of any global goals. However, this approach might not be sufficient to model some simulations where agents are expected to cooperate towards a global goal, and a hybrid model with both local and global control might offer a solution. Still according to (Parker, 1992), the key difficulty when designing control laws governing the behavior of individual agents is to find the proper balance between local and global control and how to design such controls: as global goals designed into the agents, as global or local knowledge or through a behavioral analysis method.

3. Multi-agent based simulation platforms

This section presents two frameworks that aid in the building and execution of SMA's: Swarm (Group, 2009) and Mason (Cioffi-Revilla & Rouleau, 2010). In this context, the main aspects of each platform are covered in order to establish a comparative overview among these platforms. Although the scope of this section is limited to these two platforms, it is worth emphasizing that there are many others multi-agent frameworks being used by scientists, for example Repast, NetLogo, and etc.

3.1 The Swarm Platform

The Swarm Platform was created in 1994 at the *Santa Fe Institute, USA* by Christopher Langton with the help of other researchers. It was written in *Objective C*, but later a *Java* interface was also developed. The Swarm Platform offers multi-agent researchers a good variety of resources such as memory management, action scheduling, graph generation, real-time simulation updating/interference, etc.

A Swarm can be described as a type of animal behavior characterized by the reunion of many similar entities that together seem to behave as a bigger, single organism, such as a school of fishes swimming at the sea or a swarm of bees flying in the sky. This type of behavior displays a noticeable degree of flexibility (a swarm of insects adapting to environmental changes such as the wind, rain, smoke, etc.), robustness (a global objective will still be pursued - and most likely achieved - even if some of the members of the swarm are lost during the execution of the task), decentralization (there is no central control as in a fish shoal) and self-organization (insects in a swarm will organize themselves to achieve a global objective).

Following that philosophy, the Swarm Platform was developed to allow the mimic of such features and concepts, modeling the agents with reactive features and actions. A second feature provided by this platform is the creation of hierarchical models. In other words, it could be possible to design multi-agent simulations which the agents are composed by other agents, forming a multi-agent simulation by itself, or a simulation of nested simulations. This allows the formation of systems with a high level of complexity.

3.1.1 The Swarm Platform architecture

The basic component that organizes agents in the Swarm platform is called SWARM. A SWARM can be described as a collection of agents under a schedule of events and represents the entire model, as it contains all agents within then model as well as the representation of time. The basic architecture of a swarm simulation is comprised of a MODELSWARM, an OBSERVERSWARM and, optionally, simulation PROBES. Figure 1 displays the basic architecture of a simulation in the Swarm Platform.

The MODELSWARM contains the conceptual model implementation, and is comprised by a SWARM and optional sub-swarms. In this architecture, active and passive agents are defined

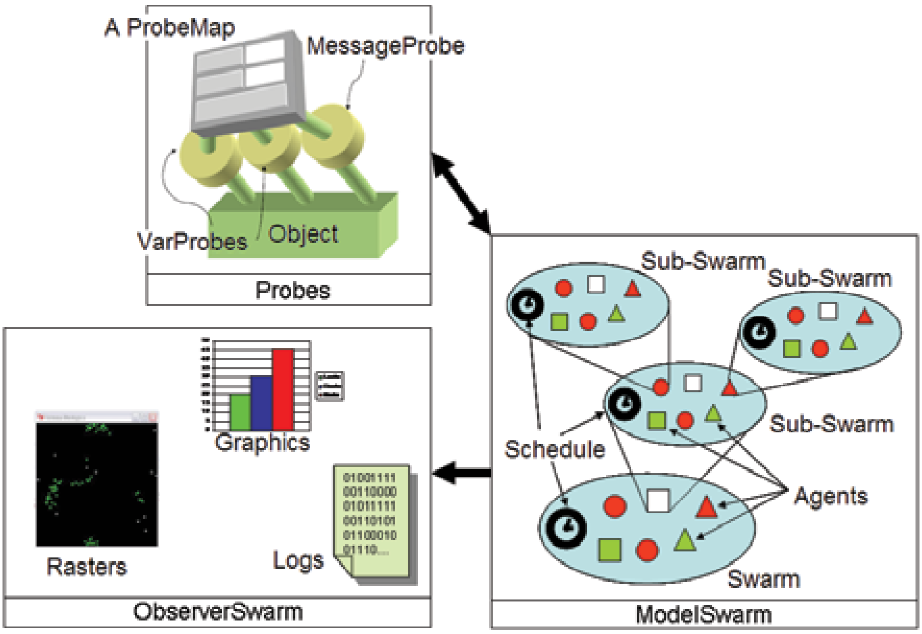


Fig. 1. Basic Architecture of a Simulation in the Swarm Platform.

along with their structures, features and their influential environment. The OBSERVERSWARM is responsible for collecting the simulation data and the consequent presentation of such data through charts, animations, files, etc. Finally, optional PROBES can be implemented so the user can interact with the simulation in real time by observing and changing variables, agent states and so forth.

Another essential element in a Swarm simulation is the SCHEDULER, which is responsible for synchronizing the actions of agents of each swarm. Only one scheduler is allowed for each swarm in a Swarm simulation

3.2 The Mason platform

The "Multi-Agent Simulator Of Neighborhoods" (MASON) Platform is a multi-agent simulation framework written in Java, comprised of a model library and a set of 2D and 3D visualization tools. It is a result of a coordinated effort between The Evolutionary Computation Laboratory and The Center for Social Complexity at George Mason University. Among its characteristics, it is worth mentioning:

- Models are completely independent from their visualization;
- Portability between different platforms allowing the production of identical results;
- Native generation of simulation snapshots and movies from the simulation data.

Figure 2 shows the basic architecture of the Mason platform.

Just as the Swarm Platform, the Mason Platform contains a scheduler that allows the simulation of discrete-time events. However, a significant difference between the two schedulers is that in Mason the scheduler schedules agents instead of events, while in Swarm the events themselves are the ones scheduled by the scheduler.

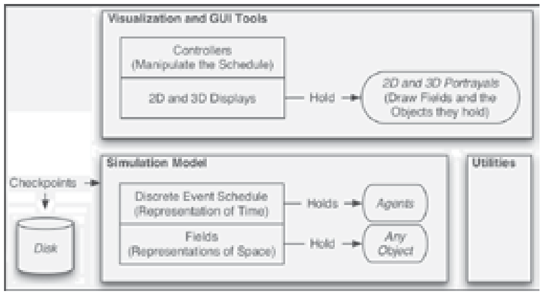


Fig. 2. Basic Architecture of MASON.

4. Integrating artificial life and Multi-Agent Simulation: A new approach to making science

Most of social science research work follows two approaches. In a theoretical approach, a theory is chosen and it is used as a foundation for the whole research, which develops that theory itself or proposes the creation of new ones derived from that primary theory. On the other hand, a practical approach proposes the building of a model that could be represented by equations, following an empirical line of work. Both approaches look into the social phenomena using a different pair of glasses, although both glasses were made by the same manufacturer (Social Science). Also, this shows how some social scientists can be. Anything out of the theoretical or empirical approach might be seen as far-fetched and does not fit into Social Science. In some cases, it could even be treated as "no-science" at all.

A third way to represent and analyze the social phenomena translates the social theories to computer programs and presents itself as an alternative scientific method. According to this method, artificial societies can be perceived as social laboratories that could be used to analyze and test the social theories (Axelrod, 1998). This section's goal is to set forth this third way of doing science.

Generally speaking, simulations (especially multi-agent based simulations) are becoming more relevant in such ways that some researchers consider that they are a "third way to make science", along with induction and deduction. Induction attempts to analyze the micro-elements and to extrapolate such analysis in order to describe the global scenario (micro to macro), which implies a deep understanding of each individual element. Deduction uses the macro structures and global concepts as a guide to observe and to describe the micro-elements. This method takes the opposite direction of induction; instead of describing individual behaviors, the global behaviors are described in detail and they are employed as rules that must be followed by all elements.

On the other hand, the way of how multi-agent based simulations deal with their goals and their development may explain why they are considered a new form of science. At first, simulations start with high level, explicit premises (as in the deductive process) but not applying proofing theorems. Also, simulations provide effective results for an inductive analysis based on well-defined premises. That way, it could be said that simulations gather some of the best aspects of both classical scientific methods of reasoning in order to fulfill their goals, while keeping a logical coherence. Finally, due to the complex nature of the social phenomena and the appliance of distributed properties in that field, multi-agent based simulations can fit well in both demands by design (Goldspink, 2002).

Besides the integration of these two classical approaches, the multi-agent based simulations provide: (i) a conceptual model that describes the system to be simulated according to theories and general concepts, (ii) a computation model which, after being implemented, could be analyzed empirically in a safe and controlled environment, (iii) tools for improving and revising the theories used for building the model, (iv) new ways of looking into the social theories using, for example, artificial worlds or a distinct set of rules for a real case scenario that would never be feasible in a physical experiment.

Thanks to the multi-agent based simulations, a new family of multidisciplinary projects is coming. Social theorists, mathematicians, computing specialists, biologists, linguists and many other professionals are working together to model and to improve the social simulations.

4.1 Artificial life and MABS in practice

At the following sections, a general overview of the research work of the Artificial and Social Intelligence Group at Center of Mathematics, Computation and Cognition, Federal University of ABC (UFABC) is presented to exemplify how multi-agent simulations can be used to model artificial life systems. The applications are classified into groups according to the following: Group I (basic applications), Group II (focus on information exchange) and Group III (advanced models). Such division is performed to provide a better understanding of the MABS' potential.

4.1.1 Group I: Basic applications

Two basic applications are well-known in the MABS area: Conway's Game of Life and the Prey-Predator Model. Such applications provide the basis for several other MABS applications, as they successfully represent most basic MABS concepts, such as emergency, self-organization, bottom-up modeling, etc.

4.1.1.1 Conway's game of life

A Cellular Automata (CA) is a set of cells disposed in a bi-dimensional grid where each cell might assume a certain state, with the number of possible states being finite. All of the Cellular Automata cells will simultaneously evolve to the next generation according to the same set of evolution rules.

The Game of Life was created in 1970 by a British mathematician called John Conway, which extended the work of John von Neumann. Having simplicity as the guiding rule for his work, Conway achieved impressive results. Contrary to the work of John von Neumann, which was comprised by a large number of rules, Conway's Game of Life is comprised by few simple rules. In an orthogonal grid, each cell can assume two states, alive or dead. During each turn (or time-step), the automata determine whether each cell will be alive or dead based on four simple rules, as follows:

- any live cell with fewer than two live neighbors dies by loneliness;
- any live cell with more than three live neighbors dies by overpopulation;
- any dead cell with exactly three live neighbors becomes alive;
- any live cell with two or three live neighbors survives.

Figure 3(A) shows the execution of the first rule, where isolated cells die by loneliness. Figure 3(B) shows the execution of the second rule, where a cell with more than three live neighbors dies by overpopulation. Third rule can be seen in Figure 3(C), where each cell with exactly three live neighbors becomes alive.

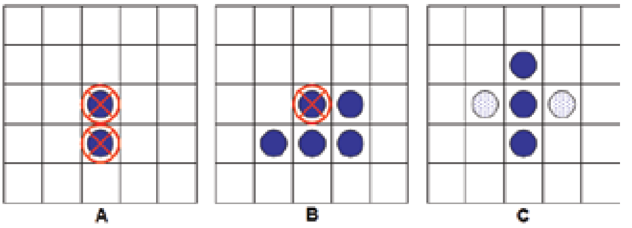


Fig. 3. Examples of rules execution in a Conway's Game of Life.

The results that turned the Game of Life an example of complex system are still a subject of research. These four simple rules are capable of generating a broad range of patterns, from static ones to complex arrangements that will look and behave as "creatures" that seem to walk through the grid, many times destroying other clusters of live cells or originating new creatures. Figure 4 shows an example of static pattern generated by such rules. Figure 5 shows a classic example of a formation known as "Pulsar", where the cluster of cells will alternate between the three states shown on each step of the simulation.

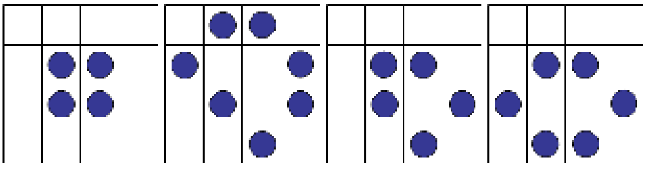


Fig. 4. An Example of a Static Pattern.

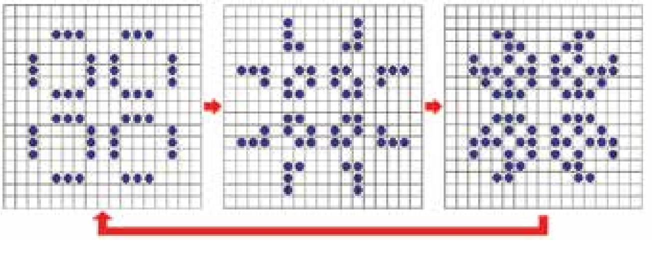


Fig. 5. An Example of a Pulsar Creature and the Corresponding States.

4.1.1.2 The Prey-Predator model

There are many examples of self-sustained, complex systems in nature. The Food Chain is one of such systems, and it can be described as a hierarchy of consumers. An herbivore eats grass and, in turn, a carnivore eats herbivores. Should something happen to this chain, there might be a minor shift in one of the chain members' population which could lead to a major shift in higher level chain members. Therefore, the Food Chain system can be understood as a complex system since the overall behavior of that system relies on how each element interact with each other, and not by the individual behaviors alone (Engbert & Drepper, 1994).
The Predator-Prey Model - modeled after the Food Chain system - is a prime example of a heterogeneous system: there are populations (groups) of distinct species. Some of the behaviors found in these species are the same, such as moving and escaping. However, some species must deal with the hunt of preys and the escape from predators. The interaction between predators and preys makes this system complex, in spite of the simple behaviors

displayed by each element. The purpose of the simulation is to model an environment where there are preys that attempt to run away from their predators while looking for food, and predators that are hunting for these preys. This kind of simulation addresses both populations and hazardous elements that could decrease them, as well as special elements (such as Cheese) that could increase them.

Environment settings

The environment is described as a bi-dimensional grid with 50 rows by 50 columns, and each cell can hold a numeric value. "0" stands for an empty space. "4" indicates a slice of cheese. Values 1,2 and 3 represent a dog, a cat and a rat respectively.

Agents

At first, each agent fights for its survival, hunting for preys and running away from predators, and performs the following actions at each simulation step:

Dog The DOG tries to notice a CAT. If there are no CATS close to it, a random movement is performed. If there is a CAT nearby, the next step is to establish whether the DOG hit the CAT. If it did, the CAT is killed. Otherwise, the newfound CAT is pursued by the DOG.

Cat The CAT follows a similar behavior to the DOG. In other words, the CAT looks for a RAT nearby and, if a RAT is noticed, the CAT starts pursuing it. Likewise, if the RAT is hit, it is killed. However, a runway element is added to the CAT: if a DOG is nearby, the CAT behaves like a pray, running away from the DOG. If there is neither a DOG nor a RAT nearby, a random movement is performed.

Rat

Essentially, the RAT runs away from the CAT and tries to follow the biggest Cheese gradient. If a Cheese is hit, the RAT eats it, and in the place where the Cheese was found, a new RAT is born. If there is neither a clear major Cheese gradient (one that represents the RAT's direction change) nor a CAT nearby, the RAT will perform a random movement.

Standard actions

There are some standard actions for all agents. For these actions, the agents' viewing area is a two-cell Von Neumann's neighborhood, and they can be described as follows:

Notice Agent To search a certain agent.

Follow Agent If a target-agent is close enough, the agent will pursuit it.

Run Agent If a predator is detected, the agent will run in the opposite direction of this dangerous agent.

Hit Agent To identify an agent in a one-cell Von Neumann neighborhood.

Kill Agent If a target-agent has been hit, it must be killed, which means that its cell must be clean (get a zero value).

Random Movement If there is no target-agent nor predator is noticed or hit, the agent will move randomly in one of the eight possible directions (N,NE,NW,S,SE,SW,E,W).

Follow Major Gradient Only available for the rats, it represents a linear search where the largest concentration of cheese in the environment can be found.

The Figure 6 shows the execution of the predator-prey model using the Swarm Framework. The yellow dots represent the cheese, the blue ones the dog agents, the red ones the cats and the white ones are the rats.

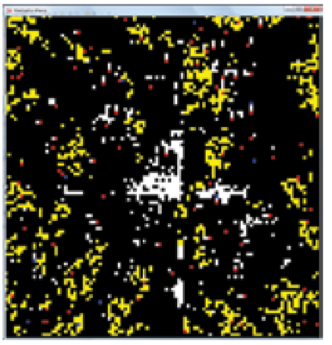


Fig. 6. The Predator-Prey Model Simulation Running.

4.1.2 Group II: Applications focused on information exchange and coordinated actions

This section presents simulations that are focused on the communication and exchange of messages among agents in order to obtain a common objective.

4.1.2.1 Exploring Robots

The resolution of search problems represents a vast research field in AI, influencing the Robotics area as well. Movement planning, execution and the recognition of an unknown territory allows autonomous devices to successfully perform such tasks in highly dynamic environments. In face of that, let's consider an unknown environment (such as a maze or a room with walls) where a group of robots share information while creating a collective memory, in order to map the environment so that an exit can be found. Besides detecting and mapping walls, the robots are supposed to navigate thought the environment avoiding collisions and using routes not visited before while determining the best escape path. Below, an overview of the model and the simulation that represent the aforementioned situation is presented.

Conceptual model

The model of a simulated exploring robot will be implemented using the Swarm platform, where several independent agents build their own maps using a simulated computational vision system. The idea is to make use of technologies such as computational vision, path finding, maps creation, etc. The basis for this type of implementation is an already discretized model, containing walls and at least one exit that will be the agent's main goal. The premises for the creation of the conceptual model are as follows:

- The agent simulates a robot, thus is important to consider its characteristics and limitations;
- The robot is equipped with four sensors which are capable of detecting any front, lateral or rear obstacle;
- Walls detected by such sensors are automatically added to the map;
- A single map is shared among all robots, which means that robots collaborate on building a collective representation of the environment;
- The map can be trusted, which means that the correctness of the information provided by the robots is assured;
- Routes already mapped must be avoided as unknown routes are prioritized;

- Whenever enough information to find the exit is available, the most efficient algorithms must be used.

The map and the environment

Figure 7 shows an example of environment used in the simulation proposed by this study. At the beginning of the simulation, the environment is loaded by the simulator and robots are scattered randomly.

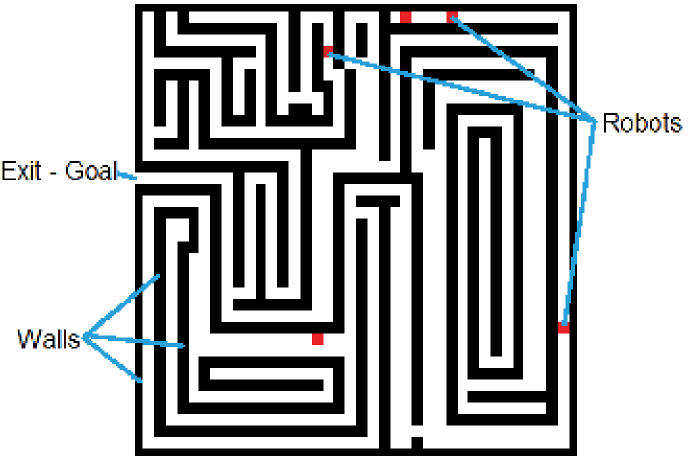
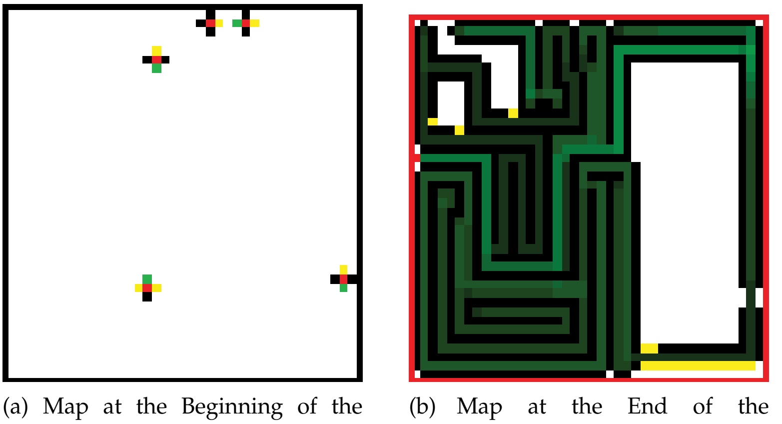


Fig. 7. Example of a Simulation Environment for Exploring Robots.

The map is a bi-dimensional representation implemented in Swarm using a structure known as Discrete2dImpl, a type of matrix where each position holds an integer. The meaning of such number is given by the following pattern: 0 - no information, 1 - identified wall, 2 - identified free pathway, 3 to 15 - path already taken (incremented in each step) and 50 - exit.
Figure 8(a) shows the map at the beginning of the simulation. Figure 8(b) shows the map at the end of the simulation, when robots have found the exit.



(a) Map at the Beginning of the Simulation.

(b) Map at the End of the Simulation.

Fig. 8. Map at Both Beginning (a) and End (b) of the Simulation.

The robots

The agent that represents a robot is capable of moving towards four directions (left, right, forward and backward) as well as detecting obstacles in any of these directions. On each movement, the agent evaluates its sensor's information as well as its last movement to determine its next step. First, the agent will check to which directions a movement is possible through the detection of walls and other obstacles. Such information is added to the map, which is updated with information regarding the location of walls and free spaces. In possession of that data, the agent will try to infer the following:

- Any position occupied by a wall is immediately discarded from the list of possible movements;
- Any position occupied by another agent is also immediately discarded from the list of possible movements;

The choice of a new position among those available will be based on the priorities below, according to the following order:

- Exit the environment;
- Movement's continuity, which means that the agent will prefer to follow the same direction as the one taken in its last movement, as changing the direction will most likely incur cost;
- Never visited locations;
- Least visited locations;

At the moment an agent occupies a new position, the related point in the map is incremented indicating that a new visit was made. In the next inference, such position will be despised by all agents.

Search algorithm

Considering the heuristics discussed so far, the search algorithm used might be classified as a blind search algorithm, which is fairly inefficient (Russell & Norvig, 2004). However, as the map doesn't provide broad range information (the robot is capable of detecting only whatever lies in its surroundings), it is impossible to make use of more sophisticated algorithms. On the other hand, exploring algorithms greatly favor the exploration and mapping of the environment.

Whenever enough information is gathered by one or many agents to link their position to the map's exit, the A^* (read as Star) algorithm is used to get the shortest path to the exit. The A^* search algorithm is one of the most common solutions to finding the shortest path in a graph (Russell & Norvig, 2004).

In the context of an exploring robot simulation, the graph used is obtained by discretizing the environment. Figure 9 shows how such discretization is performed and the resulting graph. The weight of the edges is 1, and the total weight of the path from the robot's current position to the exit is the same as the number of cells (or vertexes) visited during the walk.

4.1.2.2 Autonomous Robot Navigation Through Fuzzy Logic

The example presented in 4.1.2.1 shows the navigation in a discretized Cartesian environment where well-defined geometric basis are established. The autonomous navigation in real environments finds application from automated machine movement in factories to the creation of partially or completely automated robots used in outer space exploration. Such

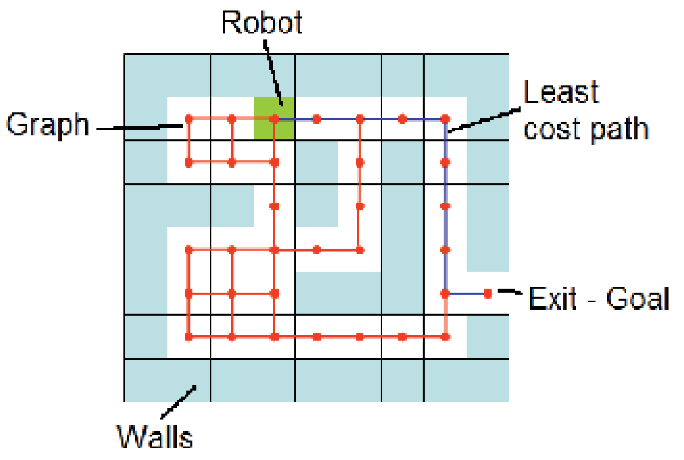


Fig. 9. Discretized Environment and the Resulting Graph.

broad range of applications also includes automating vehicles navigation, which is the focus of the study presented below. In this context, it is necessary to take into consideration the following:

- How inhospitable is the environment where the vehicle navigates;
- Which variables are involved;
- How to represent such variables and how they influence the movement.

The use of Fuzzy Logic offers a better interpretation of unpredictable environments, where the knowledge of the environment is required on each step for making as much of a natural decision as possible. This type of decision involves concepts like close, far away, fast, slow, slight turn, very slight turn, etc. Such concepts cannot be easily modeled by using classical logic, making Fuzzy Logic a very attractive choice.
Created in the 60's by Lotfi Zadeh, Fuzzy Logic is derived from Classical Logic. Zadeh considered that dealing with ambiguity (as the one caused by imprecise information) is an innate human ability. The application of Fuzzy Logic occurs in three stages:

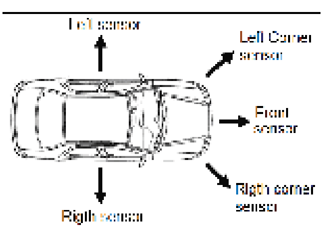
- Fuzzification (to transform values into linguistic variables);
- Logical Inference (to apply logical rules);
- Defuzzification (to transform linguistic variables into values).

Fuzzification is the process of mapping a problem's input values into a function called pertinence function, which represents the parameter's variation and its linguistic counterpart. For instance, if 1.60 meters is considered short and 1.80 meters is considered tall, 1.70 meters might be considered as in-between (not short nor tall, and at the same time kind of short and kind of tall). On the other hand, Defuzzification is the inverse process. The result of logical inference is a value that needs to be translated to a linguistic value. For instance, is a 30-year-old a young, adult or old person?
Using Fuzzy Logic to solve the autonomous navigation problem allows a better modeling of the system's rules, as the ambiguity found when determining if something is far away, close, moving fast or moving slow is much better described by the tools offered by Fuzzy Logic.

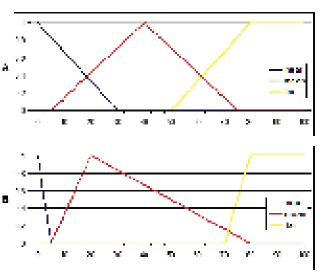
Also, when considering the broad range of choices available in a decisions domain outside a formal logical context (where only true and false exist), the application of Fuzzy Logic is probably the most adequate for solving the aforementioned problem.
There are numerous worth mentioning studies using Fuzzy Logic, such as (Baltes & Otte, 1999) which presents a robot's navigation control system using Fuzzy Logic with heuristic functions, reducing in up to 75% the need of human control for the robots navigation. (Lu & Chuang, 2005) also presents a navigation system which is capable of navigating through obstacles controlling the angle of turn of the robot according to its acceleration. (Farhi & Chervenkov, 2008) shows a design of a fuzzy controller for autonomous vehicle navigation in unknown environments. (Jianjun et al., 2008) presents a practical example of an autonomous navigation system for farm tractors using diffuse logic. In that system, the robot's position is obtained with the help of a GPS (Global Positioning System) and other digital sensors.
A common point can be seen in all of the studies mentioned: the use of few Fuzzy Logic rules. In contrast to the traditional modeling of this type of problem, the use of Fuzzy Logic in all of those works proved to be very simple and efficient.

Conceptual model

The model proposed in this work for the autonomous navigation problem uses Fuzzy Logic and considers as input all the distance measurements obtained by the car's exterior sensors. Fundamental questions of an autonomous navigation system include keeping the vehicle in a desirable path as well as keeping an appropriated ground velocity.
Six inference variables are used as input to the system, five of them being obtained by the vehicle's distance sensors and another accounting for the vehicle's instantaneous velocity. Figure 10(a) shows the location of the sensors in the vehicle. The result of the inference process determines the value of the two output variables: the turning angle of the vehicle and its acceleration.



(a) Location of the Sensors in the Vehicle.



(b) Pertinence Function: A - Left, Front, Right and Left Corners Sensors. B - Right Sensor.

Fig. 10. Conceptual Model.

The Sensors provide the distance (in a straight line) from the vehicle to an obstacle. An additional parameter could be the current visibility, which would determine how far an obstacle can be while still being detected by the sensor. The fuzzification is executed according to the readings of such sensors using the composition shown in Figure 10(b) as the pertinence function. The choice of different pertinence functions for the right sensor (Figure 10(b)) is justified by the necessity of keeping the vehicle to right of the road, respecting the right-hand

traffic regulation. This is achieved by modeling the functions considering "near" in the left side as being "nearer" than in the right side.
Besides the sensor's reading, the vehicle's current velocity is also evaluated using a pertinence function. After the fuzzification of these variables, the inference process occurs following the set of rules shown in Table 2.

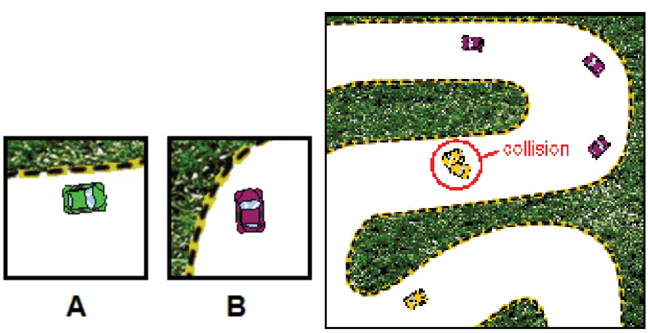
RULE	IF						THEN	
	LEFT	LEFT CORNER	FRONT	RIGHT CORNER	RIGHT	SPEED	ANGLE	ACCELERATE
1	NEAR						RIGHT	KEEP
2		NEAR					STRONG RIGHT	KEEP
3					NEAR		LEFT	KEEP
4				NEAR			STRONG LEFT	REDUCE
5				MEAN OR FAR	MEAN OR FAR		RIGHT	KEEP
6				FAR	FAR		STRONG RIGHT	KEEP
7			FAR			SLOW OR MEAN	KEEP	ACCELERATE
8		NEAR OR FAR	NEAR OR FAR	MEAN OR FAR		MEAN OR QUICK	KEEP	REDUCE

Table 2. Table of Rules Used in the Model.

The idea of using simple rules and in reduced number also favor the observation of emergent behavior. These rules are applied in each simulation step and the results are submitted to the defuzzification process.

Implementation of the simulation

Figure 11 shows the simulation running, where vehicles are seen navigating through an unknown route. The expected behavior is observed, as vehicles reduce their velocity during turns while keeping the distance to vehicles navigating in front of them or in the opposite direction.
An example of association between the rules and the agent's behavior is shown in Figure 11(a). Figure 11(a)-A shows an example of Rule 1 (see Table 2) executed as a result of the vehicle being near the left side of the road without a velocity's reduction needed. On the other hand, Figure 11(a)-B shows a hazardous situation that triggers Rule 2, which in turn changes the vehicles' trajectory and reduces its velocity drastically.



(a) Situations Associated to the Triggering of Rules.

(b) Occurrence of Accidents During the Simulation.

Fig. 11. Implementation of the Simulation - Situations Associated to the Triggering of Rules and Occurrence of Accidents.

The velocity range that the vehicle operates on is another parameter that can be informed during the creation of a new agent. In one simulation instance, some of the agents were instructed to navigate in above-average velocity. Figure 11(b) show the execution of such simulation as well as the occurrence of an accident caused by excessive velocity when making a turn. This is an example of an emergent behavior which wasn't explicitly modeled but is still completely coherent to the real system.

4.1.3 Group III: Advanced applications in various themes

This section shows the potential and flexibility of advanced multi-agent based simulations involving both more complex scenarios and agents with deeper behavior patterns.

4.1.3.1 Panic in crowds

Sociology deals with the birth and formation of social groups and institutions. On the other hand, the collective behavior field studies social phenomena that do not strictly run according to institutions and social norms. The people's behavior in such phenomena is decoupled from the social rules, and it emerges from the interactions among individuals.

Some collective behavior events include mobs, fans in a rock fest, bystanders glaring at a showcase, and so on. The social theorists of the 19th century were the firsts to study and propose theories describing the collective behavior phenomena.

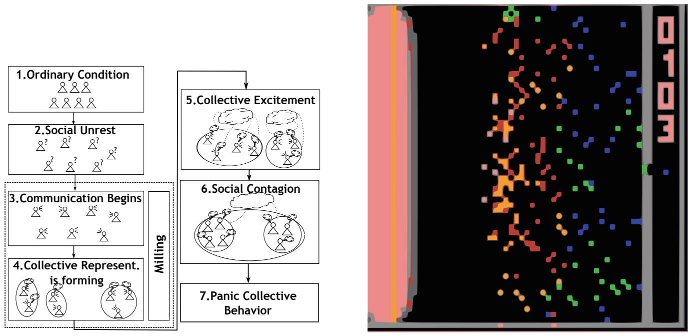
During the 20th century, some approaches were proposed to offer an overview of the collective behavior events and the individual's actions along these events. From a blind-folded action (relating groups of people to herds) to a more organized behavior with roles and some sort of hierarchy, the social phenomena studies evolved, improved and proposed new ways of looking at the collective behavior field.

The panic in crowds' phenomenon is a kind of collective behavior that happens during hazardous situations such as fire, earthquake and flooding. In such situations, life is at stake. Because of that, there is an urge to act and think fast, although the individuals do not behave in a randomly fashion. Instead, there is coordination, communication and a strong sense to act quickly (dos Santos França et al., 2009).

Because of this inherent emergency and the formation of behaviors and coordinated, social actions during the event, multi-agent based simulations could easily be applied to this social situation. Besides, it could be unethical and logistically impractical to create panic situations in real life. The social simulation approach allows "what if" scenarios and an even a more realistic portrayal of events, all without the health concerns that a simulation of panic in real life would imply. For instance, the room's disposition could be changed at will in order to observe how the individuals react to an obstacle or a small door. Collective behavior (and panic in crowds by extension) must follow some steps. Figure 12(a) shows the possible steps in an interactionist approach for the collective behavior phenomenon.

In short, the panic starts when an exciting event occurs. That event could be a fire or the furniture trembling. The individuals get curious about what is happening and what has changed the ordinary situation they were in. This social unrest leads them to find more information about the new condition they are facing. One way to take more information is milling, a verbal/non-verbal communication method that applies looks, touches, gestures and short expressions to pass a message to the others.

The milling is important because the individuals start making a collective representation of the situation. At first, this representation is at a micro level. However, as soon as the event gets more dangerous and the need of act becomes essential for survival, that representation starts to be unified by the collective excitement and the social contagion. During these steps, the individuals' communication and their actions work as a feedback mechanism that enhances and narrows the best lines of action.



(a) Possible Steps in an Interactionist Approach.

(b) A Panic Simulation Screenshot.

Fig. 12. Panic in Crowds - Model and Simulation.

When the individuals share a similar representation of the current situation, they choose a line of action and they act in order to re-establish the previous condition or to save themselves from the imminent danger, in a typical collective panic behavior. These steps are linear just for didactical purposes, since the individuals may jump or redo some of these steps when they see fit (dos Santos França et al., 2009).

The transition from a theoretical model to a computation model could be a challenge. However, if the transition occurs, the implemented model might be used to analyze the social phenomena in a privileged seat. The data, behaviors and communications shown during the simulation can be compared to the theory and even validate it. Figure 12(b) shows a panic simulation screenshot.

4.1.3.2 An Agent-Based Model for the spread of the Dengue Fever

The dengue fever is today the most spread arbovirosis in Brazil. Transmitted only by the female *Aedes aegypti* mosquito, it reaches its peak during the hot and humid Brazilian summer season. While there are many approaches to analyze the spread of the dengue fever, most of them focus on developing a mathematical model to represent that process. One disadvantage of such approach is to neglect the importance of micro-level behavior, focusing instead on the macro-level aspects of the system.

(Jacintho et al., 2010) developed an agent based simulation model for the spread of the Dengue Fever in the city of Rio de Janeiro, Brazil. Such model achieved similar results to the models currently being used, with the advantage of using just one set of agents and their interactions. The virus is transmitted to mosquitoes when they feed on the blood of a person already infected with the dengue virus. After an incubation period of eight (8) to twelve (12) days, the mosquito is then ready to propagate the disease. In humans, the incubation period might last from three (3) to fifteen (15) days, and symptoms are noticeable only after this period. Most importantly, there is no transmission through direct patient contact (including secretions) with a healthy person. The virus is not transmitted through water or food as well. To better understand and simulate the features observed in the real world, a transposition was made in order to build a model to be executed in a controlled environment. Rules were established for building a model as close to reality as possible, according to the scope of the project.

Below, a description of the simulation model based on (Otero et al., 2005; Santos et al., 2009) is presented, as well as the behavioral rules transposed to the computational model implemented in the Swarm platform.

(A) Mosquito Agent Behavior

As in the real world, this agent is modeled to display four (04) distinct stages: egg, larva, pupa and the land form, which corresponds to the adult mosquito. During simulation, each stage is represented internally in the mosquito agent, with no graphical/visual representation being used to differentiate distinct stages. The mosquito agent evolves according to the simulation progress and its behavior is internally adjusted according to its current life cycle stage.

(A.1) Egg Agent Behavior

Egg agents cannot move or feed and have an ideal temperature higher than 20 ºC, with an ideal humidity higher than 70%. Their outbreak will normally take place in about three (03) days.

(A.2) Larva Agent Behavior

These agents move only within their birthplace water spot, feeding on microorganisms and on their own egg remains. Their ideal temperature is between 25 ºC and 29 ºC, and their ideal humidity is higher than 70%. Under such conditions, this stage will take between three (03) to five (05) days to complete.

(A.3) Pupa Agent Behavior

Just like eggs, pupa agents cannot move nor feed. Their ideal temperature and humidity is around 20 ºC e 70% respectively, and they will have an 83% chance to become adult mosquitoes within three (03) days approximately.

(A.4) Adult Mosquito Agent Behavior

In this stage, agents are able to move freely through the environment up to 100m from their birthplace. Only females are capable of transmitting the disease, and that rarely happens at temperatures below 16 ºC, normally taking place under temperatures above 20 ºC. The mosquitoes proliferate at an estimated temperature between 16 ºC and 29 ºC, and have an average egg positivity of four (04) during their lifetime. Females will lay about 300 eggs on clean water with a 40% survival rate and 60% chance of being capable of transmitting the disease, i.e., other females. That means about 72 eggs will be considered in the simulation.
The mosquitoes can be killed by either exterminator agents or traps in the environment. They have an incubation period of about 8 to 11 days, by the time at which they become infectious and remain so for the rest of their life. Each infected female mosquito can propagate the disease to healthy humans by only a simple bite.

(B) Human Agent Behavior

As in the real world, this agent represents a human being, which might or might not become infected by the disease. Humans can move freely through the environment. After being bitten, it takes three (03) to six (06) days for the symptoms to become apparent. The Dengue Fever might last from three (03) to fifteen (15) days, with an average of five (05) to six (06) days. After being infected, the human agent can transmit the virus to others non-infected mosquitoes by blood contact during a mosquito's bite. This can occur one day before the first symptom appears and continues up to the last day of the disease. The death rates on multiple infections (also called the hemorrhagic dengue) are: 0.5% when infected twice; 10% when infected three times; 15% when infected four times and 25% when infected more than four times.

(C) Exterminator Agent Behavior

The exterminator agent moves freely through the environment based on the mosquitoes' gradient, attracted by areas of high density of mosquitoes in the map. This represents the public health organizations that map and notify all risk areas when planning control actions. Their role in the simulation exterminate adult mosquito agents.

(D) The Environment

The environment is not modeled as an agent by itself, but influences the agents' behaviors. Environmental factors such as temperature, food rate (probability of finding food) and humidity are globally defined as average values for the entire simulation, simplifying the simulation model and allowing the study of scenarios with different average values. There will be two states presented in the scenario: clean water and trap. Clean water servers as the place where mosquitoes will lay their eggs. Traps, on the other hand, are placed by exterminators to eliminate mosquitoes.

The conceptual model is transposed to a computational model that was later implemented in the Swarm simulation platform.

The environment interacts with all agents offering food for the mosquitoes, water for their reproduction and traps with substances to inhibit their proliferation. The results of such interactions between agents and the environment can be visualized in a 2D raster provided by the Swarm platform. According to Dantas et al. (Dantas et al., 2007) and Dibo et al.(Dibo et al., 2008), meteorological aspects such as temperature, humidity and precipitation can be used as predictors for Dengue incidence. In that sense, evaluating different climatic seasons allows a better understanding of the spread of the disease.

In this work, a tropical wet and dry climate region (Aw) is considered, according to the Köppen-Geiger climate classification (McKnight & Hess, 2000; Peel et al., 2007), as this is the predominant climate for most of Brazil. The weather in Brazil is characterized by high average annual temperatures and by a pluviometric regime that separates two distinct seasons: a rainy summer and a dry winter season.

Scenario I: Winter season

To simulate the Brazilian winter season, the Rio de Janeiro's climate information was used, with an average temperature of 18 ºC e average humidity of 45%. As the winter in Brazil is characterized by high dryness, 20 water spots were considered, with 5% of them set as traps. As the winter is a season with historically low dengue fever occurrence, only 10 exterminators were made available in this simulation scenario. The simulation started with 100 human agents, with 8% already infected with the disease, and 50 mosquitoes, with 60% already infected by the disease.

After 180 simulation cycles (60 days), no occurrences of hemorrhagic dengue (a human agent being infected more than once) were noticed, and the infection rate actually dropped to 7%. The number of mosquitoes in the environment also dropped from 50 to 30, with a 100% infection rate and with 111 mosquitoes in non-adult stages of their life cycles. Figure 13 (a) shows the simulation screen after 60 days. The color gradation shows the density of infected mosquitoes, and clearly displays few concentration spots. Figure 13 (b) shows the chart of mosquitoes in non-adult stages against adult mosquitoes. It was observed that even though there were many non-adult mosquitoes, several adults were unable to survive due to the harsh winter conditions (temperature, humidity and lack of water spots). Figure 13 (c) shows the

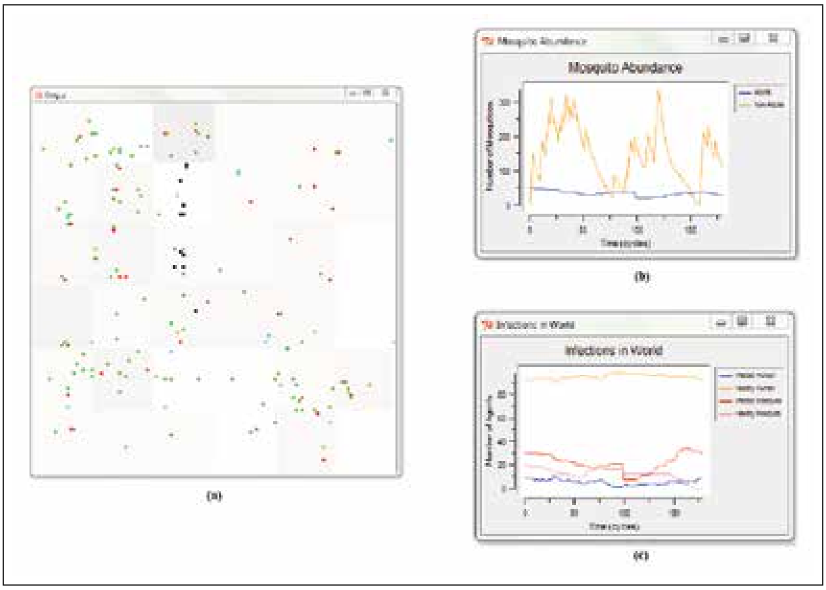


Fig. 13. Simulation Environment during a Winter Season and after 60 Simulated Days.

infections chart for the simulation world, and indicates that only a few humans were infected during that season.

Scenario II: Summer season

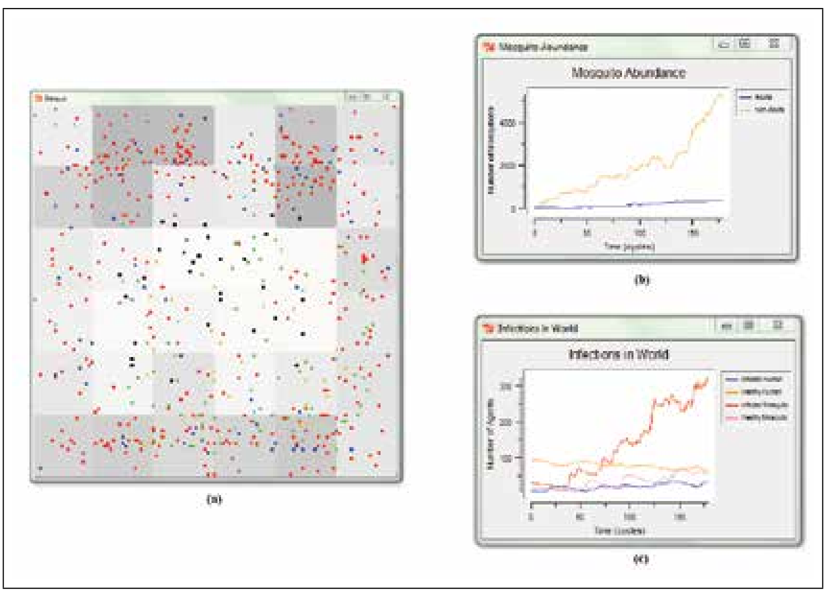


Fig. 14. Simulation Environment during a Summer Season and after 60 Simulated Days.

In order to simulate the Brazilian summer season, once again it was considered the Rio de Janeiro's climate data, with an average temperature of 29 ºC and humidity of 85%. As the summer in Brazil is predominantly rainy, we considered 38 water spots, with 40% set as traps.

Also, as the summer season shows a historical high infection rate, we made 30 exterminators available. The simulation started with 100 human agents, being 8% infected by dengue and 50 mosquitoes, with 60% infected as well.
After 180 simulation cycles (60 days), we observed that 3 humans died due to hemorrhagic dengue and the infection rate raised considerably to 33%. We also noticed that the number of mosquitoes jumped to 394, with 83% infected and with 5086 mosquitoes in non-adult stages. Figure 14 (a) shows the simulation screen after 60 days. Figure 14 (b) show the chart for the number of mosquitoes in non-adult stages against adult mosquitoes. An almost exponential growth of mosquitoes in non-adult stages is clearly observed, which corroborates to the rapid increase seen in the population of adult mosquitoes. Figure 14 (c) shows the infections chart for the simulation world, with a noticeable increase in the number of infections of both humans and mosquitoes.
The two simulated scenarios allowed the validation of the proposed model, showing very similar results to the ones found by (Câmara et al., 2007; Dantas et al., 2007; Dibo et al., 2008; Otero et al., 2005; Santos et al., 2009). Simulating scenarios that considered climatic seasons allowed the validation of the model against infection rates and mosquitoes proliferation data during the same seasons in real life. In both scenarios, it was also possible to notice the emergency of a spreading behavior for the dengue fever caused by the interaction of the simulation agents (mosquitoes, humans, exterminators) and the environment (food, clean water and traps).

5. Final considerations

This chapter offered to readers a concise overview of the current outlook on artificial life and multi-agent based simulations theory and practice. It also presented some examples of simulation models implemented by the Social and Artificial Intelligence research team at the Centre of Mathematics, Computation and Cognition of the Federal University of ABC in São Paulo, Brazil.
The information presented in this chapter can be used as a valuable reference for those who are new to multi-agent based simulation and are interested by this subject. Also, this chapter provides information to the more seasoned and experienced MABS researcher looking for examples of simulations that integrate MABS and other research areas.
The main aspects of multi-agents simulation were covered, a comparison between traditional and agent-based approaches was made, and important topics on multi-agent simulation, such as emergent behavior, bottom-up modeling and self-organization were explored; two simulation platforms were analyzed, providing the reader a general understanding about MABS.
Through the examples presented in this chapter, it was provided the means for applying an agent-based approach to tackle artificial life problems, thus offering a different perspective when compared to traditional methods. By using this approach, the models studied can be exhaustively modified in order to improve the proposed theoretical models, reinforcing the main idea of the MABS area, which is to study how the modeled system behaves under certain conditions and, to ultimately examine in varying degrees of detail, the consequences of changing internal behaviors of the system, and vice versa.
Therefore, the simulation models presented in this chapter evidence the potential of transposing very complex systems to a computer environment. Through the proposed models and theoretical framework used, the MABS approach allowed us to control the inner parts that comprised the main systems, improving them without the need of retheorizing the template

and, finally, demonstrating that complex and difficult behavior can now be reproduced and modified without exceptional efforts.
For further studies, an advice should be brought to life: Multi-Agent Based Simulation is a multidisciplinary field, and that implies a combined effort among scientists from as diverse fields as such as biology, sociology, physics, mathematics, law, linguistics and so on. Researchers from these fields will challenge artificial intelligence scientists which, after being "incited" by such issues, will look at such challenges from a computer scientist's point of view. However, this is not an isolated view. These other researchers will also affect the computer scientist by presenting a new way of perceiving such issues, feeding back the research cycle and fomenting a virtuous round.

6. References

Axelrod, R. (1998). Advancing the art of simulation in the social sciences, *Complexity* 3(2): 16–22.

Baltes, J. & Otte, R. (1999). A fuzzy logic controller for car-like mobile robots, *Prcoeedings of the International Symposium on Computational Intelligence in Robotics and Automation*, IEEE, Monterey, CA, pp. 89–94.

Cioffi-Revilla, C. & Rouleau, M. (2010). MASON RebeLand: an Agent-Based model of politics, environment, and insurgency, *International Studies Review* 12(1): 31–52.

Câmara, F. P., Theophilo, R. L. G., dos Santos, G. T., Pereira, S. R. F. G., Câmara, D. C. P. & de Matos, R. R. C. (2007). Regional and dynamics characteristics of dengue in brazil: a retrospective study, *Revista da Sociedade Brasileira de Medicina Tropical* 2(20): 192–196.

Dantas, R. T., Limeira, R. C., Menezes, H. E. A. & Sousa, N. M. N. (2007). Influence of the meteorological variables on the incidence of primness at joão pessoa – pb, *Revista Fafibe On Line*, number 3.

Dibo, M. R., Chierotti, A. P., Ferrari, M. S., Mendonça, A. L. & Chiaravalloti Neto, F. (2008). Study of the relationship between aedes (stegomyia) aegypti egg and adult densities, dengue fever and climate in mirassol, state of são paulo, brazil, *Mem. Inst. Oswaldo Cruz*, Vol. 103, pp. 554–560.

dos Santos França, R., das Graças B. Marietto, M. & Steinberger, M. B. (2009). A multi-agent model for panic behavior in crowds, *Fourteenth Portuguese Conference on Artificial Intelligence* .

Engbert, R. & Drepper, F. (1994). Chance and chaos in population biology–Models of recurrent epidemics and food chain dynamics, *Chaos, Solitons & Fractals* 4(7): 1147–1169.

Farhi, O. & Chervenkov, Y. (2008). Optimal fuzzy control of autonomous robot car, *Intelligent Systems, 2008. IS '08. 4th International IEEE Conference*, IEEE, Varna, pp. 62–65.

Ferber, J. (1996). Reactive distributed artificial intelligence: Principles and applications, *in* G. O'Hare & N. Jennings (eds), *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons, New York, pp. 287–314.

Gardelli, L., Viroli, M., Casadei, M. & Omicini, A. (2008). Designing self-organising environments with agents and artefacts: a simulation-driven approach, *Int. J. Agent-Oriented Softw. Eng.* 2(2): 171–195.

Gilbert, N. & Terna, P. (2000). How to build and use agent-based models in social science, *Mind & Society* 1(1): 57–72.

Gilbert, N. & Troitzsch, K. G. (1999). *Simulation for the Social Scientist*, Open University Press.

Goldspink, C. (2002). Methodological implications of complex systems approaches to sociality: Simulation as a foundation for knowledge, *Journal of Artificial Societies and Social Simulation* 5(1): 1–19.

Group, S. D. (2009). The swarm development group wiki, `http://www.swarm.org/`.

Jacintho, L. F. O., Batista, A. F. M., Ruas, T. L., Marietto, M. G. B. & da Silva, F. A. (2010). An agent-based model for the spread of the dengue fever: A swarm platform simulation approach, *SpringSim'10 Preliminary Proceedings* .

Jianjun, Z., Man, Z., Gang, L. & Su, L. (2008). Fuzzy control for automatic steering and line tracking of agricultural robot, *CSSE '08: Proceedings of the 2008 International Conference on Computer Science and Software Engineering*, IEEE Computer Society, Washington, DC, USA, pp. 1094–1097.

Lempert, R. (2002). Agent-based modeling as organizational and public policy simulators, *PNAS* 99(90003): 7195–7196.

Lu, H.-C. & Chuang, C.-Y. (2005). The implementation of fuzzy-based path planning for car-like mobile robot, *ICMENS '05: Proceedings of the 2005 International Conference on MEMS,NANO and Smart Systems*, IEEE Computer Society, Washington, DC, USA, pp. 467–472.

McKnight, T. L. & Hess, D. (2000). *Climate Zones and Types: The Köppen System*, Prentice Hall.

Otero, M., Solari, H. G. & Schweigmann, N. (2005). A stochastic population dynamics model for aedes aegypti: formulation and application to a city with temperate climate, *Elsevier Science* .

Parker, L. E. (1992). Local versus global control laws for cooperative agent teams, *Technical report*, Cambridge, MA, USA.

Peel, M. C., Finlayson, B. L. & McMahon, T. A. (2007). Updated world map of the köppen–geiger climate classification, *Hydrol. Earth Syst. Sci.* (11): 1633–1644.

Russell, S. & Norvig, P. (2004). *Inteligência Artificial*, 2^a. ed. edn, Elsevier Editora, Rio de Janeiro, Brasil.

Santos, L. B. L., Costa, M. C., Pinho, S. T. R., Andrade, R. F. S., Barreto, F. R., Teixeira, M. G. & Barreto, M. L. (2009). Periodic forcing in a three-level cellular automata model for a vector-transmitted disease, *The American Physical Society* .

4

Development of Multi-Agent Control Systems using UML/SysML

Iskra Antonova and Idilia Batchkova
University of Chemical Technology and Metallurgy, Department of Industrial Automation
Bulgaria

1. Introduction

The successfully implementation of control systems depends greatly on achieving lower development costs, short time of design and high level of safety and reliability. However the growing amount of ICT applications in the field of automation and control, from another side, increases rapidly the size and complexity of the software systems used in this domain. The main trends in automation are connected with the development and implementation of reconfigurable distributed control systems consisting of multiplicity non-hierarchical modules linked together via different types of communication systems. In order to control the complexity of distributed real-time systems the following main requirements have to be fulfilled: reliable concept for decomposition and modularity, openness for extensions in the cases of new products, machines and devices, the general architecture model of the system has to be producer independent, use of encapsulated, reusable components.
To successfully meet the new requirements to the developed distributed control systems various modelling approaches and methods are increasingly resorting to be used. The control functions in distributed systems can be suitably described with any discrete-event model like automata, Petri nets, Grafcet/SFC, formal languages or IEC-61499 based Function blocks. The major drawback of these approaches is the lack of inheritance and polymorphism that may limit the reusability in non-vital applications. In order to overcome these shortcomings new approaches for software system development in control and automation are needed. Many working groups are trying to fill the gap between state of the art in software engineering and state of the practice in the control application domain through the use of object-oriented and agent-oriented modelling techniques.
The agent community has considerable interest in developing methods and techniques for specifying, modeling, implementing and verifying of Multi-Agent Systems (MAS) for distributed control, but so far no standardized design methodology has been recognized. The successfully mission of Unified Modelling Language (UML) [OMG-UML, 2010] as unification of numerous different object-oriented approaches led to the idea of applying UML to the design of MAS. From a software point of view, agent-based systems can be regarded as a specialization of object-oriented systems but UML does not provide enough means for capturing all agents related modeling aspects like autonomy, pro-activity and cooperation. The software agents have their own thread of control, localizing not only code and state but their invocations as well [Parunak, 1999]. In an open and distributed, agent-based, integrated control environment, the need of standard mechanisms and specifications are vital for ensuring the interoperability of the autonomous agents.

The approach presented in this chapter aims to extend UML for development of multi-agent control systems based on its profile for System Engineering – SysML [OMG-UML, 2003], [OMG SysML, 2006], the upcoming IEC-61499 standard defining the basic concepts and reference architecture and models for design of modular, re-usable, open and vendor independent distributed control applications [IEC-61499, 2005], and the FIPA's reference model for agent communications [FIPA, 2010]. SysML is the first UML profile for system engineering, which supports the specification, analysis, design, verification and validation of a broad range of complex systems. Using this profile, essential aspects of control engineering issues such as concurrency, hardware architecture, and requirements traceability can be applied to multi-agent and embedded systems. The customization of UML for systems engineering supports modeling of the whole closed loop control system including hardware, software, data, procedures and facilities. There is a clear correspondence between UML/SysML and IEC-61499 standard. The modelling concepts in both standards share many similarities and this allows the mapping between basic elements from IEC-61499 and UML/SysML profile using additional stereotypes. In order to extend the standardized concept to the whole development life cycle of a control application and to use the all benefits of object-oriented paradigm, the concepts of IEC-61499 standard are extended with different UML/SysML notations. The UML/SysML profile is extended and at the same time restricted with different stereotypes from FIPA communication standard in order to fill the gap from absence of unified communication protocol in applications based on the IEC-61499. The basic idea is to create an executable functional model of the software at an early stage of the development process based on requirements specifications, possibility for formal specification, verification and validation of the system based on modified Harmony SE methodology.

The chapter is organized in 6 parts. After this introduction, in part 2 the current trends in automation and control domain are discussed. In part 3 the main features of multi-agent systems in respect to the development of multi-agent control systems are summarized and the approaches, methods and tools for their achievement are shortly described. Part 4 of the chapter analyses different extensions of UML for modelling of hard real time systems including the special topic of agent-based control systems. As the suggested approach is based on the collaborative use of UML and the reference architecture and models proposed in the IEC-61499 standard, a short overview of these models is presented in this part too. In part 5 the proposed approach for development of multi agent control systems based on the UML profile for system engineering - SysML, and the standardized concepts, models and architectures of IEC-61499 and FIPA standards is described. A modified methodology based on a subset of the large methodology for integrated system and software development process Harmony-SE is used. The development process includes the following basic cycles Requirements Analysis, System functional analysis, Architecture design and Hardware/Software design specification supported by different UML/SysML notations and diagrams, restricted according IEC-61499 and FIPA standards. A case study, illustrating the suggested approach is presented in part 6. This example shows the control system development processes to the FESTO Distribution station. Finally some conclusions are made.

2. Current trends in the automation and control domain

The global competition between enterprises and the new production strategies adopted by business require new types of automation and control systems, characterized by a high degree of horizontal and vertical integration and self-organizing features to enable rapid adaptation to changes in the environment. To meet the challenges of modern enterprises, automation and control systems must meet certain requirements, which can be summarized as follows [Bussmann&McFarlane, 1999]:

- The architecture of the control should be decentralized and product- / resource- based;
- Control interactions should be abstract, generalised and flexible;
- Control must be proactive and reactive;
- Control must be self-organizing.

The main challenges facing the field of automation and control are associated with a high degree of flexibility and adaptability to the emergence of unexpected needs and the wide variety of products with a view to achieving the agility of manufacturing systems. These requirements can be met by control systems that have reduced complexity, increased flexibility and adaptability in real time, are extendable, heterogeneous and support autonomous operations. The architecture of control systems has to ensure production agility through rapid adaptation and cheap changes in production environment.

To fulfil the requirements for agility, control systems of new generation are challenged to solve two major and diametrically opposed tasks:

- Full integration of information and control systems in order to automate the entire product lifecycle;
- Maintenance of the relative autonomy of the elements of the production system through transition from centralized to decentralized, distributed structures of control and information processing.

The main trends in automation are connected with the development and implementation of reconfigurable distributed control systems consisting of multiplicity non-hierarchical modules linked together via different types of communication systems. In order to control the complexity of distributed real-time systems, the following main requirements have to be fulfilled: reliable concept for decomposition and modularity, openness for extensions in the cases of new products, machines and devices, the general architecture model of the system has to be producer independent, use of encapsulated, reusable components.

The object-oriented approach ensures three main properties of the modelled system and its entities: encapsulation, inheritance and polymorphism. Encapsulation allows software to be developed on a modular principle, inheritance supports it's automatically reuse, thereby increasing robustness and significant facilitates the development processes. Polymorphism facilitates the development of independent modules in a constructive manner. The main disadvantages of object-oriented modelling are related to the absence of abstraction or support to the relationships and interactions between objects. Complex applications usually require interoperability between multiple objects. When designing and creating objects is not always known how and when to use their functionality and their state changes. This can lead to unexpected errors. Component-based development goes a step further by introducing more abstraction in terms of self-descriptive characteristics, which enable connections to other objects and usage of external functionality. This facilitates the development processes, but unfortunately does not contribute significantly the achievement of reuse and modification of the interface in real time, i.e. components used are also passive and can be described or used from outside. One of promising directions for achieving the above objectives in the development of control systems is the development of multi-agent systems by applying the principles of agent-oriented software engineering. Some of the most important features and principles of multi-agent systems are briefly discussed in the next part of the chapter.

3. Multi-agent systems

Multi-agent systems are characterized by local autonomy, social interaction, adaptability, robustness and scalability, and for these reasons are a very promising paradigm for addressing the challenges facing automation and control systems, associated with the development, operation and maintenance of reconfigurable distributed control systems. On the one hand multi-agent systems are widely accepted research topic in the field of automation and control systems, on the other, however, the field of automation and control is a major challenge to the agent-oriented software engineering to revealing its power, efficiency and relevance. MAS can be defined as "a loosely coupled network of problem solvers (agents) that work together to solve problems that are beyond the individual capabilities or knowledge of each problem solver" [Durfee & Lesser, 1998]. Jennings and Wooldridge [Jennings & Wooldridge 1998] have defined an agent as "a computer system situated in some environment and capable of autonomous action in this environment, in order to meet its design objectives". Agents have the following main properties and characteristics [Wooldridge & Jennings, 1995]:

- autonomy: agents encapsulate some state (that is not accessible to other agents), and make decisions about what to do based on this state, without the direct intervention of humans or others;
- socialability: agents interact with other agents (and possibly humans) via some kind of agent-communication language, and typically have the ability to engage in social activities (such as cooperative problem solving or negotiation) in order to achieve their goals.
- reactivity: agents are situated in an environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the INTERNET, or perhaps many of these combined), are able to perceive this environment (through the use of potentially imperfect sensors), and are able to respond in a timely fashion to changes that occur in it;
- pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative;

Opportunities for modelling of multi-agent systems at a high level of abstraction is a research topic in the field of Agent-Oriented Software Engineering (AOSE), which is developing very actively over the last few years. The research community in the field of AOSE investigates the UML extensions to support the modelling of basic agent-oriented concepts such as agents, ontology and interaction protocols. From a software engineering perspective, agent systems are a specialization of object-based systems, in which individual objects have their own threads of control and their own goals or sense of purpose. With the development of UML and the introduction of active objects, the differences between objects and agents decreased, owing to the improvement of the language meta-model and the establishment of new notations and formalization of the various diagrams used in the language. This allows achieving a higher degree of coverage of the basic properties of the agents and their interactions. Moreover, major advantage of using UML in the development of MAS compared with other alternatives is that the use of standard mechanisms and specifications is critical to ensure interoperability between autonomous agents in an open and distributed, agent-based integrated control environment.

Use of UML for development of multi-agent control systems beyond the problems with MAS, raises other important issues too. They relate to the satisfaction of real-time requirements to the control systems, their quantitative assessments, as well as the procedures for verification and validation of their behaviour. The next section gives an overview and analysis of some different approaches to solving these problems.

4. UML extensions for modelling of multi-agent based control systems

4.1 Overview of real time extensions to UML

UML is a general purpose modelling language, which gives uniform notations and a meta-model for object-oriented modelling and software design. It does not specify a methodology for software design but supports the Model Driven Architecture (MDA) consisting in transformation of different platform independent models towards executable applications. UML aims to provide an integrated modelling framework, covering structural, functional and behaviour descriptions.

The main challenges towards a real-time UML are connected with the creation of different mechanisms to handle real-time features such as: models of physical time, timing specifications, timing facilities, modelling and management of physical resources and concurrency. In the last years there are many working groups developing and using different approaches and techniques in order to extend the object-oriented software development and especially UML to the field of real time system modelling and analysis. Further in the next paragraphs we try to summarize and analyze these research activities and the current state of the art in order to select and use the most suitable and effective approach for design and analysis of multi-agent based distributed control systems.

The first attempts to model control systems using UML appear almost parallel with the emergence of UML, setting some guidelines for the development of language. There are many different proposals for extending UML to support the design and analysis of real-time systems and they can be joined in 2 main abstract classes, shown in Fig.1. The first class of extensions follows the idea of Douglas [Douglass, 1998] and Selic [Selic, 1998] that the behaviour of complex real-time control systems can be fully described by using the standard capabilities of UML and is subdivided in two classes applying the following extension strategies:

- Changing the UML meta-model by explicitly adding new meta-classes and other meta-constructs;
- Creating UML profiles (standard and specific) on the base of stereotypes, constraints and tagged values. The 3 built-in extension mechanisms can be used separately or together.

Extensions to the first subclass are a priority of OMG (Object Management Group), but there are other initiatives too as for example the UML+, where new elements are added to the UML metamodel in order to achieve RT capabilities [Lavazza et al., 2001], [Bianco et al., 2001], [Bianco et al., 2002]. The current work on the evolution of UML standard at OMG aims the integration of the most successfully contributions to the real time issues. UML2.x provides means for architectural modelling inspired from UML-RT [Gerard et al., 2002], with structured classes, ports (isolate an object internals from its environment), connectors (which link communicating ports) and protocols (defined, reusable, interaction sequences). A structured class may have an internal structure. This structure is composed of structured classes, ports, protocols, etc. The idea of hierarchy appears too. UML 2.x presents some features that support real-time aspects and they may be summarized as follows:

- Support the modelling of concurrency through introduction of active objects, concurrent composite states and concurrent operations.

- For expressing of timing constraints two new data types are provided – Time and Time Expression, which may be used in state and sequence Diagram.
- Introducing of new diagram, so called Timing Diagram for expressing the timing behaviour of the system.
- Better semantic definition of the state diagram.
- Changes in the activity diagram, inspired from the advantages of Petri nets.
- New graphical features for the class diagram and new notations for the existing diagrams.
- Better definition of components and subsystems.
- Improved extension mechanism in respect to the user-defined meta-classes and relations, and profile mechanisms.
- UML 2.x is more open to formalization and includes more run time semantics.

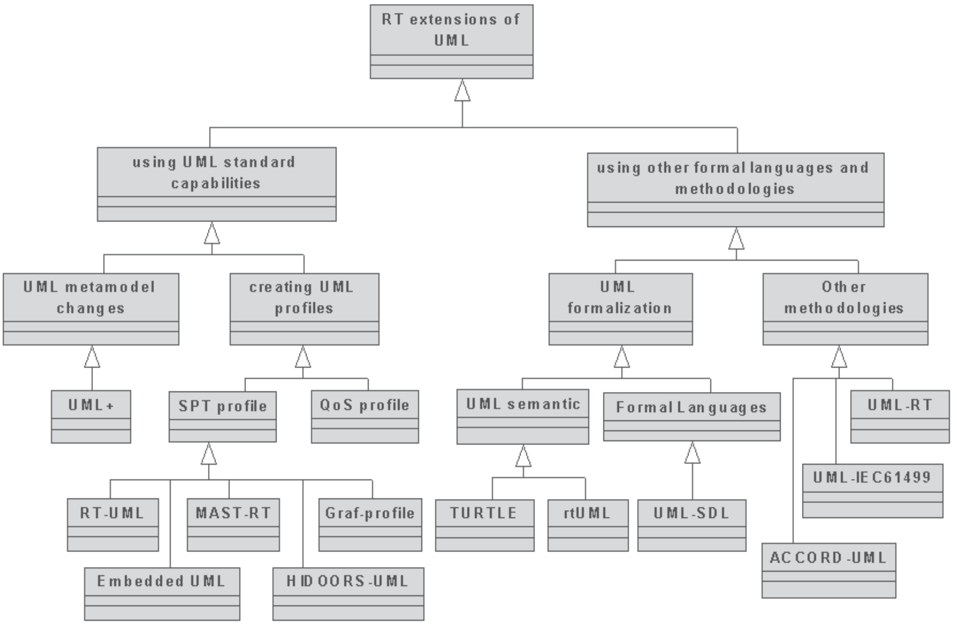


Fig. 1. UML extensions for real time

Despite the variety of structural and behaviour improvements, the new versions of UML2.x has still insufficiently support for development of distributed control system in respect to expressing hardware-software interdependences and timing models. That is why the using of specialized profiles based on UML2.x is the best solution for these purposes.

The objective of UML profiles, which are the focus of the second subclass of approaches, is to package terminology and substructures, specific for a particular application domain. One of the first attempt to provide RT capabilities of UML in this direction is the OMG initiative for creating of the profile for Schedulability, Performance, and Time Specification (SPT-profile) that is proposing a framework to model quality of service, resource, time and concurrency concepts in order to support predictive quantitative analysis of the UML models [OMG, 2003]. This profile supports two well-established forms of time-based model analysis: schedulability analysis based on schedulability theory and performance analysis based on queuing theory or Stochastic Petri Nets. The SPT-profile is used as a basis for some other UML profiles some of them are RT-UML [DiPippo, 2000], MAST-RT [Medina et al., 2001], embedded UML [Grant&Jean, 2001], HIDOORS UML profile [Kesten, 1992], Graf-Ober profile [Graf&Ober, 2003] etc. An implementation of this idea is the CASE tool of I-logix – Rhapsody that will be used to illustrate the proposed approach in this chapter. The new UML MARTE profile supporting the specification, design and verification/validation in the model driven development of real time and embedded systems will replace the SPT profile with the appearance of the new UML2.3 version [OMG-UML, 2010]. UML profile for modelling Quality of Service and fault tolerance characteristics and mechanisms (QoS-profile) adopted by OMG in 2004 supports the implementation of SPT-profile through the definition of a catalogue with different categories of QoS characteristics, Quality models and UML model annotations using QoS requirements [Miguel, 2003].

The second class of RT extensions is connected with the combination of standard UML capabilities with those of other real-time frameworks, languages and methods, covering different real time features of the designed systems.

One very large class of RT extensions of UML is connected with the filling the gap from the lack of well-defined formal semantic and ensuring of capabilities for formal verification of designed UML models. In general, these extensions can be separated into two major categories. First of them is based on defining of the semantic domain and giving a sound semantics to the graphical notations in this domain. Examples of such profiles are TURTLE [Apvrille et al., 2004] and rtUML (krtUML) [Damm et al., 2002]. The plentiful graphical notations however make the semantic domain very complex, and impede the analysis tool support. This obstacle gives an advantage to the more workable approach including a combination of graphical notations and formal specification languages, and exploring the formal reasoning capabilities of formal methods. There is variety of works in the field of software engineering based on well-established traditional formal methods such as VDM, Z, B, and their object-oriented extensions such as VDM++, Z++, Object-Z etc. [Mwaluseke&Bowen, 2001]. One of the mostly used combinations in the development of RT applications is that between UML and SDL, which share a number of qualities, like having a graphical notation, good readability and good tool support. They also incorporate object orientation and state machines, which make UML and SDL suitable to work together [Verschaeve, 1999].

Typical representative of the approaches, which combine UML with other methodologies, is the UML-RT profile of IBM Rational Rose. It is based on some concepts of the ObjecTime ROOM methodology and provides three principal constructs (capsules, ports, and connectors) for modelling the structures of a real-time system using the basic UML1.4 mechanisms of stereotypes and tagged values. The main shortcomings of UML-RT are in specification of time constrains and timeliness properties because there are not means for it. It is more suitable for modelling the structure and communications between the different elements in the system.

Another very promising approach especially applied for development of distributed real-time systems in control and automation domain is the collaborative use of UML and the models proposed in the IEC-61499 standard. A short overview of the proposals in this field are presented and analyzed in part 4.3 of the chapter.

The main conclusions, which are deduced from the analyzed approaches for real time extensions of UML, may be summarized as follows:

- There is a strong interest worldwide on the real-time extensions of UML and their application in different applications domain;

- Current proposals for real-time UML are not complete and entirely compatible;
- Most of the proposals provide a good support for modelling of concurrent processes but are meagre to express quantitative real-time features (such as deadlines, periods, priorities);
- There are many proposals based on proprietary solutions, which are not fully compliant with the UML standard.

The main disadvantages of UML2.1 in respect to solving the tasks of control and system engineering are:

- Inability to model requirements and parametric equations and linking them to the structure class diagrams;
- Inability to model flows in the structure class diagrams;
- Inability to model hierarchical structures in detail.

4.2 UML profile for system engineering - SysML

SysML is a general-purpose modelling language for system engineering that reuses a subset of the last UML2.1 version and provides additional extensions through stereotypes, diagram extensions and model library in order to model a wide range of system engineering problems as for example specifying requirements, structure, behaviour, allocations and constraints on system properties to support engineering analysis. The reusable subset of UML, known as UML4SysML includes Interactions, State machines, Use Cases and Profiles. In Fig.2 the set of SysML diagrams in respect to their modelling aspects is summarized.

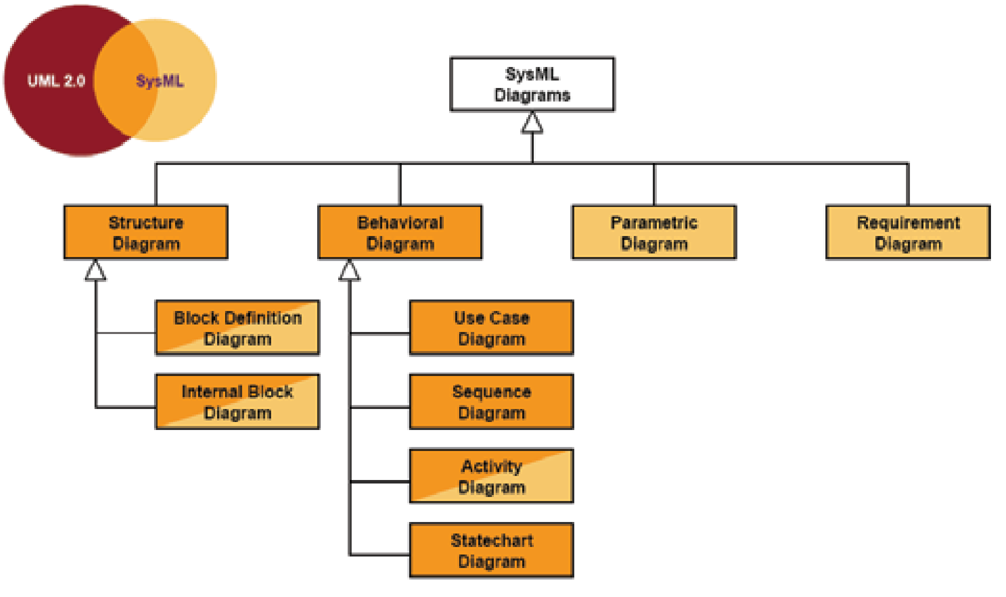


Fig. 2. SysML Diagrams [OMG-SysML, 2006]

The system structure design is supported by four types of diagrams: Block Definition Diagram (BDD), Internal Block Diagram (IBD) reinforced by Parametric and Packages diagrams. The BDD is an extended and restricted UML class diagram that is composed of blocks and relationships between them, such as associations, generalizations and dependences. The SysML blocks are based on UML classes with extended composite structure and are defined in terms of properties, operations and relationships. The IBD captures the internal structure of a block in terms of properties and connectors between properties. Three general categories of properties are recognized: values, parts and references to other blocks. Ports are a special kind of parts, which give access to internal structures from the outside of a composite object. In SysML, unlike UML, ports can also be complex hierarchical structures. There are two basic types of ports – standard and flow ports. Standard ports are the same as in UML2.x and allow specifying services to or from the environment through the definition of provided and required interfaces. Flow ports are an extension of UML2.x ports and specify what "can" flow in or out of a block. Different items such as data, material or energy may flow through these interaction points in order to achieve asynchronous, broadcast, or send and forget interactions. The concept of item flows specifies what "does" flow between blocks and/or parts and across associations or connectors in a particular usage context. Parametric diagrams allow capturing system properties and constraints in form of different equations (algebraic, logical, differential, etc.) and support the analysis and verification processes. Packages diagrams are the same as in UML2.x and facilitate the decomposition and encapsulation of the projects.

The Behaviour diagrams incorporate four diagrams too, namely: activity diagram, sequence diagram, state machine diagram, and use case diagram. The activity diagrams are used to describe the flow of control and flow of inputs and outputs among actions. The state machines describe the constructs used to specify state based run time behaviour in terms of system states and their transitions. The sequence diagrams describe the communications between different structural elements over the time and identify the required relationships and messages. The use case diagrams model the relationships between users and a system enabling the specification of requirements to the system behaviour at the different decomposition levels.

The Requirements diagrams, which can be presented in graphical, tabular or tree structure format, are used to specify different constructs for system requirements and to cover the relationships between them. In SysML two kinds of requirements are used – functional and performance, as they specify the capabilities or the conditions which must be performed or satisfied by the system.

Other modelling capabilities of SysML, not shown in fig.2 are the cross-cutting constructs, such as allocations for connecting the different views, and Profiles & Model libraries allowing further to customize and extend SysML to specific applications. SysML includes extensions supporting the causal analysis, the verification and testing processes and the decision tree development.

SysML is developed as an open source specification and the last draft SysML1.2 is submitted to the OMG for technology adoption in June 2010. There are numerous modelling tool vendors who have already updated their UML2.x based tools to comply with the OMG SysML specification as for example ARTiSAN Studio, Rhapsody, MagicDraw, Enterprise Architect etc. [Huynh&Osmundson, 2005]. For the case study, described in this chapter Rhapsody of Telelogic is used [Rhapsody].

The advantages on the UML profile for system engineering SysML in comparison to these based on UML2.x may be summarized as follow:

- SysML supports the whole life cycle by the development of control engineering applications from the requirements definition to the software implementation;
- Based on extended activity diagrams, parametric diagrams and flow ports and items, the proposed approach may be applied for continuous and hybrid systems;

- The possibilities to model the physical systems in detail enhance the procedures for analysis, testing and validation of designed closed loop behavior of the system;
- Through encapsulated objects in UML 2.x and hierarchical structure, which they gain using SysML profile the object-oriented systems get near to agent-based systems;
- In order to improve the modelling processes, the UML/SysML notations may be mapped to concepts from IEC-61499 standard.

4.3 Development of IEC-61499 based control systems using UML

The IEC-61499 standard defines the basic concepts, reference models and architecture for development of modular, re-usable and open, vendor independent distributed process measurement and control applications, characterized through three main features: interoperability, portability and configurability [IEC-61499, 2005]. This standard uses the function block (FB) concept as a main building block of a control system, represented at different level of integration – device, resource and application. Three different kinds of FB are defined: basic (BFB), composite (CFB) for encapsulation of complex functionality through networks of BFB and service interface function block (SIFB) for providing interfaces for unidirectional (publish/subscribe) and bi-directional (client/server) communications. The control system may be modelled as logically connected function blocks through their input and output data and events. The BFB, shown in Fig.3 is presented by an input and an output interface composed of input and output events and data. The internal view of a basic function block includes an Execution Control Chart (ECC), internal data and internal algorithms. The ECC as shown in Fig.4 is a state machine, consisting of states, transitions and actions, invoking the execution of algorithms, associated to ECC states, in response to event inputs. One of the states is initial state and all other execution control states may have one algorithm and/or one output event associated. The evolution of the ECC state machine from an execution control state to other is realized by the execution of control transitions. In general the ECC is the relationship between events and algorithm executions, which are specified by the special kind of even-driven state machines. Every function block is characterized by its type name and instance name, which are used to identify a function block.

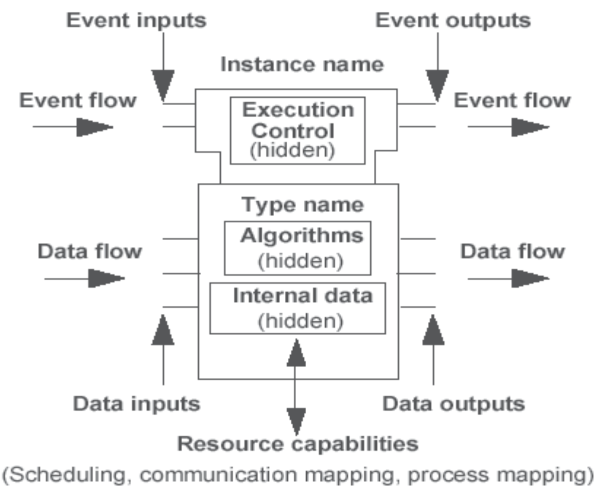


Fig. 3. Structure of a basic function block (IEC-61499)

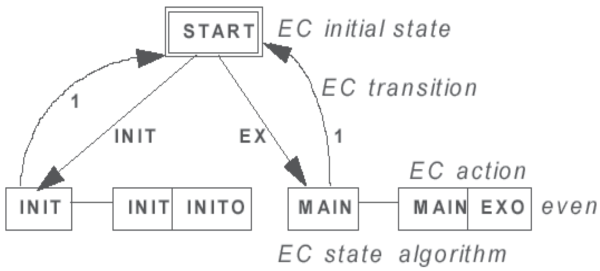


Fig. 4. Basic elements of the Execution Control Chart

Some of the main disadvantages of IEC-61499 standard are the absence of unified methodology for development of IEC-61499 based control systems and of formal semantic for describing control applications. This means that the approach based on the function block concept do not take into account the current situation and practice in the field of software engineering and makes the development processes and validation/verification phases of designed control systems difficult.

There are mainly two methodologies, which try to overcome these shortcomings using both UML and IEC-61499 standard for development of distributed control systems. The first approach, suggested by Thramboulidis, describes the model structure using class and sequence diagrams [Thramboulidis, 2001]. For these purposes, different stereotypes in respect to the IEC-61499 standard are defined as the FB stereotype, IPT (Industrial Process Terminator), the Data- and Control Dependency stereotypes, and Real-World Object Dependency stereotype [Thramboulidis, 2004]. The second approach suggests the control systems to be designed in an UML environment and after that the model to be transformed by well-defined transformation rules in FB based model using FBDK [FBDK, 2010]. A new modelling language UML-FB is defined [Dubinin&Vyatkin, 2004].

The approach, suggested in this chapter overcomes the shortcoming of the purely functional IEC-61499 standard by applying combined methodology based on the benefits of SysML and IEC-61499 standard in order to extend the standardized concept to the whole development life cycle of a control application, to model the close loop system and to use all benefits of object-oriented paradigm. There is a clear correspondence between the UML profile for System engineering SysML and IEC-61499 based models. The modelling concepts in both standards share many similarities. For example FB and parts are analogous, IEC-61499 data and event interfaces and ports are similar. `

4.4 Modelling MAS control systems using UML/SysML

The agent community has considerable interest in developing methods and techniques for specifying, modelling, implementing and verifying of multi-agent systems (MAS) for distributed control, but so far no standardized design methodology has been recognized. The successful industrial deployment of agent technology promotes techniques that reduce the risk inherent in any new technology, i.e. using standard representation of methods and tools. The successfully mission of UML as unification of numerous different object-oriented approaches led to the idea of applying UML to the design of MAS and it is widely applied for the purposes of agent-oriented software engineering. From a software point of view, agent-based systems can be regarded as a specialization of object-oriented systems but UML does not provide enough means for capturing in fully degree all agents related modelling aspects like autonomy, pro-activity and cooperation. The software agents have their own thread of control, localizing not only code and state but their invocations as well. The UML's notations are extended to reflect the characteristic properties of the agents. OMG and FIPA (FIPA, 2010), (Bauer, 1999) are aiming to increase the acceptance of agent-based technology in industry by relating to de-facto standards for object-oriented software development and supporting the development environment throughout the full system lifecycle.

There are different extensions of UML trying to extend different aspects and in different manner the object-oriented modelling language. For example MAS-ML extends UML on the base of TAO (Taming Agents and Objects) meta-model (Silva et al., 2004), Different often used methodologies for development of MAS are supported by UML and AUML, such as for example Tropos, SABPO, Prometheus, MESSAGE, MaSE, PASSI and GAIA (only conceptually). A short overview of the most of them is given in [Mellouli et al., 2004], [Dam & Winikoff, 2003]. In [Sudeikat et al.] a genealogy of the most popular MAS methodologies is proposed in order to examine the gap between modelling and platform.

The mostly used and popular extension of UML for specification of agent-based systems is called AUML (Agent UML). AUML attempts to extend UML with agent-specific concepts and notations in order to model agents and interactions between them. The most significant are connected with the development of Agent Interaction Protocol (AIP) and are in the field of sequence diagrams, interaction overview diagrams, communication diagrams and timing diagrams. Interactions as key component of MAS are represented by set of rules known as interaction protocol, based on Speech Act Theory using verbs and contents [FIPA Modelling TC, 2003]. The communication protocols, which are brightly applied in distributed systems, are not suitable for describing the interactions in MAS, because of their process orientation [Lind, 2002]. The definition of interaction protocols is part of the specification of the dynamical model of an agent-based system. In UML, this model is captured by interaction diagrams (sequence and collaboration diagrams), statechart and activity diagram [Odell et al., 2000]. While the sequence diagram is one of the most commonly used for this purpose diagrams, then the statechart is not commonly used to express interaction protocols because it is a state-centric view, rather than an agent- or process-centred view. Our approach uses the activity diagram in order to work with specifications with very clear processing - thread semantics [Odell et al., 2000], [Bauer et al., 2004]. The Activity Diagram differs from the Interaction Diagrams because it provides an explicit thread of control. It is particularly useful for complex interaction protocols that involve concurrent processing. In UML2.x Activity Diagram, agents can be allocated to one dimension and locations are represented by orthogonal dimension so that the behaviour of multi-agents are easily captured and visualized. Signal sending and Signal receipt in UML Activity Diagrams enables agents to exchange messages, and synchronization of messages is achieved only when the parameters (message description) are exactly the same between signal sending and expecting signal receipt. Communications between classes are described by Agent Communication Language (ACL) messages using the pre-defined stereotype <<ACLMessage>>.

Based on the basic standards, methods and tools described and analysed in this and the previous two parts as well as their advantages and disadvantages for the development of advanced multi-agent based control systems, in the next part of the chapter the core idea and most important steps of the proposed approach are briefly presented.

5. Shortly description of suggested approach

The approaches in the area of distributed control systems existing till this moment are based only on UML/SysML or on the combined use of UML and IEC-61499 standard, as described above. In the approach using UML/SysML profile the SysML stereotypes define new modelling constructs by customizing the existing UML constructs with new properties and constructs in order to create a framework which is a dynamic evolutionary environment, providing traceability and consistency. A SysML specification would be a much better start for the system development than a specification in natural language. But there is a fundamental difference between UML and SysML in the sense that UML models for software systems are intended to employ the same concepts during the complete development phases, reflecting the final software. Through encapsulated objects in UML 2.x and hierarchical structure which they gain using SysML profile the OO systems get near to agent-based systems.

SysML like UML is not a methodology and there is a need to choose or suggest an approach, which to be applied for the development processes. There are some successfully applied Model Based Systems Engineering (MBSE) methodologies such as Harmony SE, INCOSE Object-Oriented System Engineering (OOSE) Methodology, and IBM Rational Unified Process for SE (RUP SE), Vitech methodology, State Analysis (SA) methodology etc. A survey of these methodologies is given in [Estefan, 2007]. The proposed methodology is based on Harmony-SE, which is a subset of a large methodology for integrated system and software development process [Hoffman, 2006]. The task flow development process includes the following basic cycles as shown in Fig.5: Requirements Analysis; System functional analysis; Architecture design; Hardware/Software design specification.

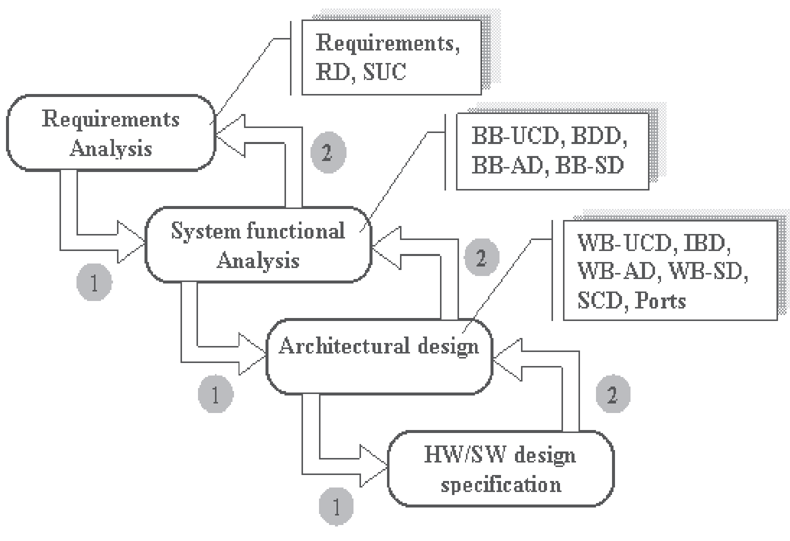


Fig. 5. Harmony SE based MBSE methodology

The proposed methodology uses service-request-driven modelling approach based on SysML structure diagrams using blocks as basic structural elements. The requirements analysis phase starts with the analysis of process inputs. Customer requirements are translated into a set of requirements that define what the system must do and how well it must perform. For the purposes of this first stage the SysML requirements diagrams (RD) to create taxonomy of the captured requirements and System Use Cases (SUC) are used. The system functional analysis phase is presented through transformation on the identified functional requirements into coherent system functions. Use Case Diagram (UCD) present the system functional phase. For each use case the analysis is performed. BDD and IBD are applied to present the composite system structure. Different Black-Box Activity (BB-AD) and Sequence diagrams (BB-SD) are used to capture the high level system functionality. The Architecture design is the third stage in the proposed methodology and includes the design of subsystem structures and behaviour based on White-Box (WB) variants of the diagrams used in the previous stage (WB-UCD, WB-AD, WB-SD). Other important tasks are the ports and interfaces definitions and the description of subsystem state-based behaviour using the SysML statecharts (SCD). The last stage is the Hardware/Software design specification and is closely connected with the system implementation. All methodology stages include verification and validation tasks, based on the model developed in the previous stages and the defined requirements.

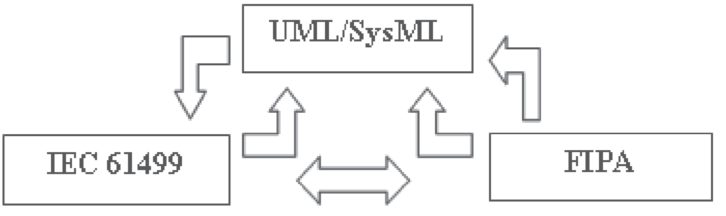


Fig. 6. Research framework

The methodology described above is used to model the internal structure of control system and to define how this system can be mapped to physical entities. However the standards used aren't enough to model the communications between Human Machine Interface (HMI) and control systems. For this reason the FIPA reference model is suitable to be used. The proposed research framework is shown in Fig. 6.

The FIPA agent reference model was chosen in order to provide a normative framework within which agents can be deployed and operate. FIPA specification establishes the logical reference model for creation, registration, location, communication, migration and retirement of agents. FIPA standard do not attempt to prescribe the internal architecture of agents nor how they should be implemented, but it specifies the interfaces necessary to support interoperability between agent systems. The FIPA modelling standard for development of multi-agent systems uses Activity Diagram as part of AUML. According Lind, we adopted the SysML Activity Diagram to model agent interaction protocol without introducing any new modelling elements [Lind, 2002]. The approach uses existing UML concepts only and requires no additional modelling elements, which makes it easy for UML users to understand the notation without learning a completely new type of diagram. For this case, to specify the agent communication protocols, two predefined stereotypes: <<ACLMessage>> and <<role>> are used.

6. Case study

The suggested approach will be illustrated in this part with a simple example of distributed control system for Distribution Station that is a part of didactic test bench, manufactured by FESTO Corp. and is located at the Martin Luther University of Halle-Wittenberg. The Distribution Station, shown in Fig. 7, separates work pieces from a magazine and then feeds them for processing. The Distribution Station is decomposed into two modules - the feed magazine, known as Storage module and the transfer module. The Storage module separates work pieces from the magazine by a double acting cylinder, which pushes out the bottom work piece from the gravity feed magazine. The end pusher positions and the filling level of the magazine are monitored by means of binary sensors. The magazine might be fed with work pieces in any order. The Transfer Module is a pneumatic handling device. The work pieces are picked by a vacuum suction cap and transferred in a range of 0º to 180º by a swivel drive. The available sensors and actuators for both modules are shown in Table 1. The distribution station is controlled by an individual controller, connected to the controller of another stations via an I/O interface. The distribution station contains console with some buttons, such as “Start”, “Stop”, “Reset” and “Ack” for achieving different operation modes of the station through operator interventions. Both mechatronic modules are controlled separately by distributed controllers that communicate with each other. The scenario for distributed control follows the idea, submitted in [Vyatkin et al., 2006] according to which the functioning of both controllers is based on the mutually exclusive admission to the end position of the feeder unit (from the Storage module), where work pieces are picked up by the Transfer module.

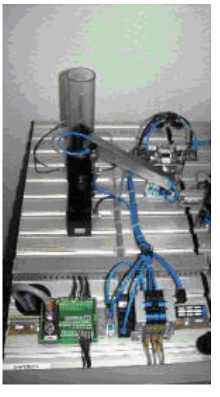


Fig. 7. View of the Distribution station

Inputs and outputs of the Feed Magazine module			
Outputs		Inputs	
DI 0.0	Pusher retracted	DO 0.0	Extend pusher
DI 0.1	Pusher extended		
DI 0.6	Magazine empty		
Inputs and outputs of the Transfer module			
Outputs		Inputs	
DI 0.2	Swivel drive at magazine	DO 0.1	Drive to pos. magazine
DI 0.3	Swivel drive at testing station	DO 0.2	Drive to pos. testing station
DI 0.5	Work pieces sucked in	DO 0.3	Vacuum off
		DO 0.4	Vacuum on

Table 1. Sensors and actuators of the feed magazine and transfer modules

The first two stages in the design of control system according to the suggested approach are the definition of system requirements through the Requirements Diagram (Fig.8-1) and system functionality definition using Use Case Diagram (Fig.8-2) that captures the interactions between the system and its users and the requirements refinements to the distributed control application. The static composite structure of the system consisting of distributed controllers, console, distribution station and connections between them is presented through BDD diagram (Fig.8-5). It corresponds to a Subapplication (Application) from the IEC-61499 standard. The basic elements of this diagram are presented as "system blocks" and their analogue according IEC-61499 is the CFB. The "parts" nested in these system blocks correspond to BFB. The third development stage includes modelling of the internal structure of the system blocks included in the overall static structure using BDDs and IBDs. The internal structure of the station is presented as BDD that covers the system hierarchy described through composition relationships (Fig.8-4). The message exchange among the system blocks is presented by Sequence Diagrams as shown in Fig.8-6. The model view browser as planned according the suggested approach is presented in Fig.8-3.

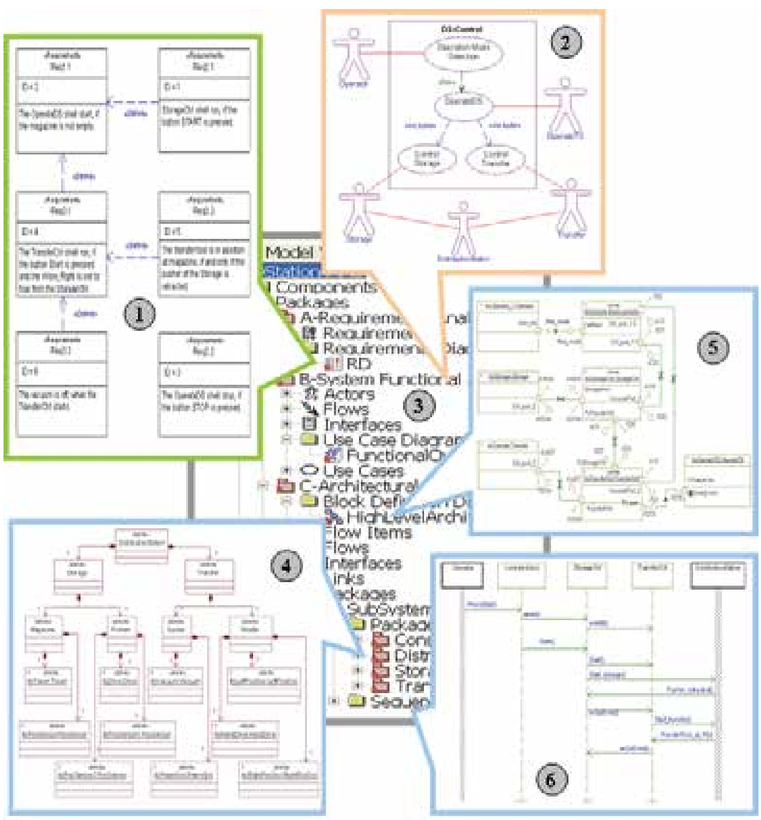


Fig. 8. Illustration of the first stages of the proposed approach

Both distributed controllers "StorageCtrl" and "TransferCtrl", modelled as "system blocks", are decomposed in two main "parts": "StorageCtrl_FB" and "E_Restart", and "TransferCtrl_FB" and "E_Restart" respectively. The controller internal structure covers the system hierarchy described through composition relationships and it is modelled through IBD as shown in Fig.9-1 for the "StorageCtrl" and in Fig.9-2 for the"TransferCtrl". Both parts in IBDs are modelled by parts with "BasicFunctionBlock" stereotype. The description of part "StorageCtrl_FB" includes the following attributes - *Empty, Pos_E, Pos_R, Retract, Left_OK, Right_OK, Allow_Left* and *Allow_Right* representing the input and output data and operations, representing the input (*Init, Reset, Start, Sense, Stop, In_Cmd*) and output events (*Init0, Act* and *Out_Cmd*). The attributes of "TransferCtrl_FB" part are the following: *at_Mag, at_TS, Vac_on, Left_OK, Right_OK, to_Mag, to_TS, VCM_on, VCM_off*. The corresponding operations are the same and with the same meanings as for the "StorageCtrl_FB". The "E_Restart" part, used in both controllers, represents one restart event generator providing single events when the resource Cold and Warm starts and also when the resource is stopped by some external agents. It is likely that most applications will require at least one instance of this block to initiate the triggering event that starts execution of a chain of function blocks in a network.

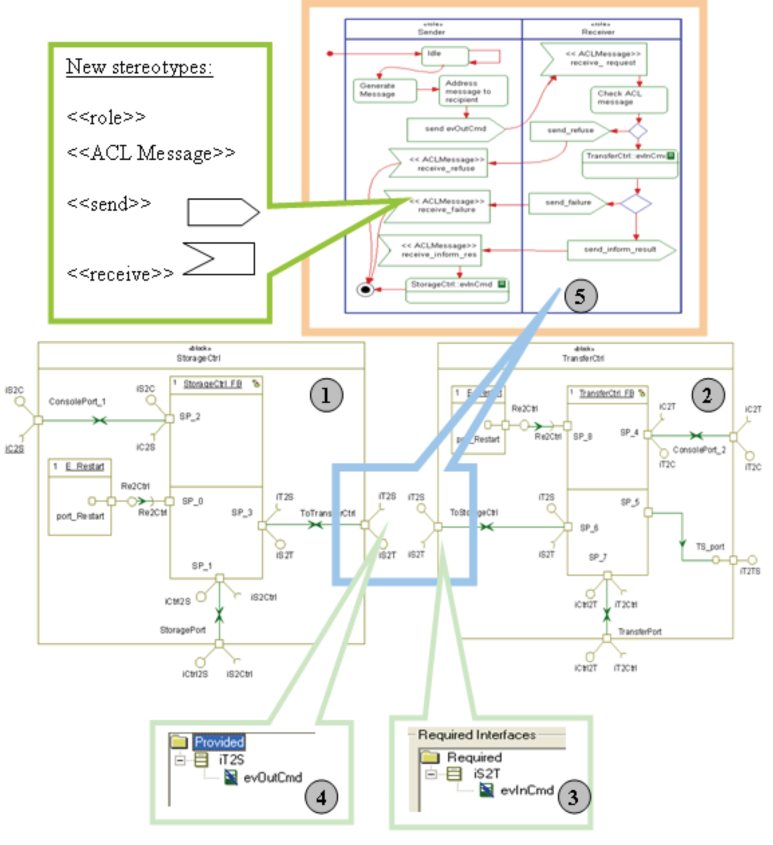


Fig. 9. IBD of the IEC-61499 based distributed controllers "StorageCtrl" and "TransferCtrl"

According to IEC-61499 standard, the link between function blocks and the interfaces of resources is provided through service interface FB (SIFB's). Communication interfaces in UML2.x can be defined by different types of ports. Contract-based ports are used in order to define contracts, which specify the precise allowable inputs and outputs of system components. The contract-based port between distributed controllers as shown in Fig.9 have two types of interfaces: provided (Fig.9-4) and required (Fig.9-3). The provided interface characterizes the requests from the environment and the required one – the request from the port. Because the ports are connected to other ports which are exactly "reversed" and with ports providing and requiring the same set of interfaces, the explicit contracts are used.

The FIPA "request when" protocol, as a kind of agreement for information exchange in a distributed system, can be used for peer-to-peer interactions between distributed controllers in order to obtain information or to request the execution of a task to other agents. This interaction protocol is modelled through activity diagram as behaviour of interface used and is illustrated in Fig.9-5. The diagram is customized through the introduction of stereotype <<role>>, associating swim lanes with roles. There are two roles in the FIPA "request when" protocol: sender and receiver, which are linked via implicit communication channels. In a protocol context the communication messages are grouped representing the sender – receiver direction. The protocol is used by the "sender" role of agent to request that "receiver" role performs some action at the time a given precondition becomes true (input event "evInCmd"). If an explicit agreement is required, the "receiver" can "refuse" to perform the action or "agree" to perform the action. If the "receiver" refuses, no further communication needs to take place. If the "receiver" agrees, it will attempt to perform the action when the precondition becomes true and then communicate to the "sender" one of the following: "failure" (the "receiver" failed to complete the requested action) or "inform-result" (the "receiver" has completed the action and notifies the "sender" of the result). According to FIPA reference model a class with stereotype "ACLMessage" is defined for representing of Agent Communication Language messages. Sending and receiving activities in this diagram allow agents to exchange messages based on communication protocols. The main benefits of using the FIPA Agent Interaction protocol is that it contains unified agreements on the methods used for initiation and termination of protocol data units (message), formatting and encoding data, synchronization of senders and receivers, and detection and correction of transmission errors. The protocol specification consists of the following parts: service specification, assumption about the environment, precise message format for (syntax), procedure rules for data exchange (grammar) and a vocabulary of messages used with their meaning (semantics).

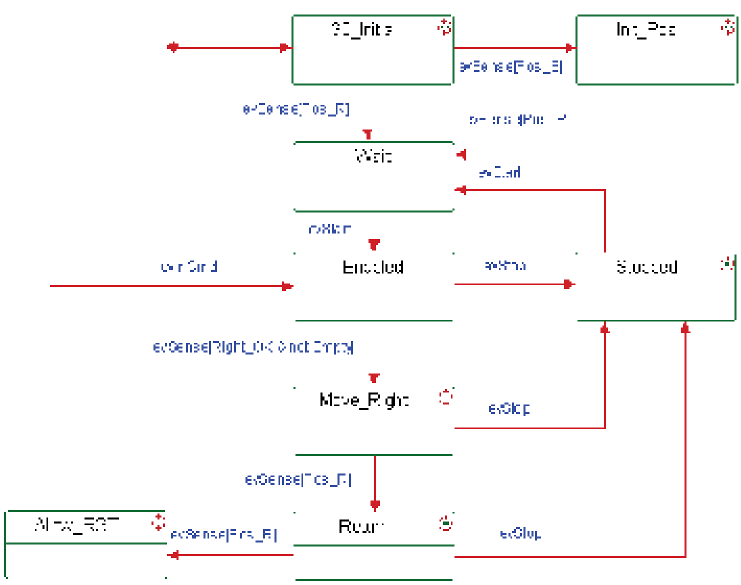


Fig. 10. Statechart of the "StorageCtrl_FB" system block

The system behaviour of basic FB according IEC-61499 standard is specified through Execution Control Chart (ECC). The functionality of ECC may be achieved for the system behaviour of IBD parts using SysML statecharts. The statecharts for both parts – "StorageCtrl_FB" and "TransferCtrl_FB" are shown in Fig.10 and Fig.11 respectively.
The statechart of part "StorageCtrl_FB" describes its basic states and the relations between them, modelled through the so called transitions. The control instance is in initial state when it is first created. From this state, after occurrence of input event "evInit", "StorageCtrl_FB" passes in initialization state "S0-Initial" where the controller memory is initialized and of the output data "Allow_Right" and "Allow_Left" is assigned value "false" in the frame of the algorithm "Alg_Init". From this state the object can move into two new states "Init_Pos" and "Wait". If the pusher is heading "back", i.e. the transition condition Sense[pos_R] is valid, control changes to state "Wait", which does not perform any actions. Otherwise, i.e. if the transition condition Sense[pos_E] is "true", the object "StorageCtrl_FB" goes into state "Init_Pos". In this state an action is initiated resulting in invoking of algorithm "Alg_Retract1" according that of the output data "retract" the value "true" is assigned (retract:= 1) and the output event "evAct" is generated (OUT_PORT (SP_1) -> GEN (evAct)), connected with activating the actuator to bring the pusher into position "retract". By satisfying the transition condition Sense[pos_R] (i.e. when the pusher is already in position "retract"), the object goes into a state of waiting "Wait". From that state the control object may be out after the occurrence of event "evStart", and than is able to go in the state "Enable", in which the object falls from the state "Allow_RGT" also, after the occurrence of "evInCmd" event. The control object can leave the state "Enable" in two ways: either after the occurrence of event "evStop", after that is passing in a state "Stopped", or by satisfying the guard condition [Right_OK & not Empty] (which means that the part can be manipulated by the swivel drive and the magazine is not empty) and than is passing in the state "Move_Right". In this state, on the base of algorithm "Alg_Retract0" the output data "retract" gets value "false", and on the base of "Alg_AllowRight0", the output data "Allow_Right" – the value "true" by generating output events "evAct" and "evOutCmd", respectively associated with the data. From this state, by occurrence of input event "evStop", the object falls in the already known state "Stopped", and by the occurrence of event "evSense" and guard condition [pos_E] (pusher is extended), the object passes in a state "Return", where the algorithm "Alg_Retract1" is running and the output data "retract" gets the value "true" after that the output event "evAct" is generated in order to return the pusher. As in previous states, and here by an occurrence of event "evStop", the object is passed to state "Stopped" where an algorithm "Alg_Retract1" (retract:=0) is executed and after that the output event "evAct" is generated. If an event "evSense" occurs in state "Return" and the guard condition [pos_E] is true (pusher is extended), the object passes into the state "Allow_Right", in which the variable "Allow_Right" gets value "true" as a result of the execution of "Alg_AllowRight1" algorithm and after that the output event "evOutCmd" is generated.
The statechart of part "TransferCtrl_FB" shown in Fig.11, starts from the initial state and after occurring of input event "evInit", the control object goes into state "Init". In this state the algorithm "Alg_Init" is invoked and after that an action on exit is executed, connected with the generating of output event "evInit0". From this state the object can move into two new states "Init_Pos" and "Enabled". If the swivel drive is not in position "at_Mag" and the left standing operation is completed, i.e. the guard condition [at_TS and Left_OK] is satisfied, the state "Init_Pos" is reached. In this state after the execution of algorithm

“Alg_ToMag1”(To_Mag:=1), an action “evAct/To_Mag:=1” is initiated in order to return the swivel drive in position at magazine. In this state a second algorithm “Alg_AckLed1” is invoked too and is connected with assignment of value “true” to the output data [Ack_led]. By satisfying the guard condition [at_Mag], the object goes into state “Enabled”. From state “Init_Pos” the control object may be passed to state “Stopped” upon the occurrence of an event “evStop”. In this state 3 algorithms are invoked: “Alg_ToTS0” (To_TS:=0), “Alg_ToMag0” (To_Mag:=0) and “Alg_StartLed1” (Start_led:=1). After the execution of the first of them, the output event “evAct”/To_TS:=0 is initiated. The state “Stopped” may be terminated by satisfying the guard condition [not Right_OK] after that the object is switched to state “Enabled” where the algorithm “Alg_AllowLeft0” (Allow_Left:=0) is executed. From state “Enabled”, after pressing the button “Start” (trigger “evStart”), the state “Set_Led” is activated, where the [Right_OK] is not valid. In this state the algorithm “Alg_AckLed0” is running that assigns the value “false”to the output data [Ack_led] and generates the output event “evAct/Ack_led:=0”. By satisfying the guard condition [Right_OK] the object enter the state “Move_TS”, where after the execution of algorithm “Alg_ToTS1” (To_TS:=1), the output event “evAct” is initiated, that drives the transfer tool to position testing station. From this state the object can move into two alternative states: "Stopped" and "Hold". In state “Hold” the algorithms “Alg_AckLed0” (Ack_led:=0), “Alg_ToTS0” (To_TS:=0) and “Alg_AllowLeft1” (Allow_Left:=1”) are executed, after that the actions “evAct/To_TS:=0” and “evOutCmd/Allow_Left:=1” are initiated in order to move the transfer tool to the Magazine and to enable the StorageCtrl_FB. From this state, by occurrence of input event "evStop", the object falls in the already known state "Stopped" and by occurrence of “evInCmd/Left_OK:=1” the object is passing in state “to_Mag” where two combinations of algorithms and actions are executed: the first one includes the algorithm “Alg_AllowLeft0” (Allow_Left:=0) and the action “evOutCmd/Allow_Left:=0”, and the second “Alg_ToMag1” (to_Mag:=1) and “evAct/to_Mag:=1”. By activating the transition condition [at_Mag:=1] the object is going to the next state “Suck_in”, where the algorithm “Alg_VacOn1” (VCM_on:=1) is executed and the output event “evAct/VCM_on = true” is generated, where the work piece is picked by a vacuum suction cap. Another possible state is the “Stopped” state, appearing by pressing the button “Stop” (trigger “evStop”). If an event “evSense” occurs and the guard condition [Vac_on] is true, i.e. work piece is sucked in, the object passes into the state “To_TS” in which the algorithm “Alg_ToTS1” (To_TS:=1) is executed and the output event “evAct/To_TS:=1” is generated i.e. the swivel drive transfers the work piece to the Testing Station. After satisfying the guard condition [at_TS] the state “Drop” is achieved and the algorithm “Alg_VacOff” (VCM_off:=1) is executed and the output event “evAct/VCM_off:=1” is generated that is connected with release of the work piece. The state “Return_Mag” is reached from the state “Drop” in case that [Vac_on] = false and in this state the algorithm “Alg_ToMag1” (to_Mag:=1) is executed and output event “evAct/to_Mag:=1” is generated, which returns the swivel drive at magazine. If the event “evSense” occurs and the guard condition [at_Mag] is satisfied, the next state “Wait” is reached. In this state the algorithm “Alg_AckLed1” (Ack_Led:=1) is executed and after that the output event “evAct/ Ack_led:=1” is generated. A second algorithm “Alg_AllowRight1” (Allow_Right:=1) is performed too, after that the output event “evOutCmd/Allow_Right:=1” is initiated. From the state “Wait” by occurrence of events “evAck” or “evNotSingle” the control object is moving to state “Set_Led” and the module is ready to carry the same sequence of actions and related algorithms. The last four states include also transitions to state “Stopped” in the case of the occurrence of input event “evStop”.

Algorithms which are executed in some states of the above described statecharts may be presented with Activity Diagrams. For this case study the algorithms used are very simple and there is not a need to model them.

7. Conclusions

The suggested methodology for development of multi-agent control systems based on the combined use of UML/SysML, FIPA and IEC-61499 standards is suitable for development of open, interoperable, re-configurable and distributed control system. This methodology allows describing the whole live-cycle of the control system and achieving software encapsulation and re-use of the defined entities (agents). One of the most essential features of this approach is that control engineers are able to model the closed loop control system and to apply the different type of analysis techniques in order to determine whether these models meet their performance and schedulability requirements, without requiring a deep understanding of the inner working of those techniques.

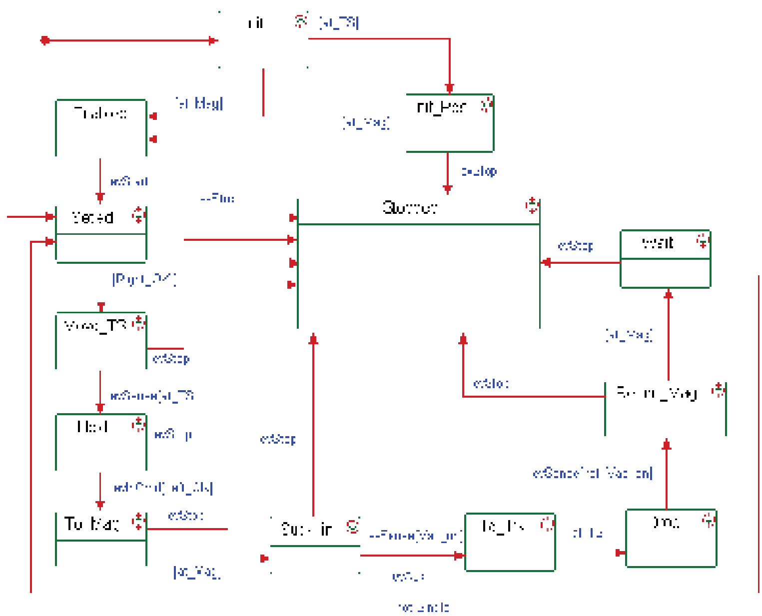


Fig. 11. Statechart of the “TransferCtrl-FB” part

The UML/SysML profile is extended and at the same time restricted with different stereotypes from the FIPA communication standard in order to fill the gap from absence of unified communication protocol in applications based on the IEC-61499.

The main advantages of the proposed approach may be summarized as follows:

- SysML supports the whole life cycle by the development of control engineering applications from the requirements definition to the software implementation;
- The possibilities to model the physical systems in detail enhance the procedures for analysis, testing and validation of designed closed loop behaviour of the system;
- Based on extended activity diagrams with FIPA standard, the proposed approach get near to agent-based systems.

8. Acknowledgment

This work has been partially supported by the Bulgarian National Research Fund to the Ministry of Education and Science in the frame of Project ВУ-TH-208.

9. References

Apvrille, L., Courtiat, J.-P., Lohr, Ch. & Saqui-Sannes, P. (2004). TURTLE: A Real-Time UML Profile Supported by a Formal Validation Toolkit, *IEEE Transactions on Software Engineering*, Vol. 30, No. 7, July 2004, pp. 473-487

Bauer, B. (1999). Extending UML for the Specification of Interaction Protocols, submitted for *the 6th Call for Proposal of FIPA*.

Bauer, B., Muller, J. P. & Odell, J. (2001). Agent UML: A Formalism for Specifying Multi-agent Software Systems, *In Proceedings of the First International Workshop on Agent-Oriented Software Engineering AOSE'00*, Limerick, Ireland, pp. 91-103, LNCS 1957 Springer

Bianco, V., Lavazza, L. & Mauri, M. (2001). An introduction to the DESS approach to the specification of real-time software, *CEFRIEL Technical Report RT01002*, April

Bianco, V, Lavazza, L. & Mauri, M. (2002). A Formalization of UML Statecharts for Real-Time Software Modelling, *6th Biennial World Conference on Integrated Design Process Technology (IDPT2002), "Towards a rigorous UML"*, Pasadena, California, June 23-28.

Bussmann S., McFarlane D.C. (1999), Rationales for holonic manufacturing control, *Proc. of the 2nd International Workshop on Intelligent manufacturing Systems,* pp.177-184, September, Leuven, Belgium.

Damm, W., Josko, B., Pnueli, A. & Votintseva, A. (2002). Understanding UML: A Formal Semantics of Concurrency and Communication in Real-Time UML. *First International Symposium, FMCO 2002*, Leiden, The Netherlands, November 5-8, 2002, Revised Lectures, pp. 71-98

Dam, K. H. & Winikoff, M. (2003). Comparing Agent-Oriented Methodologies. in: *Proceedings of the Fifth International Bi-Conference Workshop on Agent-Oriented Information Systems*, Melbourne, July (at AAMAS03).

DiPippo, L. C. (2000). A UML Package for Specifying Real-Time Objects, *Computer Standards & Interfaces*, Volume 22 , Issue 5 (December 2000) pp. 307–321,

Douglass, B. (1998). *Real-Time UML: Developing Efficient Objects for Embedded Systems*, Addison-Wesley Longman Publishing Co., Inc., ISBN:0201325799, Boston, MA,USA

Dubinin, V. & Vyatkin, V. (2004). UML-FB – A Language for Modelling and Implementation of Industrial-Process Measurement and Control Systems on the Basis of IEC 61499 Standard, *Proc. of the 6-th International Conference of Science and Technology "New Information Technologies and Systems (NITS'2004),* pp.77-83, Penza, Russia, Part 2, June 17-19

Durfee, E. H. & Lesser, V. (1989). Negotiating task decomposition and allocation using partial global planning, in: L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence Volume II.* Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA, 1989, pp. 229–244.

Estefan, J. A. (2007). Survey of model-based system engineering (MBSE) methodologies, pp.1-47, *INCOSE MBSE Focus Group,* May 25

FIPA (2010). Foundation for Intelligent Physical Agents, http://www.fipa.org

FIPA Modeling TC (2003). FIPA Modelling Area: Deployment and Mobility, *FIPA Technical report*. www.auml.org/auml/documents/DeploymentMobility.zip

Gerard, S., Terrier, F. & Tanguy, Y. (2002). Using the Model Paradigm for Real-Time Systems Development: ACCORD/UML, *Proc. Conf. Advances in Object-Oriented Information Systems, OOIS 2002 Workshops*, pp. 260-269, ISBN:3-540-44088-7, Springer-Verlag, London, UK

Graf, S. & Ober, I. (2003). A real-time profile for UML and how to adapt it to SDL, *In Proceedings of the 11th international conference on System design*, pp. 55-76, ISBN ~ ISSN:0302-9743, 3-540-40539-9 Springer Verlag, Berlin, Heidelberg: http://citeseer.ist.psu.edu/graf03realtime.html.

Hoffman, H.-P. (2001). Harmony-SE/SysML Deskbook Model-based Systems Engineering with Rhapsody, Rev.1.51, *Telelogic/I-Logix whitepaper*, May 24.

Huynh, T. & Osmundson, J. (2005). A System Engineering Methodology for Analysis Systems of Systems Using the System Modelling Language, Proceeding *of 1st Annual System of System Engineering Conference*, Johnstown, PA, June 13-14,

IEC-61499 (2005). *International Standard IEC-61499, Function Blocks, Part 1 - Part 4*, International Electrotechnical Commission (IEC), Technical Committee TC65/WG6, IEC Press, Jan.

Jennings N. R. & Wooldridge M. (1998). Applications of Agent Technology, in: N. R. Jennings and M. Wooldridge, editors, *Agent Technology: Foundations, Applications, and Markets*. Springer-Verlag, March 1998.

Kesten, Y. (1992). Timed and Hybrid Statecharts and Their Textual Representation, Formal Techniques in Real-Time and Fault-Tolerant Systems, *2nd International Symposium,* pp. 591 – 620, ISBN: 3-540-55092-5, Springer-Verlag, London, UK

Lavazza L., Quaroni G. & Venturelli M. (2001). Combining UML and formal notations for modelling realtime systems. V. Gruhn (ed.) *Proceedings of ESEC/FSE 2001*, Vienna, Austria, September 10-14, ACM Press

Lind, J. (2002). Specifying Agent Interaction Protocols with Standard UML, *In 2nd International Workshop on Agent-Oriented Software Engineering AOSE'02*, pp. 136-147, LNCS 2222 Springer

Medina, J. L., Drake, J. M. & González, M. H. (2001). UML-MAST: Modelling and Analysis Methodology for Real-Time Systems Developed with UML CASE Tools, *Proceedings of the Euromicro Conference on Real-Time Systems*, Delft, The Netherlands, June

Mellouli, S.; Moulin, B. & Mineau, G. W. (2004), Towards a Modelling Methodology for Fault-Tolerant Multi-Agent Systems, in: *Informatica Journal* 28, pp.31-40.

Meunier, J.-N., Lippert, F. and Jadhav, R. (2003). RT modeling with UML for safety critical applications: the HIDOORS project example, Proceedings of SVERTS 2003, Co-located with UML 2003, San Francisco, CA, U.S.A., October.

Miguel, A. (2003). General Framework for the Description of QoS in UML. *In EEE Computer Society, editor, Sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'03)*, pp 61–70, Hakodate, Hokkaido, Japan, May

Mwaluseke, G. W. & Bowen, J. P. (2001). UML Formalisation Literature Survey, 6 September,

Odell, J., Parunak, H. & Bauer, B. (2000). Extending UML for Agents, In Proceedings of the Agent-Oriented Information Systems Workshop - AOIS 2000, pp. 3-17, Eds., Austin

Odell, J.; Parunak, H. V. D. & Bauer, B. (2001). Representing Agent Interaction Protocols in UML, *Agent-Oriented Software Engineering*, Paolo Ciancarini and Michael Wooldridge eds., Springer-Verlag, Berlin, pp.121–140.

OMG (2003). UML Profile for Performance, Schedulability and Time, *Specification OMG*,

OMG-UML (2010).Unified Modelling Language Specification, June: http://www.omg.org/

OMG-UML (2003). UML for Systems Engineering, Request for Proposals - RFP ad/2003-3-41, *Object Management Group*, Needham, September.

OMG-SysML (2006). The OMG Systems Modelling Language, May http://omgsysml.org/index.htm

Parunak, H. V. D. (1999). Practical and Industrial Application of Agent-Based Systems, in: *Multiagent Systems*, G. Weiss, ed., Cambridge: The MIT Press.

Rhapsody, http://modeling.telelogic.com/standards/

Selic, B. (1998). Using UML for Modeling Complex Real-Time Systems, *Proceedings of the ACM SIGPLAN Workshop on Languages, Compilers, and Tools for Embedded Systems*, pp. 250 – 260, ISBN 978-3-540-65075-1, Springer-Verlag, London, UK

Selic, B. (2002). *The Real-Time UML Standard: Definition and Application*, IEEE Computer Society, ISSN:1530-1591, Washington, DC, USA

Selic, B. (2004). Tutorial: An Overview of UML 2.0, *Proceedings of the 26th International Conference on Software Engineering (ICSE'04)*, IBM Rational Software Canada

Silva, V.; Choren, R. & Lucena, C. (2004). A UML Based Approach for Modelling and Implementing Multi-Agent Systems. In: *Proceeding of the third International Conference on Autonomous Agents and Multi-Agents Systems*, USA.

Sudeikat, J.; Braubach, L.; Pokahr, A., & Lamersdorf W. (2004). Evaluation of Agent–Oriented Software Methodologies – Examination of the Gap between Modelling and Platform, in: *Agent-Oriented Software Engineering V, Fifth International Workshop AOSE*, pp.126-141.

Telelogic AB (2000). *User Guide Rhapsody*, Rhapsody Tool.

Thramboulidis, K. C. (2001). Using UML for the Development of Distributed Industrial Process Measurement and Control Systems, *IEEE Conference on Control Applications (CCA)*, September, Mexico.

Thramboulidis, K. C. (2004). Using UML in Control and Automation: A Model Driven Approach, *2nd IEEE International Conference on Industrial Informatics INDIN'04,* 24^{th} - 26^{th} June, Berlin, Germany.

Parunak, H. V. D. (1999). Practical and Industrial Application of Agent-Based Systems, in: *Multiagent Systems*, G. Weiss, ed., Cambridge: The MIT Press.

Vyatkin, V., Hirsch, M. & Hanisch H.-M. (2006). Systematic Design and Implementation of Distributed Controllers in Industrial Automation, *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA´2006) proceedings*, pp.633-640, Prague, Czech Republic.

Verschaeve, K. (1999). Combining UML and SDL, *Eighth SDL Forum*, Montréal, Canada,

Wooldridge, M., Jennings, N. R. & Kinny, D. (2000). The Gaia Methodology for Agent-Oriented Analysis and Design, *International Journal of Autonomous Agents and Multi-Agent Systems*, 3(3):285-312.

5

Stackelberg Solutions to Noncooperative Two-Level Nonlinear Programming Problems through Evolutionary Multi-Agent Systems

Masatoshi Sakawa, Hideki Katagiri and Takeshi Matsui
Faculty of Engineering, Hiroshima University
Japan

1. Introduction

In the real world, we often encounter situations where there are two or more decision makers in an organization with a hierarchical structure, and they make decisions in turn or at the same time so as to optimize their objective functions. Decision making problems in decentralized organizations are often modeled as Stackelberg games (Simaan & Cruz Jr., 1973), and they are formulated as two-level mathematical programming problems (Shimizu et al, 1997; Sakawa & Nishizaki, 2009). In the context of two-level programming, the decision maker at the upper level first specifies a strategy, and then the decision maker at the lower level specifies a strategy so as to optimize the objective with full knowledge of the action of the decision maker at the upper level. In conventional multi-level mathematical programming models employing the solution concept of Stackelberg equilibrium, it is assumed that there is no communication among decision makers, or they do not make any binding agreement even if there exists such communication. Computational methods for obtaining Stackelberg solutions to two-level linear programming problems are classified roughly into three categories: the vertex enumeration approach (Bialas & Karwan, 1984), the Kuhn-Tucker approach (Bard & Falk, 1982; Bard & Moore, 1990; Bialas & Karwan, 1984; Hansen et al, 1992), and the penalty function approach (White & Anandalingam, 1993). The subsequent works on two-level programming problems under noncooperative behavior of the decision makers have been appearing (Nishizaki & Sakawa, 1999; Nishizaki & Sakawa, 2000; Gumus & Floudas, 2001; Nishizaki et al., 2003; Colson et al., 2005; Faisca et al., 2007) including some applications to aluminium production process (Nicholls, 1996), pollution control policy determination (Amouzegar & Moshirvaziri, 1999), tax credits determination for biofuel producers (Dempe & Bard, 2001), pricing in competitive electricity markets (Fampa et al, 2008), supply chain planning (Roghanian et al., 2007) and so forth.
However, processing time of solution methods for noncooperative two-level linear programming problems, for example, Kth Best method by Bialas et al. (1982) and Branch-and-Bound method by Hansen et al. (1992), may exponentially increases at worst as the size of problem increases since they are strict solution methods based on enumeration. In order to obtain the (approximate) Stackelberg solution with a practically reasonable time, approximate solution methods were presented through genetic algorithms (Niwa et al., 1999) and particle swarm optimization (PSO) (Niwa et al., 2006).

As one of the most promising approximate solution methods, Socha et al. (2002) proposed a fast computational method through an evolutionary multi-agent system (EMAS) for obtaining (approximate) Pareto optimal solution sets for multiobjective programming problems. However, there is no study on the EMAS-based method for solving two-level nonlinear programming problems.
In this chapter, we propose an efficient EMAS-based computational method for obtaining (approximate) Stackelberg solutions to two-level nonlinear programming problems.

2. Two-level programming problems and solution concepts

In this chapter, we consider two-level programming problems formulated as follows:

$$\left.\begin{array}{l} \underset{x_1}{\text{minimize}}\ f_1(\boldsymbol{x}_1, \boldsymbol{x}_2) \\ \quad \text{where } \boldsymbol{x}_2 \text{ solves} \\ \underset{x_2}{\text{minimize}}\ f_2(\boldsymbol{x}_1, \boldsymbol{x}_2) \\ \text{subject to } g_i(\boldsymbol{x}_1, \boldsymbol{x}_2) \le 0,\ i = 1, 2, \ldots, m \end{array}\right\} \quad (1)$$

where $\boldsymbol{x}_1$ is an n_1 dimensional decision variable column vector for the DM at the upper level (DM1), $\boldsymbol{x}_2$ is an n_2 dimensional decision variable column vector for the DM at the lower level (DM2), $f_1(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is the objective function for DM1, $f_2(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is the objective function for DM2 and $g_i(\boldsymbol{x}_1, \boldsymbol{x}_2)$, i=1, 2,…, m are constraint functions. In general, $f_l(\cdot)$, l=1,2 and $g_i(\cdot)$, i=1,2, …, m are nonlinear. In (1), if the DM at the upper level (DM1) adopts a decision $\boldsymbol{x}_1$, the DM at the lower level (DM2) is supposed to select a decision to minimize $f_2(\cdot)$ in the feasible region of (1) under the DM1's decision, $\hat{x}_2(\hat{x}_1)$, called a rational reaction. Then, the optimal solution (Stackelberg solution) to (1) is the point $(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*(\boldsymbol{x}_1^*))$ which minimizes $f_1(\cdot)$ in the inducible region (IR) which is the set of points $(\hat{x}_1, \hat{x}_2(\hat{x}_1))$ for all possible decisions $\hat{x}_1$. Figure 1 illustrates an example of a Stackelberg solution for a two-level linear programming problem.

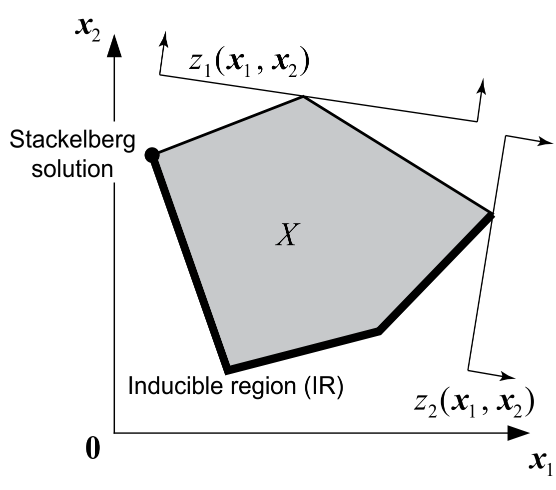


Fig. 1. An example of Stackelberg solution

3. EMAS for two-level programming problems

In this section, we outline the framework of a computational method through EMAS for obtaining Stackelberg solutions to two-level programming problems.

In general, EMAS-based methods consist of N agents $\boldsymbol{a}_r$, r = 1, 2, …, N, each of which is characterized by some attributes, i.e., the current position ($\boldsymbol{x}_{r1}$, $\boldsymbol{x}_{r2}$), the upper level objective function value $f_1(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$, the lower level objective function value $f_2(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$, the upper level energy e_r^U and the lower level energy e_r^L. In the EMAS-based method, the attributes of agents are updated through the evolutionary process with some operations like energy exchange, reproduction and move.

First, carry out the search to the direction of improving the lower level objective function by moving each agent toward IR. To be more specific, in the upper level decision variable space, if there exists at least one agent in the neighborhood of each agent $\boldsymbol{a}_r$, select one of agents, $\boldsymbol{a}_{r'}$, as a communicating opponent, and compare the lower level objective function value of $\boldsymbol{a}_r$ with that of $\boldsymbol{a}_{r'}$. Then, the superior agent gains the lower level energy from the inferior one. The inferior agent is killed if the lower level energy of it becomes empty. Otherwise, it is moved according to some rule whose details are described later in Section 4. If there exists no agent in the neighborhood of $\boldsymbol{a}_r$, move $\boldsymbol{a}_r$ to the position where the lower level objective function value becomes better by changing $\boldsymbol{x}_{r2}$.

Next, carry out the search to the direction of improving the upper level objective function by moving each agent near IR toward a Stackelberg solution. For each agent $\boldsymbol{a}_r$, after selecting an agent, $\boldsymbol{a}_{r''}$, which is the nearest agent around IR as a communicating opponent, compare the upper level objective function value of $\boldsymbol{a}_r$ with that of $\boldsymbol{a}_{r''}$. Then, the superior agent gains the upper level energy from the inferior one. The inferior agent is not killed even if its lower level energy of it becomes empty. It gains the same amount of the upper level energy as the initial value. This supplement is done to maintain the number of agents with nonzero upper level energy. After exchanging the upper level energies between the superior and inferior agents, if the upper level energy of the inferior agent is sufficiently large, generate a new agent in the direction from the inferior one to the superior one.

By repeating these procedures, agents with large upper level objective function values congregate around IR, which means the current position of the agent with the largest upper level objective function value can be regarded as the (approximate) Stackelberg solution.

The procedure is summarized as follows.

Step 1. Generate N agents $\boldsymbol{a}_r$, r = 1,2, …, N at random.

Step 2. Let $T := 1$.

Step 3. Let $r := 1$.

Step 4. For the rth agent $\boldsymbol{a}_r$, carry out the search to the direction of improving the lower level objective function value in order to move the current position of the agent toward IR.

Step 5. If the lower level energy of the rth agent $\boldsymbol{a}_r$ is greater than a threshold, i.e., the current position of the agent is regarded as being in IR, carry out the search to the direction of improving the upper level objective function value so that the agent is moved toward the Stackelberg solution.

Step 6. If $r = N$, go to step 7. Otherwise, let $r := r+1$ and return to step 4.

Step 7. If $T = T_{\max}$, terminate the procedure and the current best solution is regarded as a Stackelberg solution. Otherwise, let $T := T+1$ and return to step 3.

4. EMAS for two-level nonlinear programming problems

This section devotes to introducing some basic ideas of a new EMAS for two-level nonlinear programming problems. In applying the original EMAS directly to the nonlinear case, there often occur two problems; one is that it is difficult to obtain feasible initial agents generated randomly in step 1, and the other is that the judgment of the existence of an agent around IR by the amount of the lower level energy is insufficient in most cases since the shape of IR for nonlinear cases is fairly complicated in general than that for the linear case.

In order to resolve the former problem, we incorporate the idea of homomorphous mapping used in (Koziel & Michalewicz, 1999) into the proposed method in order to generate feasible initial agents. In addition, in order to widen the search area, we permit the reproduction in the infeasible region together with the search of the infeasible region by infeasible agents.

On the other hand, for tacking the latter problem, we utilize the Kuhn-Tucker condition of problem (2), which is the necessary condition for the current position to be in IR, in order to obtain the rational reaction $\bar{x}_2(\bar{x}_1)$ corresponding to $\bar{x}_1$ for the purpose of more accurately check whether an agent with the current position ($\bar{x}_1$, $\bar{x}_2$) exists around IR or not.

$$\left.\begin{array}{ll}\text{minimize} & f_2(\bar{x}_1, x_2)\\ \text{subject to} & g_i(\bar{x}_1, x_2) \le 0,\ i=1, 2, \dots, m\end{array}\right\} \quad (2)$$

The Kuhn-Tucker condition of (2) is expressed as follows.

$$\nabla f_2\left(\bar{x}_1, x_2^*\right) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i\left(\bar{x}_1, x_2^*\right) = 0, \lambda^* \ge \mathbf{0},$$

$$g_i\left(\bar{x}_1, x_2^*\right) \le 0,\ \lambda_i^* g_i\left(\bar{x}_1, x_2^*\right) = 0, i = 1,\ 2,\ \dots, m$$

It should be noted here that the best position $(\tilde{x}_1, \tilde{x}_2)$ obtained through the search by EMAS does not always exist in IR even if it satisfies the Kuhn-Tucker condition. In order to find a rational reaction for the upper level decision $\tilde{x}_1$ of the best solution, the following problem with respect to the lower level objective function is solved.

$$\left.\begin{array}{ll}\text{minimize} & f(\tilde{x}_1, x_2)\\ \text{subject to} & g_i(\tilde{x}_1, x_2) \le 0,\ i=1, 2, \dots, m\end{array}\right\} \quad (3)$$

Since (3) is a single-objective nonlinear programming problem, we can solve the problem by using an approximate solution method based on PSO (Matsui et al, 2007) to check whether the (approximate) optimal solution (x^*_2) is equal to $\tilde{x}_2$ or not. If (x^*_2) = $\tilde{x}_2$, we can regard $(\tilde{x}_1, \tilde{x}_2)$ as the (approximate) Stackelberg solution since it exists in IR. Otherwise, repeatedly check whether the current position of an agent satisfying Kuhn-Tucker condition exists in IR or not in the same manner mentioned above in order of the quality of the upper level objective function value. If an agent whose current position exists in IR, the position is regarded as the (approximate) Stackelberg solution.

5. Detailed procedure of the proposed EMAS

In this section, we describe the details of the computational procedures in the proposed EMAS for obtaining (approximate) Stackelberg solutions to noncooperative two-level programming problems.

In the proposed EMAS-based method, we use N agents $\boldsymbol{a}_r$, r=1, 2 …, N, each of which is characterized by some attributes, i.e., the current position ($\boldsymbol{x}_{r1}$, $\boldsymbol{x}_{r2}$), the upper level objective function value $f_1(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$, the lower level objective function value $f_2(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$, the energy e_r and the agent state variable k_r. If Kuhn-Tucker condition is satisfied for the rth agent $\boldsymbol{a}_r$ whose current position is in the feasible region, let k_r=1. If not, let k_r=0. For agents in the infeasible region, let k_r<0.

The procedure of the proposed EMAS-based method is summarized as follows.

Step 1. Generate N agents $\boldsymbol{a}_r$, r=1, 2, …, N by using the homomorphous mapping (Koziel & Michalewicz, 1999).

Step 2. Let T := 1.

Step 3. Let r := 1.

Step 4. If k_r=0, go to step 5. If k_r=1, go to step 6. If k_r<0, go to step 8.

Step 5. Carry out the search to the direction of improving the lower level objective function value. To be more specific, for the rth agent $\boldsymbol{a}_r$, choose an agent $\boldsymbol{a}_{r'}$ randomly. Then, compare $f_2(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$ with $f_2(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r'2})$. If $f_2(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2}) < f_2(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r'2})$, let $\boldsymbol{x}_{p2} := \boldsymbol{x}_{r2}$ and $\boldsymbol{x}_{p'2} := \boldsymbol{x}_{r'2}$. Otherwise, let $\boldsymbol{x}_{p2} := \boldsymbol{x}_{r'2}$ and $\boldsymbol{x}_{p'2} := \boldsymbol{x}_{r2}$.

Then, update $\boldsymbol{x}_{p'2}$ by the following scheme:

$$\boldsymbol{x}_{p'2} := \boldsymbol{x}_{p'2} + 2R\left(\boldsymbol{x}_{p2} - \boldsymbol{x}_{p'2}\right) \tag{4}$$

where R is a uniform random number in [0,1]. Repeat the comparison between $f_2(\boldsymbol{x}_{p1}, \boldsymbol{x}_{p2})$ and $f_2(\boldsymbol{x}_{p'1}, \boldsymbol{x}_{p'2})$ and the update of $\boldsymbol{x}_{p'2}$ n times. Let the final $\boldsymbol{x}_{p2}$ be $\boldsymbol{x}_{r2}$. If the current position of $\boldsymbol{a}_r$ satisfies Kuhn-Tucker condition, let k_r:=1 and go to step 6. Otherwise, go to step 9.

Step 6. If k_r=1, carry out the search to the direction of improving the upper level objective function value. To be more specific, choose an agent $\boldsymbol{a}_{r''}$ which satisfies Kuhn-Tucker condition at random as a comparing opponent and compare the upper level objective function value of $\boldsymbol{a}_r$ with that of $\boldsymbol{a}_{r''}$. Then, the superior agent gains the energy from the inferior one. Let the superior agent denote $\boldsymbol{a}_{r''}$ and the inferior one denote $\boldsymbol{a}_r$. If the inferior agent is killed by the disappearance of energy, go to step 7. Otherwise, go to step 9.

Step 7. Carry out the reproduction of the killed agent. Then, reproduce $\boldsymbol{a}_r$ with the current position which is determined as:

$$\left(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2}\right) := \left(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2}\right) + 2R\left(\left(\boldsymbol{x}_{r''1}, \boldsymbol{x}_{r''2}\right) - \left(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2}\right)\right). \tag{5}$$

where R is a uniform random number in [0,1]. If the current position of the reproduced $\boldsymbol{a}_r$ is infeasible, let k_r:=-1 and go to step 8. Otherwise, let k_r:=0 and go to step 9.

Step 8. Carry out the search in the infeasible region. If $-t_1 \le k_r < 0$, go to substep 8-1. If $-t_2 \le k_r < -t_1$, go to substep 8-2. If $kr = -t_2$, go to substep 8-3. Here, let $0<t_1<t_2$.

Substep 8-1: Let an agent with the current position ($\boldsymbol{x}_{r,\,1}$, $\boldsymbol{x}_{r,\,2}$, …, $\boldsymbol{x}_{r^*,\,i}$, …, $\mathbf{x}_{r,\,n2}$) denote $\boldsymbol{a}_{r'}$. Here, r^* is randomly chosen from among {1, 2, …, N} and i is randomly chosen from among {1, 2, …, n_1+n_2}. Compare $f_1(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$ with $f_2(\boldsymbol{x}_{r'1}, \boldsymbol{x}_{r'2})$. Let the superior agent and the inferior one denote $\boldsymbol{a}_s$ and $\boldsymbol{a}_{s'}$, respectively. Then, update $(\boldsymbol{x}_{s'1}, \boldsymbol{x}_{s'2})$ by the following scheme:

$$(\boldsymbol{x}_{s'1},\boldsymbol{x}_{s'2}) = (\boldsymbol{x}_{s'1},\boldsymbol{x}_{s'2}) + 2R\big((\boldsymbol{x}_{s1},\boldsymbol{x}_{s2}) - (\boldsymbol{x}_{s'1},\boldsymbol{x}_{s'2})\big) \quad (6)$$

where R is a uniform random number in [0,1]. Repeat the comparison between $f_1(\boldsymbol{x}_{s1}, \boldsymbol{x}_{s2})$ and $f_1(\boldsymbol{x}_{s'1}, \boldsymbol{x}_{s'2})$ and the update of $(\boldsymbol{x}_{s'1}, \boldsymbol{x}_{s'2})$ n' times. Let the final $(\boldsymbol{x}_{s1}, \boldsymbol{x}_{s2})$ be $(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$. If $(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$ is feasible, let k_r:=0. Otherwise, let k_r:=k_r-1. Go to step 9.

Substep 8-2: Choose an agent $\boldsymbol{a}_{r'}$ randomly. Compare $(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$ with $(\boldsymbol{x}_{r'1}, \boldsymbol{x}_{r'2})$ using the following function

$$h(\boldsymbol{x}_1,\boldsymbol{x}_2) = \sum_{i=1}^{m} g_i(\boldsymbol{x}_1,\boldsymbol{x}_2), g_i(\boldsymbol{x}_1,\boldsymbol{x}_2) > 0$$

which is the degree of violation of the constraints. Let the superior position denote $(\boldsymbol{x}_{u1}, \boldsymbol{x}_{u2})$ and the inferior one denote $(\boldsymbol{x}_{u'1}, \boldsymbol{x}_{u'2})$. Then, update $(\boldsymbol{x}_{u'1}, \boldsymbol{x}_{u'2})$ by the following scheme:

$$(\boldsymbol{x}_{u'1},\boldsymbol{x}_{u'2}) = (\boldsymbol{x}_{u'1},\boldsymbol{x}_{u'2}) + 2R\big((\boldsymbol{x}_{u1},\boldsymbol{x}_{u2}) - (\boldsymbol{x}_{u'1},\boldsymbol{x}_{u'2})\big)$$

where R is a uniform random number in [0,1]. Repeat the comparison between $h(\boldsymbol{x}_{u1}, \boldsymbol{x}_{u2})$ and $h(\boldsymbol{x}_{u'1}, \boldsymbol{x}_{u'2})$ and the update of $(\boldsymbol{x}_{u'1}, \boldsymbol{x}_{u'2})$ n'' times. Let the final $(\boldsymbol{x}_{u1}, \boldsymbol{x}_{u2})$ be $(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$. If $(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$ is feasible, let k_r:=0. Otherwise, let k_r:=k_r-1. Go to step 9.

Substep 8-3: Choose an agent $\boldsymbol{a}_{r'}$ whose current position is feasible randomly, and move $\boldsymbol{a}_r$ to the feasible region by the bisection method between $\boldsymbol{a}_{r'}$ and $\boldsymbol{a}_r$. Go to step 9.

Step 9. If $r = N$, go to step 10. Otherwise, let $r := r+1$ and return to step 4.

Step 10. If $T = T_{\max}$, go to step 11. Otherwise, let $T := T+1$ and return to step 3.

Step 11. Check whether the current position of the best agent exists in IR or not by solving (3) through the revised PSO method (Matsui et al., 2007), which is one of most promising solution methods for nonlinear programming problems. If the current position of the best agent exists in IR, we can regard it as the (approximate) Stackelberg solution. Otherwise, repeatedly check whether the current position of an agent satisfying Kuhn-Tucker condition exists in IR or not in the same manner mentioned above in order of the quality of the upper level objective function value. If an agent whose current position satisfies Kuhn-Tucker condition and exists in IR, the position is regarded as the (approximate) Stackelberg solution, and the solution procedure is terminated.

6. Numerical examples

In order to investigate the efficiency of the proposed method, we conduct some numerical experiments.

First, we consider a two-level nonlinear programming problem with 4 decision variables and 8 constraints (P1) and one with 6 decision variables and 10 constraints (P2), and compare the computational time of generating the initial population including 1000 agents by the homomorphous mapping (Koziel & Michalewicz, 1999) with that by the random method. The results are shown as in table 1.

	Computational time (sec.)	
	P1	P2
Homomorphous mapping	1.500	3.109
Random method	177.650	2691.710

Table 1. Comparison of computational times of generating the initial population including 1000 agents

The results in table 1 show the effectiveness of the use of the homomorphous mapping in generating the initial population.

Second, in order to investigate the efficiency of substeps 8-2 and 8-3, we compare the result of EMAS without 8-2, 8-3 with that of EMAS with 8-2, 8-3 by setting both the number of agents and the maximal generation number to 1000s. Results are shown as in table 2.

	Upper level objective function	Lower level Objective function
EMAS without 8-2 and 8-3	-14.999991	0.999997
EMAS with 8-2 and 8-3	-24.0	0.0
Optimal value	-24.0	0.0

Table 2. The efficiency of substeps 8-2 and 8-3

The results in table 2 show that both substeps 8-2 and 8-3 are worth being introduced in the proposed method.

Next, in order to investigate the efficiency of substep 8-1 for improving the upper level objective function value, we apply EMAS without 8-1 and one with 8-1.

	Upper level objective function	
	Problem A (8 decision variables)	Problem B (20 decision variables)
EMAS without 8-1	453.620022	-0.202166
EMAS with 8-1	452.087664	-0.286156

Table 3. The efficiency of substeps 8-1

To be more specific, problem A and B are formulated as:

Problem A: Upper level decision variables $\boldsymbol{x}_1 = (x_1, x_2, x_3, x_4)$, lower level decision variables: $\boldsymbol{x}_2 = (x_5, x_6, x_7, x_8)$.

$$\begin{aligned}
\underset{x_1}{\text{minimize}} \quad & f_1(\boldsymbol{x}_1, \boldsymbol{x}_2) = x_1^3+(x_2-2)^2-x_3x_4-2x_5^2+x_6^4+(x_7-x_8)^2 \\
& \text{where } \boldsymbol{x}_2 \text{ solves} \\
\underset{x_2}{\text{minimize}} \quad & f_2(\boldsymbol{x}_1, \boldsymbol{x}_2) = -2x_1^2+x_2^3+2x_3^2-x_4^4+3x_5^3-2x_6+5x_7+4x_8^2 \\
\text{subject to} \quad & 4x_1^2-3x_2^3+5(x_3-4)^2-6x_4-x_5x_6-3x_7^4+5x_8 \leq 0 \\
& 2x_1x_7-x_2^2-3x_3-4x_4^2+x_5^2-x_6^3-2x_8^2 \leq -12 \\
& 5x_1+x_2-6x_3+4x_4-6x_5-x_6-3x_7+x_8 \leq 5 \\
& -5 \leq x_j \leq 5,\ j = 1, \ldots, 8,
\end{aligned}$$

Problem B: Upper level decision variables $\boldsymbol{x}_1 = (x_1, \ldots, x_{10})$, lower level decision variables $\boldsymbol{x}_2 = (x_{11}, \ldots, x_{20})$.

$$\underset{x_1}{\text{minimize}} \quad f_1(\boldsymbol{x}_1, \boldsymbol{x}_2) = -\left| \frac{\sum_{j=1}^{20} cos^4(x_j) - 2\prod_{j=1}^{20} cos^2(x_j)}{\sqrt{\sum_{j=1}^{20} j x_j^2}} \right|$$

where $\boldsymbol{x}_2$ solves

$$\underset{x_2}{\text{minimize}} \quad f_2(\boldsymbol{x}_1, \boldsymbol{x}_2) = (x_1-x_{14})(x_{17}-x_6)-(x_4-x_{11})(x_{18}-x_7)+(x_8-x_{12})(x_5-x_{19}) -(x_{13}-x_3)(x_{10}-x_{16})+(x_{20}-x_9)(x_2-x_{15})$$

$$\text{subject to} \quad 0.75-\prod_{j=1}^{20} x_j \le 0$$

$$\prod_{j=1}^{20} x_j -7.5\cdot 20 \le 0$$

$$0 \le x_j \le 10, j = 1, \ldots, 20.$$

From the results in table 3, the procedure of substep 8-1 is meaningful for enhancing the efficiency of the proposed method.

Furthermore, in order to investigate the efficiency of the proposed EMAS, we compare the result obtained by it with that by two-level PSO method (Niwa et al., 2006) in the application of both methods to a two-level nonlinear programming problem with 10 decision variables and 3 constraints. In the numerical experiment, the number of agents is 1000, the maximal generation number is 1000 and the number of trials is 10. Table 4 shows the best value, the average value, the worst value of the upper level objective function obtained by the proposed EMAS in 10 trials, the best value obtained by the two-level PSO (Niwa et al., 2006) and the average computational time.

	Upper level objective function value	Computational time (sec.)
Best	-199.515027	94.3028
Average	-196.160929	
Worst	-192.904405	
PSO	-184.600761	1357.313

Table 4. Comparison of the proposed EMAS with two-level PSO method

Table 4 shows that the proposed EMAS is superior to PSO because the best value obtained by PSO is worse than the worst value of the proposed EMAS.

Finally, table 5 shows the effect of the number of agents and the number of generations on computational time. Table 5 shows the computational time of EMAS linearly increases as the number of agents and the number of generations.

	The number of agents		
	1000	2000	3000
Generation 1000	165.375	330.500	489.343
Generation 2000	344.203	689.984	1034.234
Generation 3000	525.718	1054.718	1587.437

Table 5. Effect of the number of agents and the number of generations on computational time (sec)

7. Conclusion

In this chapter, we discussed an efficient approximate solution method based on evolutionary multi-agent systems to obtain Stackelberg solutions to noncooperative two-level programming problems. In particular, we proposed a new EMAS by incorporating the concept of homomorphous mapping to generate feasible initial agents, the theory of the Kuhn-Tucker condition for checking whether an agent exists around IR or not, and the idea of reproduction in the infeasible region, together with the introduction of the searching process of the infeasible region by infeasible agents in order to widen the search area. Furthermore, we showed the efficiency of the proposed EMAS by comparing it with an existing method, the two-level PSO method, through some numerical experiments. From the numerical experimental results, it is indicated that the proposed EMAS is superior to the two-level PSO method, and that the proposed EMAS is promising as an optimization method for two-level nonlinear programming problems. In the near future, we will extend the proposed method to noncooperative and cooperative multi-level programming.

8. References

M. A. Amouzegar & K. Moshirvaziri. (1999). Determining optimal pollution control policies: an application of bilevel programming, *European Journal of Operational Research*, Vol. 119, No. 1, pp. 100--120

W. F. Bialas & M. H. Karwan. (1982). On two-level optimization, *IEEE Transactions on Automatic Control*, Vol. AC-27, No. 1, pp. 211–214

W. F. Bialas & M. H. Karwan. (1984). Two-level linear programming, *Management Science*, Vol. 30, No. 8, pp. 1004—1020

J. F. Bard & J. E. Falk. (1982). An explicit solution to the multi-level programming problem, *Computer and Operations Research*, Vol. 9, No. 1, pp. 77—100

J. F. Bard & J. T. Moore. (1990). A branch and bound algorithm for the bilevel programming, *SIAM Journal on Scientific and Statistical Computing*, Vol. 11, No. 2, pp. 281--292

B. Colson, P. Marcotte, G. Savard. (2005). A trust-region method for nonlinear bilevel programming: algorithm and computational experience, *Computational Optimization and Applications*, Vol. 30, No. 3, pp. 211—227

S. Dempe & J. F. Bard. (2001). Bundle trust-region algorithm for bilinear bilevel programming, *Journal of Optimization Theory and Applications*, Vol. 110, No. 2, pp. 265--288

N. P. Faisca, V. Dua, B. Rustem, P. M. Saraiva, E. N. Pistikopoulos. (2007) Parametric global optimization for bilevel programming, *Journal of Global Optimization*, Vol. 38, No. 4, pp. 609—623

M. Fampa, L. A. Barroso, D. Candal, L. Simonetti. (2008). Bilevel optimization applied to strategic pricing in competitive electricity markets, *Computational Optimization and Applications*, Vol. 39, No. 2, pp. 121--142

Z. H. GUMUS & C. A. Floudas. (2001). Global optimization of nonlinear bilevel programming problems, *Journal of Global Optimization*, Vol. 20, No. 1, pp. 1--31

P. Hansen, B. Jaumard, G. Savard. (1992). New branch-and-bound rules for linear bilevel programming, *SIAM Journal on Scientific and Statistical Computing*, Vol. 13, No. 5, pp. 1194--1217

S. Koziel, Z. Michalewicz. (1999). Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization, *Evolutionary Computation*, Vo. 7, No. 1, pp. 19--44

T. Matsui, K. Kato, M. Sakawa, T. Uno, K. Morihara. (2007). Particle swarm optimization based heuristics for nonlinear programming problems, *Proceedings of International MultiConference of Engineers and Computer Scientists 2007*, pp. 2312—2317

M. G. Nicholls. (1996). The applications of non-linear bi-level programming to be aluminium industry, *Journal of Global Optimization*, Vol. 8, No. 3, pp. 245--261

I. Nishizaki & M. Sakawa. (1999). Stackelberg solutions to multiobjective two-level linear programming problems, *Journal of Optimization Theory and Applications*, Vol. 103, No. 1, pp. 161—182

I. Nishizaki & M. Sakawa. (2000). Computational methods through genetic algorithms for obtaining Stackelberg solutions to two-level mixed zero-one programming problems, *Cybernetics and Systems: An International Journal*, Vol. 31, No. 2, pp. 203—221

I. Nishizaki, M. Sakawa, H. Katagiri. (2003). Stackelberg solutions to multiobjective two-level linear programming problems with random variable coefficients, *Central European Journal of Operations Research*, Vol. 11, No. 3, pp. 281--296

K. Niwa, Ichiro Nishizaki, M. Sakawa. (1999). Computational methods for obtaining Stackelberg Solutions to two-level non-linear programming problems, *In: Proceedings of Second Asia-Pacific Conference on Industrial Engineering and Management Systems*, pp. 489—492

K. Niwa, K. Kato, I. Nishizaki, M. Sakawa. (2006). Computational methods through particle swarm optimization for obtaining Stackelberg solutions to two-level nonlinear programming problems, *In: Proceedings of 22nd Fuzzy System Symposium*, pp. 229--230 (in Japanese)

E. Roghanian, S. J. Sadjadi, M. B. Aryanezhad. (2007). A probabilistic bi-level linear multiobjective programming problem to supply chain planning, *Applied Mathematics and Computation*, Vol. 188, No. 1, pp. 786--800

M. Sakawa & I. Nishizaki. (2001). Interactive fuzzy programming for multi-level nonconvex nonlinear programming problems, *In: Dynamical Aspects in Fuzzy Decision Making (Ed. Y. Yoshida)*, Physica-Verlag, Heidelberg

M. Sakawa & I. Nishizaki. (2009). *Cooperative and Noncooperative Multi-Level Programming*, Springer, New York

K. Shimizu, Y. Ishizuka, J. F. Bard. (1997). *Nondifferentiable and Two-Level Mathematical Programming*, Kluwer Academic Publishers, Boston

M. Simaan & J. B. Cruz Jr. (1973). On the Stackelberg strategy in nonzero-sum games, Journal of Optimization Theory and Applications, Vol. 11, No. 5, pp. 533--555

K. Socha & M. Kisiel-Dorohinicki. (2002). Agent-based evolutionary multiobjective optimization, *Proceedings of Congress on Evolutionary Computation*, pp. 109--114

D. J. White & G. Anandalingam. (1993). A penalty function approach for solving bi-level linear programs, *Journal of Global Optimization*, Vol. 3, No. 4, pp. 397--419

Part 2

Control, Negotiation, Reasoning, Tracking and Networking on Agents Environments

6

Convergence and Collision Avoidance in Formation Control: A Survey of the Artificial Potential Functions Approach

Eduardo G. Hernández-Martínez[1] and Eduardo Aranda-Bricaire[2]
[1]*Tecnológico de Estudios Superiores de Coacalco*
[2]*CINVESTAV*
Mexico

1. Introduction

Multi-agent Robots Systems (MARS) can be defined as sets of autonomous robots coordinated through a communication system to achieve cooperative tasks. During the last 20 years, MARS have found a wide range of applications in terrestrial, spatial and oceanic explorations emerging as a new research area (Cao et al. (1997)). Some advantages can be obtained from the collective behavior of MARS. For instance, the kind of tasks that can be accomplished are inherently more complex than those a single robot can accomplish. Also, the system becomes more flexible and fault-tolerant (Yamaguchi (2003)). The range of applications includes toxic residues cleaning, transportation and manipulation of large objects, alertness and exploration, searching and rescue tasks and simulation of biological entities behaviors (Arai et al. (2002)). The study of MARS extends the classical problems of single robots with new issues like motion coordination, task decomposition and task assignment, network communications, searching and mapping, etc. Therefore, the study of MARS encompass distributed systems, artificial intelligence, game theory, biology, ethology, economics, control theory, etc.
Motion coordination is an important research area of MARS, specifically formation control (Chen & Wang (2005)). The main goal is to coordinate a group of mobile agents or robots to achieve a desired formation pattern avoiding inter-agent collisions at the same time. The formation strategies are decentralized because it is assumed that every agent measures the position of a certain subset of agents and, eventually, it detects the position of other agents when a minimal allowed distance is violated and collision danger appears. Thus, the main intention is to achieve desired global behaviors through local interactions (Francis et al. (2004)). Also, the decentralized approaches offer greater autonomy for the robots, less computational load in control implementations and its applicability to large scale groups (Do (2007)). According to Desai (2002); Muhammad & Egerstedt (2004), the possible inter-agent communications and the desired relative position of every agent with respect to the others can be represented by a Formation Graph (FG). The application of different FG's to the same group of robots produces different dynamics on the team behavior. In the literature, some special FG topologies are chosen and the convergence to the desired formation and non-collision is analyzed for any number of robots.
A decentralized formation strategy must comply with two fundamental requirements: Global convergence to the desired formation and inter-agent collision avoidance (Cao et al. (1997)).

The standard methodology of Artificial Potential Functions consists in applying the negative gradient of a mixture of Attractive (APF) and Repulsive Potential Functions (RPF) as control inputs to satisfy the convergence and non-collision properties, respectively (Leonard & Fiorelli (2001); Tanner & Kumar (2005)). The APF's are designed according the desired inter-robot distances and steer all agents to the desired formation. The RPF's are based on functions of the distance of a pair of agents. In a decentralized non-collision strategy, a local RPF tends to infinity when two agents collide and vanishes smoothly until the minimal allowed distance is reached. A control law based on APF's only guarantees the global convergence to the formation pattern, however inter-robot collision can occur. The addition of RPF's guarantees the non-collision. However, the main drawback of mixing APF and RPF is the appearance of equilibria where the composite vector field vanishes and the robots can get trapped at undesired equilibrium points. Therefore, a proof of global convergence to the desired formation for all initial conditions becomes quite involved because the analysis to calculate these equilibria and the trajectories which do not converge to the desired formation is very complex (Do (2006)).

This chapter analyzes the global convergence and non-collisions strategies of formation strategies based on Artificial Potential Functions on the context of formation graphs for the case of point robots or omnidirectional robots. In Section 2, we present a literature review about different formation strategies with emphasis on the approaches that modify the original Artificial Potential Functions method to ensure convergence and collision avoidance at the same time. Section 3 establishes a formal problem statement and the basic concepts of FG's. The standard methodology of APF's and RPF's and the complexity of the computation of equilibria is presented in Section 4. Some contributions to the literature of formation control based on Artificial Potential Functions are presented in Sections 5 and 6 with the analysis of the centroid of positions and the design of a new repulsive vector field that improve the performance of the robots' trajectories. Finally, in Section 7, the control laws are extended to for the case of unicycle-type robots with numerical simulations and Section 8 presents some experiments using a setup composed by two o three unicycle-like robots and a computer vision system to estimate positions and orientations of the robots within the workspace.

2. State of the art

Formation control is presented in most of MARS applications because generally is required a coordination control to obtain a strategic displacement or posture of the robots within the workspace to achieve a common work (Chen & Wang (2005)). For instance, on surveillance and exploration tasks, is required that robots move forward on a specific formation pattern to maximize their detection capacities and eventually, reconfigure this pattern if some robot breaks down (Balch & Arkin (1998)). In the case of manipulation of large objects (Arai et al. (2002)), the robots must achieve strategic team positions to carry an object within the workspace (Asahiro et al. (1999); Cao et al. (1997)).

Chen & Wang (2005) suggest to consider the formation control as a regulation problem, a well known concept in control theory. As mentioned above, the goal is the design of decentralized schemes based on the following assumptions: a) every agent knows its desired position on the group but not the goals of the others and b) every robot knows only the position of a certain subset of robots to converge to the desired formation (Dimarogonas, Kyriakopoulos & Theodorakatos (2006); Feddema et al. (2002)). However, there is no general consensus to delimit the decentralized schemes. The most accepted idea is to identify the degree of decentralization of the formation control. For instance, the case of zero decentralization (or full centralization) consists on team robots where every agent knows the positions and goals of the others. The next level arises when the agents know the positions of the others but not their goals and the maximum degree of decentralization is the case where the agents share the minimum information to converge to the desired formation.

Formation control schemes can be classified into two categories. First, the behavior-based schemes come from the study of animal behaviors where the agents are formed following simple behavior rules, as maintaining a distance between neighbors, swarm intelligence and self-organization, aggregation, flocks, hunter-prey system, etc. (Balch & Arkin (1998); Reynolds (1987); Spears et al. (2004); Yamaguchi (2003)). This scheme considers to all agents with the same sensing capacities and generally converge to formation patterns without a specific position for every agent. The second scheme is related to model-based behaviors or emergent behaviors on the context of FG's. Some tools of graph theory and linear systems are used to analyze the closed-loop system (Desai (2002); Desai et al. (1998); Fax & Murray (2004); Olfati-Saber & Murray (2002); Tanner (2004)). Similar to FG's, the focus on geometric patterns is found in works as Marshall et al. (2004). Other model-based behaviors are mentioned in Chen & Wang (2005), as nonlinear servomechanisms, genetic algorithms, Distributed Artificial Intelligence, attractive forces of particles, etc.

Some works in the literature are related to the analysis of convergence without considering the inter-robot collisions. For instance, the analysis of *agreement problem* or *consensus problem* (Dimarogonas & Kyriakopoulos (2006b); Francis et al. (2004); Olfati-Saber & Murray (2003)) establishes the minimum conditions for the convergence of the robots to a common point considering sensing capacities limited to a certain influence area. Other works are based on graph theory and linear systems to prove the convergence for some typical cases of FG's, described bellow (see Baillieul & McCoy (2007); Fax & Murray (2004); Muhammad & Egerstedt (2004)). Finally, works as (Dimarogonas & Kyriakopoulos (2006a); Hendrickx et al. (2007); Lafferriere et al. (2004); Li & Chen (2005b); Swaroop & Hedrick (1996); Tanner et al. (2002; 2004)) complement the convergence analysis with the study of formation infeasibility, formation rigidity and formation stability. For instance, the formation infeasibility studies the conditions of the desired vectors of position in a FG that eliminate the equilibrium points and consequently the possibility of the convergence to the desired formation. The *chain stability* and rigidity analyze the disturbance propagation in a group of robots when has achieved the desired formation. The *leader-formation stability* studies how the leader behavior affects the formation of all robots.

On the other hand, the non-collision strategies include reactive schemes based on simple behavior-based rules (Ando et al. (1999); Balch & Arkin (1998); Egerstedt & Hu (2001); Reynolds (1987)), hybrid architectures (Cao et al. (2003); Das & Fierro (2003); Mallapragada et al. (2006)), physics-based and swarms techniques (Spears et al. (2004)) or repulsive forces based on Artificial Potential Functions or repulsive vector fields (Ogren & Leonard (2003); Rimon & Koditschek (1992); Schneider & Wildermuth (2005)). As mentioned above, decentralized RPF's appear only within the influence zone of every robot, equivalently, every agent does not know the position of other agents unless there is danger of collision. Due to the possible scenarios of collision for the general case, Do (2006) establishes that a proof of convergence to the desired formation for all initial conditions becomes quite involved by the appearance of undesired equilibria. In Dimarogonas & Kyriakopoulos (2006a), it is shown the complexity analysis of decentralized RPF's applied to all FG's with bidirectional communication. However, the convergence analysis discards the undesired equilibria.

In the literature, there exist different approaches to modify the original Artificial Potential Functions method to ensure convergence and collision avoidance. For example, in (Dimarogonas & Kyriakopoulos (2005); Dimarogonas, Loizou, Kyriakopoulos & Zavlanos (2006); Gennaro & Jadbabaie (2006); Olfati-Saber & Murray (2002); Tanner & Kumar (2005)), composed functions or navigation functions are designed with attractive and repulsive behavior to eliminate the undesired equilibria. The drawback is that the non-collision strategy becomes centralized because it requires full-knowledge of the system. Also, most of these functions are high-order with a corresponding high computational cost for the implementation on real robots. Do (2006) demonstrates that the undesired equilibrium points are unstable (saddle point) for the case of the complete FG. Exploiting the unstable behavior of these equilibria, some approaches propose small disturbances in order to agents escape of these equilibria using online strategies such that virtual obstacle method (Lee & Park (2003; 2004); Li & Chen (2005a); Ogren & N.E. Leonard (2004)), instantaneous goal approach (Ge & Fua (2005)), etc. However, the previous strategies are patch-type and they do not include formal proofs about the convergence to the desired formation. On the other hand, the use of non-smooth vector fields can rule out the existence of undesired equilibria. Some works about discontinuous vector fields in formation control are (Hernandez-Martinez & Aranda-Bricaire (2009b); Loizou & Kyriakopoulos (2002); Loizou, Tannert, Kumar & Kyriakopoulos (2003); Loizou, Tannert & Kyriakopoulos (2003); Tanner (2004); Yao et al. (2006)). The analysis falls on the control of variable structure systems (Itkis (1976)). In most works, the repulsive discontinuous forces are designed heuristically and no formal proofs are presented (for instance, Barnes et al. (2007); Kim et al. (2005)).

Finally, other strategies of non-collision are listed in Chen & Wang (2005) like the predictive model control, social potential fields, fuzzy logic and neural networks. In these schemes, a hierarchical control scheme is proposed, where the higher level coordinates reactive collision avoidance actions. A few works deal about the non-adequacy on communication and the delay effects on the formation stability.

3. Problem statement and formation graphs

Inspired in (Chen & Wang (2005); Francis et al. (2004); Tanner & Kumar (2005)), a general definition of formation control for point robots or omnidirectional robots is established as follows:

Denote by $N = \{R_1, ..., R_n\}$, a set of n agents moving in plane with positions $z_i(t) = [x_i(t), y_i(t)]^T$, $i = 1, ..., n$. The kinematic model of each agent or robot R_i is described by

$$\dot{z}_i = u_i, \quad i = 1, ..., n, \tag{1}$$

where $u_i = [u_{i1}, u_{i2}]^T \in \Re^2$ is the velocity of i-th robot along the X and Y axis . Let $N_i \subseteq \{z_1, ..., z_n\}$, $N_i \neq \emptyset$, $i = 1, ..., n$. denote the subset of positions of the agents which are detectable for R_i. Let $c_{ji} = [h_{ji}, v_{ji}]^T$, $\forall j \in N_i$ denote a vector which represents the desired position of R_i with respect to R_j in a particular formation. Thus, we define the desired relative position of every R_i in the formation by

$$z_i^* = \varphi_i(N_i) = \frac{1}{n_i} \sum_{j \in N_i} \left(z_j + c_{ji} \right), i = 1, ..., n \tag{2}$$

where n_i is the cardinality of N_i. Thus, the desired relative position of R_i can be considered as a combination of the desired positions of z_i with respect to the positions of all elements of N_i. Let $d/2$ be the radius of the closed ball that every agent occupies within the workspace.

Problem Statement. The control goal is to design a control law $u_i(t) = f_i(N_i(t))$ for every robot R_i, such that

- $\lim_{t\to\infty}(z_i - z_i^*) = 0, i = 1, ..., n$. (convergence to the desired formation) and
- $\|z_i(t) - z_j(t)\| \neq 0, \forall t \geq 0, i \neq j$ (collision avoidance).

According to (Desai (2002); Muhammad & Egerstedt (2004)), the desired relative positions of a group of agents on a desired formation can be represented by a FG defined by

Definition 1. *A Formation Graph $G = \{Q, E, C\}$ is a triplet that consists in (i) a set of vertices $Q = \{R_1, R_2, ..., R_n\}$ related to the team members, (ii) a set of edges $E = \{(j,i) \in Q \times Q\}$, $i \neq j$ containing pairs of nodes that represent inter-agent communications, therefore $(j,i) \in E$ iff $j \in N_i$ and (iii) a set of vectors $C = \{c_{ji}\}$, $\forall (j,i) \in E$ that specify the desired relative position between agents i and j, i.e. $z_i - z_j = c_{ji} \in \Re^2$, $\forall i \neq j, j \in N_i$ in a desired formation pattern.*

If $(i,j) \in E$, then the vertices i and j are called *adjacent*. The degree g_i of the $i-th$ vertex is defined as the number of its adjacent vertices. A path from vertex i to j is a sequence of distinct vertices starting with i and ending with j such that consecutive vertices are adjacent. The underlying graph of a FG, is the graph where $\forall(i,j) \in E$, is added a new edge (j,i), if it does not appear on the original FG. The underlying graph is always an undirected graph. If there is a path between any two vertices of the underlying graph of FG, then the FG is said to be *connected*. Thus, a FG is said to be well defined if it satisfied the following conditions: (1) the graph is connected, (2) there are no conflicts in the desired vectors of positions, in the sense that if $c_{ij}, c_{ji} \in C$, then $c_{ij} = -c_{ji}$ and (3) the desired vectors of positions establish a closed-formation, i.e., if there exist the vectors $c_{jm_1}, c_{m_1m_2}, c_{m_2m_3}, ..., c_{m_rj}$, then they must satisfy:

$$c_{jm_1} + c_{m_1m_2} + c_{m_2m_3} + ... + c_{m_rj} = 0. \quad (3)$$

The previous condition establishes that some position vectors form closed-polygons. The *Laplacian matrix* of a FG captures many fundamental topological properties of the graph and it is defined bellow.

Definition 2. *The* Laplacian matrix *of a FG G is the matrix*

$$\mathcal{L}(G) = \Delta - A_d \quad (4)$$

where $\Delta = diag[g_1, ..., g_n]$, where g_i is the degree of the vertex i, $A_d = \in \Re^{n\times n}$ is called the adjacency matrix *with elements*

$$a_{ij} = \begin{cases} 1, \text{ if } (j,i) \in E \text{ (or equivalently } c_{ji} \in C) \\ 0, \text{ otherwise} \end{cases} \quad (5)$$

For a connected FG, the Laplacian has a single zero eigenvalue and the corresponding eigenvector is $[1, ..., 1]^T \in \Re^n$. Fig. 1 shows an example of FG. The vertices are represented by circles and the arrows are the vectors c_{ji}. The circled elements of the Laplacian matrix are the degrees g_i. It is clear that $g_i = n_i, i = 1, ..., n$.
A FG is said to be *directed* if $\forall(j,i) \in E$, then $(i,j) \notin E$ (or $j \in N_i$ implies $i \notin N_j$), *undirected* if $\forall(j,i) \in E$ then $\forall(i,j) \in E$ (or $j \in N_i$ implies $i \in N_j$) and *mixed* otherwise. For instance, the FG of fig. 1 is mixed. For the case of undirected FG, the Laplacian is always a symmetric semidefinite positive matrix.
Fig. 2 shows some examples of FG topologies commonly found in the literature and their respective Laplacian matrices. For instance, Do (2006) analyzes the convergence of the

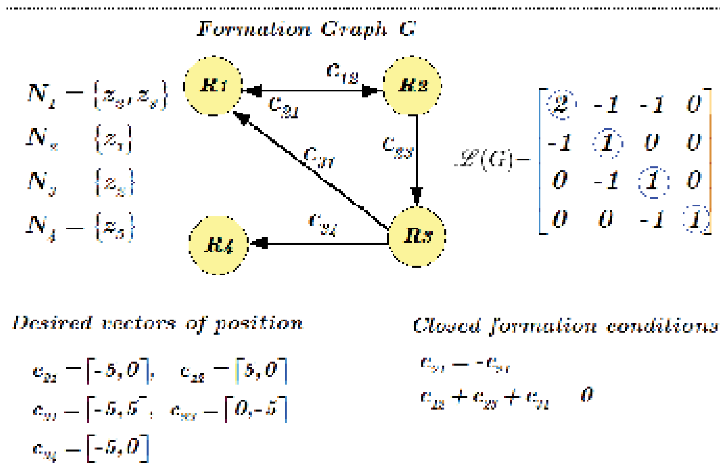


Fig. 1. Example of a Formation Graph

complete FG shown in Fig. 2a, where every robot senses the position of the others. The directed cyclic pursuit FG (fig. 2b) is studied in Francis et al. (2004), where the robot i pursues the robot $i+1$ and the n-th robot pursues the first making a closed-chain configuration. A variant of the cyclic pursuit with bidirectional communication (undirected FG) is shown in Fig. 2c and analyzed in (Hernandez-Martinez & Aranda-Bricaire (2008b)). An analysis of the convergence for all undirected FG's is presented in (Dimarogonas & Kyriakopoulos (2006a)). The FG of leader-followers is analyzed in (Hernandez-Martinez & Aranda-Bricaire (2008a)) for the case of the FG centered on a virtual leader (Fig. 2d) and (Hernandez-Martinez & Aranda-Bricaire (2009a)) for the open-chain or convoy configuration (Fig. 2e). Other approaches of leader-followers schemes are found in (Desai et al. (2001); Leonard & Fiorelli (2001); Tanner et al. (2004)) including virtual leaders, i.e., robots that does not physically exist but they are emulated in order to improve the performance of the system.
For completeness, the following definition is introduced

Definition 3. *The centroid of positions $\bar{z}(t)$ is the mean of the positions of all robots in the group, i.e.*

$$\bar{z}(t) = \frac{1}{n}(z_1(t) + ... + z_n(t)) \tag{6}$$

4. Control strategy based on APF's and RPF's

For system (1), APF's are defined by

$$\gamma_i = \sum_{j \in N_i} \|z_i - z_j - c_{ji}\|^2, \ \ \forall j \in N_i, \ \ i = 1, ..., n \tag{7}$$

The functions γ_i are always positives and reach their minimum ($\gamma_i = 0$) when $z_i - z_j = c_{ji}$, $i = 1, ..., n$, $j \in N_i$. Then, a control law based on APF's only is defined as

$$u_i = -\frac{1}{2} k \left(\frac{\partial \gamma_i}{\partial z_i} \right)^T, i = 1, ...n, \ \ k > 0. \tag{8}$$

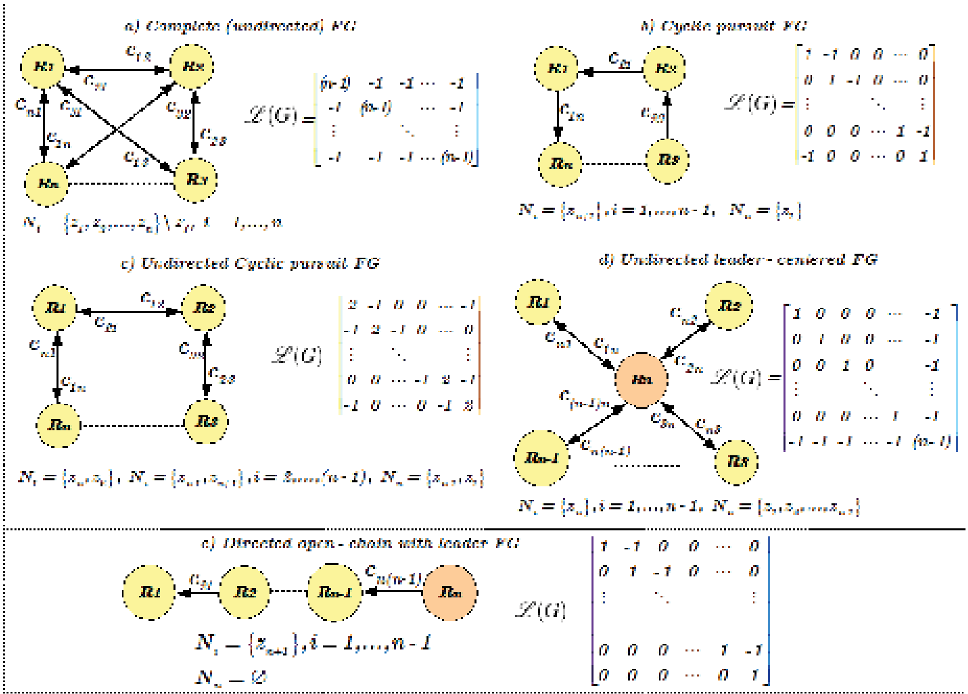


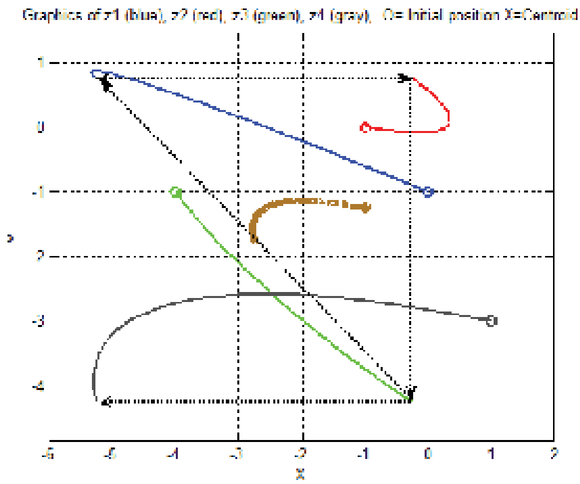
Fig. 2. Topologies of Formation Graphs

The closed-loop system (1)-(8) has the form

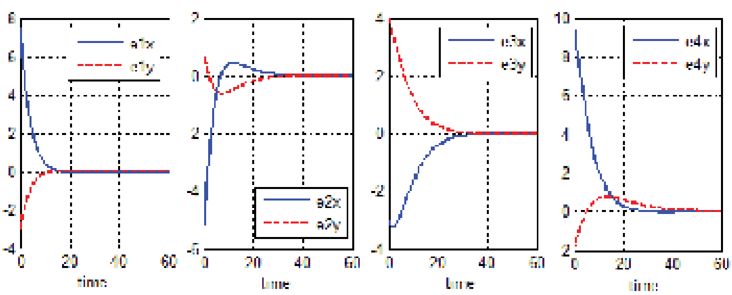
$$\dot{z} = -k((\mathcal{L}(G) \otimes I_2)z - c), \tag{9}$$

where $\mathcal{L}(G)$ is the Laplacian matrix of the FG, $z = [z_1, ..., z_n]^T$, $\otimes$ denotes the Kronecker product (Dimarogonas & Kyriakopoulos (2006a)), I_2 is the 2×2 identity matrix and $c = \left[\sum_{j \in N_1} c_{j1}, ..., \sum_{j \in N_n} c_{jn}\right]^T$.
In (Hernandez-Martinez & Aranda-Bricaire (2010)) it is shown that in the closed-loop system (1)-(8) the agents converge exponentially to the desired formation, i.e. $\lim_{t \to \infty}(z_i - z_i^*) = 0$, $i = 1, ..., n$, if the desired formation is based on a well-defined FG. The proof is based on the Laplacian Matrix and the Gershgorin circles Theorem (Bell (1972)).
Fig. 3 shows an example of the convergence to the desired formation with $n = 4$, $k = 1$ using the FG and desired vectors of positions given by Fig. 1. The initial positions in Fig. 3a (denoted by circles) are $z_1(0) = [0, -1]$, $z_2(0) = [-1, 0]$, $z_3(0) = [-4, -1]$ and $z_4(0) = [1, -3]$. We observe that the formation errors show in Fig. 3b converge to zero and therefore, all agents converge to the desired formation. The eigenvalues of $-k\mathcal{L}(G)$ are given by $0, -1, -2, -2$.
Note that the control strategies based on APF's guarantee the convergence to the desired formation. However, inter-robot collision can occur. The underlying idea of using RPF's is that every robot considers to all the others robots as mobile obstacles. The square of the distance between two robots is given by $\beta_{ij} = \|z_i - z_j\|^2$, $\forall i, j \in N$, $i \neq j$. Then, the robots R_j in danger of collision with R_i belong to the set

$$M_i = \{R_j \in N \mid \beta_{ij} \leq d^2\}, i = 1, ..., n, \tag{10}$$



(a) Trajectories of the agents in the plane



(b) Formation errors

Fig. 3. Formation control using the FG of fig. 1

where d is the diameter of the influence zone. In general, the set M_i changes in time due to the motion of agents. Then, a formation control law with collision avoidance based on APF's and RPF's is defined by

$$u_i = -\frac{1}{2}k\frac{\partial \gamma_i}{\partial z_i} - \sum_{j \in M_i} \frac{\partial V_{ij}}{\partial z_i}, i = 1, ..., n \tag{11}$$

where γ_i is the APF defined by (7) and $V_{ij}(\beta_{ij})$ is a RPF (between the pair of agents R_i and R_j) that satisfy the following properties:

1. V_{ij} es monotonously increasing when $\beta_{ij} \leq d^2$ and $\beta_{ij} \to 0$.
2. $\lim_{\beta_{ij} \to 0} V_{ij} = \infty$.
3. $V_{ij} = 0$ for $\beta_{ij} \geq d^2$, $\frac{\partial V_{ij}}{\partial z_i} = 0$ for $\beta_{ij} = d^2$.

The last condition establishes that every V_{ij} appears smoothly only within the influence area of the robot R_i. Also, it ensures that

$$\sum_{j \in M_i} \frac{\partial V_{ij}}{\partial z_i} = \sum_{j \neq i} \frac{\partial V_{ij}}{\partial z_i}. \tag{12}$$

A common function that satisfies the previous properties was proposed by Khatib (Rimon & Koditschek (1992)) as

$$V_{ij} = \begin{cases} \eta \left(\frac{1}{\beta_{ij}} - \frac{1}{d^2} \right)^2, & \text{if } \beta_{ij} \leq d^2 \\ 0, & \text{if } \beta_{ij} > d^2 \end{cases} \tag{13}$$

where $\eta > 0$. The following functions also comply with the RPF's properties.

$$V_{ij} = \begin{cases} \eta \left(\frac{1}{\beta_{ij}} - \frac{1}{d^2} \right)^r, & \text{if } \beta_{ij} \leq d^2 \\ 0, & \text{if } \beta_{ij} > d^2 \end{cases}, r = 2, 3, 4... \tag{14}$$

$$V_{ij} = \begin{cases} \eta \left(\frac{(\beta_{ij} - d^2)^2}{\beta_{ij}} \right), & \text{if } \beta_{ij} \leq d^2 \\ 0, & \text{if } \beta_{ij} > d^2 \end{cases} \tag{15}$$

Note that, in general, it is possible to rewrite $\frac{\partial V_{ij}}{\partial z_i} = 2\frac{\partial V_{ij}}{\partial \beta_{ij}}(z_i - z_j)$. Since $\beta_{ij} = \beta_{ji}$, it is satisfied that $V_{ij} = V_{ji}$ and $\frac{\partial V_{ij}}{\partial \beta_{ij}} = \frac{\partial V_{ji}}{\partial \beta_{ji}}$, $\forall i \neq j$. This ensures that the RPF's complies with the following antisymmetry property:

$$\frac{\partial V_{ij}}{\partial z_i} = -\frac{\partial V_{ji}}{\partial z_j}, \quad \forall i \neq j. \tag{16}$$

Fig. 4 shows the trajectories of three agents under the control law (11) using Khatib's RPF (13) for the case of cyclic pursuit FG (Fig. 4a) and the case of undirected cyclic pursuit FG (Fig. 4b). The initial conditions and the desired formation (horizontal line) are the same in both simulations. The agents' trajectories in Fig. 4 are modified to avoid collision. Observe that the application of different FG's to the same number of robots produces a different behavior in the closed-loop system. Note that the centroid of positions (denoted by X) in Fig. 4 remains constant for all $t \geq 0$ unlike Fig. 3, where it does not remain constant within the workspace. This property is interesting because, regardless of the individual goals of the agents, the dynamics of the team behavior remains always centered on the position of the centroid. The time-invariance of the centroid of positions is studied in Section 5. This property is inherent to the structure of the Laplacian matrix and the antisymmetry of the RPF's.

As mentioned before, the main drawback of mixing APF's y RPF's is that the agents can get trapped at undesired equilibrium points. In Dimarogonas & Kyriakopoulos (2006a), the calculation of these equilibrium points, for the case of any undirected FG, is obtained solving the equation

$$(k\mathcal{L}(G) + 2R) \otimes I_2 z = kc \tag{17}$$

where $\mathcal{L}(G)$ is the Laplacian matrix of the undirected FG, $c = [c_1, ..., c_n]$ with $c_i = \sum_{j \in N_i} c_{ji}$ and

$$(R)_{ij} = \begin{cases} \sum_{j \neq i} \frac{\partial V_{ij}}{\partial \beta_{ij}}, & \text{if } i = j \\ -\frac{\partial V_{ij}}{\partial \beta_{ij}}, & \text{if } i \neq j \end{cases}$$

For instance, analyzing the simplest case of formation with two robots R_1 and R_2, where $N_1 = \{z_2\}$ y $N_2 = \{z_1\}$, Eq. (17) reduces to

$$\left(k \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + 2 \begin{bmatrix} \frac{\partial V_{12}}{\partial \beta_{12}} & -\frac{\partial V_{12}}{\partial \beta_{12}} \\ -\frac{\partial V_{21}}{\partial \beta_{21}} & \frac{\partial V_{21}}{\partial \beta_{21}} \end{bmatrix} \right) \otimes I_2 \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = k \begin{bmatrix} c_{21} \\ c_{12} \end{bmatrix}. \tag{18}$$

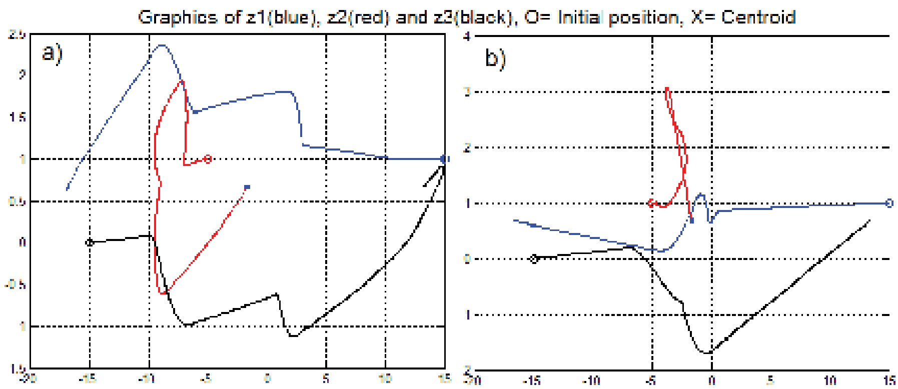


Fig. 4. Trajectories of robots in plane considering non-collision a) cyclic pursuit FG, b) undirected cyclic pursuit FG

Considering Khatib's RPF given by (13), Eq. (18) is rewritten as the following system of nonlinear simultaneous equations:

$$\begin{aligned} k\left(z_1 - z_2\right) - \delta \tfrac{4\eta}{\|z_1 - z_2\|^4}\left(\tfrac{1}{\|z_1 - z_2\|^2} - \tfrac{1}{d^2}\right)\left(z_1 - z_2\right) = kc_{21} \\ k\left(z_2 - z_1\right) - \delta \tfrac{4\eta}{\|z_1 - z_2\|^4}\left(\tfrac{1}{\|z_1 - z_2\|^2} - \tfrac{1}{d^2}\right)\left(z_2 - z_1\right) = kc_{12} \end{aligned} \tag{19}$$

where $\delta = \begin{cases} 1, & \text{if } \beta_{12} \leq d^2 \\ 0, & \text{if } \beta_{12} > d^2 \end{cases}$. The system of equations (19) is of the sixth order. However, for this particular case, clearing the term $\delta \frac{4\eta}{\beta_{12}^2}\left(\frac{1}{\beta_{12}} - \frac{1}{d^2}\right)$, it comes out that

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{y_1 - y_2 - v_{21}}{x_1 - x_2 - h_{21}} = \frac{y_2 - y_1 - v_{12}}{x_2 - x_1 - h_{12}}. \tag{20}$$

The interpretation of Eq. (20) is that, at the undesired equilibrium point, the agents R_1 and R_2 are placed on the same line as their desired positions (Fig. 5). This undesired equilibrium point is generated because both agents mutually cancel its motion when they try to move to the opposite side.

To analyze the relative position of agents R_1 and R_2, define the variables

$$p = x_1 - x_2, \quad q = y_1 - y_2 \tag{21}$$

The phase plane that represents the dynamics of these variables is shown in Fig. 6 for $k = 1$, $\eta = 10$, $d = 6$ and $c_{21} = [-3, -3]$. Off the influence zone (denote by a circle) there exists only the effect of the attractive forces generated by the APF's. Within the influence zone (inside the circle), the repulsive forces generated by the RPF's are added smoothly to the attractive forces. When $[p, q] = [0, 0]$ the distance between agents is zero and the RPF's tend to infinity. Two equilibrium points are seen in Fig. 6. One of them corresponds to the desired formation (stable node) and the other one corresponds to the undesired equilibrium point (saddle). For the case of more than two agents a similar analysis is impossible.

In general, the solution of the equation (17) is a highly complex nonlinear problem depending on the Laplacian structure and the quantity of possible combinations of RPF's that appear on these equilibria. Also, it is difficult to find general expressions similar to (17) for directed or mixed FG.

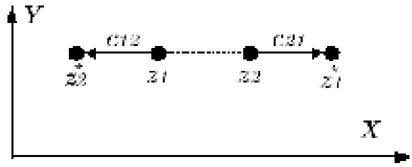


Fig. 5. Position of two robots in a undesired equilibrium point

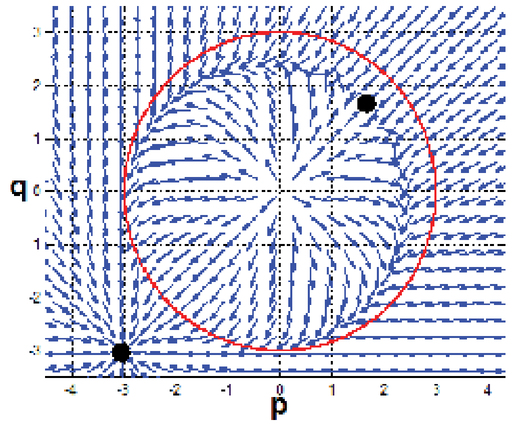


Fig. 6. Phase plane of coordinates (p, q) using the Khatib's RPF

5. Analysis of the centroid of positions

The next result was previously reported in (Hernandez-Martinez & Aranda-Bricaire (2010)) for the case of formation strategies based on APF's only. In this section, the result is extended to the case of control law (11) which mixes APF's and RPF's.

Proposition 1. *Consider the system (1) and the control law (11). Suppose that $k > 0$ and the desired formation is based on a well defined FG. Then, in the closed-loop system (1)-(11), the centroid of positions remains constant, i.e. $\bar{z}(t) = \bar{z}(0), \forall t \geq 0$ iff the FG topology satisfies the condition*

$$[1, ..., 1]\mathcal{L}(G) = [0, ..., 0]. \tag{22}$$

Proof. The dynamics of every agent R_i in the closed-loop system (1)-(11) can be written as

$$\dot{z}_i = -k\left(g_i z_i - \sum_{j\in N_i} z_j - \sum_{j\in N_i} c_{ji}\right) - \sum_{j\in M_i} \frac{\partial V_{ij}}{\partial z_i}, i = 1, ..., n \tag{23}$$

Using the property (12), note that

$$\dot{z}_i = -k\left(g_i z_i - \sum_{j\in N_i} z_j - \sum_{j\in N_i} c_{ji}\right) - \sum_{j\neq i} \frac{\partial V_{ij}}{\partial z_i}, i = 1, ..., n \tag{24}$$

Then, the dynamics of the centroid of positions is given by

$$\dot{\bar{z}}(t) = \frac{1}{n}\sum_{i=1}^{n} \dot{z}_i = -\frac{k}{n}\left(\sum_{i=1}^{n} g_i z_i - \sum_{i=1}^{n}\sum_{j\in N_i} z_j - \sum_{i=1}^{n}\sum_{j\in N_i} c_{ji}\right) - \frac{1}{n}\sum_{i=1}^{n}\sum_{j\neq i} \frac{\partial V_{ij}}{\partial z_i} \tag{25}$$

Due to the FG satisfies the closed-formation condition (3), then $\sum_{i=1}^{n}\sum_{j\in N_i} c_{ji} = 0$ and using the antisymmetry property given by (16) then $\sum_{i=1}^{n}\sum_{j\neq i}\frac{\partial V_{ij}}{\partial z_i} = 0$. Thus, equation (25) can be reduced to

$$\dot{\bar{z}}(t) = -\frac{k}{n}\left(\sum_{i=1}^{n} g_i z_i - \sum_{i=1}^{n}\sum_{j\in N_i} z_j\right) = -\frac{k}{n}\sum_{i=1}^{n}\left(g_i z_i - \sum_{j\in N_i} z_j\right) \tag{26}$$

The term $\left(g_i z_i - \sum_{j\in N_i} z_j\right)$, $i = 1, ..., n$ corresponds to the $i-th$ element of the column vector $(\mathcal{L}(G)\otimes I_2)\, z$. Thus, Eq. (26) is the sum of the elements of $(\mathcal{L}(G)\otimes I_2)\, z$ multiplied by $-\frac{k}{n}$. Therefore, Eq. (26) is equivalent to

$$\dot{\bar{z}}(t) = -\frac{k}{n}\left([1, ..., 1]\left(\mathcal{L}(G)\otimes I_2\right) z\right) \tag{27}$$

At this point, it is clear that $\dot{\bar{z}}(t) = 0$, $\forall t \geq 0$ iff condition (22) holds. Under these conditions, the centroid of positions is established by the initial positions of the robots, i.e. $\bar{z}(t) = \bar{z}(0)$ and remains constant $\forall t \geq 0$.

□

All the undirected graphs, the cyclic pursuit FG and some mixed FG satisfy the condition (22). Recall the numerical simulation of Fig. 4a (three robots in directed cyclic pursuit FG). Observe that

$$[1, 1, 1]\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} = [0, 0, 0]. \tag{28}$$

As for the case of Fig. 4b (three robots in undirected cyclic pursuit FG)

$$[1, 1, 1]\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} = [0, 0, 0]. \tag{29}$$

Therefore, as seen before, the centroid of position in both simulations remains stationary for all $t \geq 0$.

6. Repulsive vector fields based on unstable focus

Current research focuses on the design of RPF's that provide a better performance of the closed-loop system. Following this line of thought, the following Repulsive Vector Field (RVF) is proposed

$$\psi_{ij} = \begin{cases} V_{ij}\begin{bmatrix} \left(x_i - x_j\right) - \left(y_i - y_j\right) \\ \left(x_i - x_j\right) + \left(y_i - y_j\right) \end{bmatrix}, & \text{if } \beta_{ij} \leq d^2 \\ 0, & \text{if } \beta_{ij} > d^2 \end{cases} \tag{30}$$

where $V_{ij}(\beta_{ij})$ is a RPF. Note that the repulsive vector field is a clockwise unstable focus scaled by the function V_{ij} and centered at the position of another robot which appears only if a danger

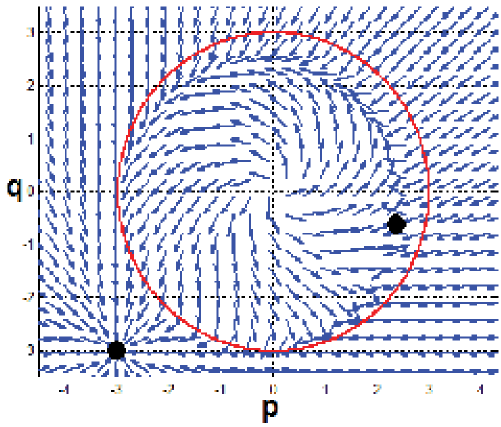


Fig. 7. Phase plane of coordinates (p, q) with RVF

of collision appears. It is interesting to point out that the vector field (30) is not obtained as the gradient of any scalar function. Using the previous RVF, we define a control law given by

$$u_i = -\frac{1}{2}k\frac{\partial \gamma_i}{\partial z_i} + \sum_{j \in M_i} \psi_{ij}, i = 1, ..., n \tag{31}$$

Note that the control law uses directly the RVF and not any partial derivative of a RPF. Therefore, the function V_{ij} is simpler to design than a standard RPF since it is only required that V_{ij} and not necessarily $\frac{\partial V_{ij}}{\partial z_i}$ vanish when $\beta_{ij} = d^2$.

In this new situation, the phase plane, for the case of two robots, of the variables (p, q) defined in (21) is shown in Fig. 7. Note that the closed-loop system (1)-(31) still displays the problem of undesired equilibria. However, the RVF provides best performance of the agent's trajectories than the classical RPF's. This comparison will be addressed by numerical simulations in Section 7.

7. Extension to the case of unicycles

In this section, the control laws developed so far are extended to the case of unicycle-type robot formations. The kinematic model of each agent or robot R_i, as shown in Fig. 8 is given by

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & 0 \\ \sin\theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ w_i \end{bmatrix}, \quad i = 1, ..., n \tag{32}$$

where u_i is the linear velocity of the midpoint of the wheels axis and w_i is the angular velocity of the robot. A celebrated result by (Brockett (1983)) states that the dynamical system (32) can not be stabilized by continuous and time-invariant control law. Because of this restriction, we will analyze the dynamics of the coordinates $\alpha_i = (p_i, q_i)$ shown in Fig. 8 instead coordinates (x_i, y_i). The coordinates α_i are given by

$$\alpha_i = \begin{bmatrix} p_i \\ q_i \end{bmatrix} = \begin{bmatrix} x_i + \ell\cos(\theta_i) \\ y_i + \ell\sin(\theta_i) \end{bmatrix}, i = 1, ..., n. \tag{33}$$

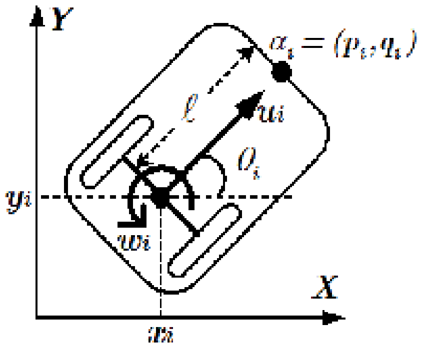


Fig. 8. Kinematic model of unicycles

The dynamics of (33) is obtained as

$$\dot{\alpha}_i = A_i(\theta_i)\,[u_i, w_i]^T, \quad A_i(\theta_i) = \begin{bmatrix} \cos\theta_i & -\ell\sin\theta_i \\ \sin\theta_i & \ell\cos\theta_i \end{bmatrix}, i = 1, ..., n \tag{34}$$

where the so-called decoupling matrix $A_i(\theta_i)$ is non-singular. The idea of controlling coordinates α_i instead of the center of the wheels axis is frequently found in the mobile robot literature in order to avoid singularities in the control law.
Following the formation control strategy with collisions avoidance presented on Section 4, the desired position of R_i, related to the coordinates α_i, is given by

$$\alpha_i^* = \frac{1}{g_i}\sum_{j\in N_i}(\alpha_j + c_{ij}), i = 1, ..., n \tag{35}$$

Then, a formation control strategy with non-collision, similar to (11) is defined as

$$\begin{bmatrix} v_i \\ w_i \end{bmatrix} = A_i^{-1}(\theta_i)\left(-\frac{1}{2}k\frac{\partial\tilde{\gamma}_i}{\partial\alpha_i} - \sum_{j\in M_i}\frac{\partial\tilde{V}_{ij}}{\partial\alpha_i}\right), i = 1, ..., n \tag{36}$$

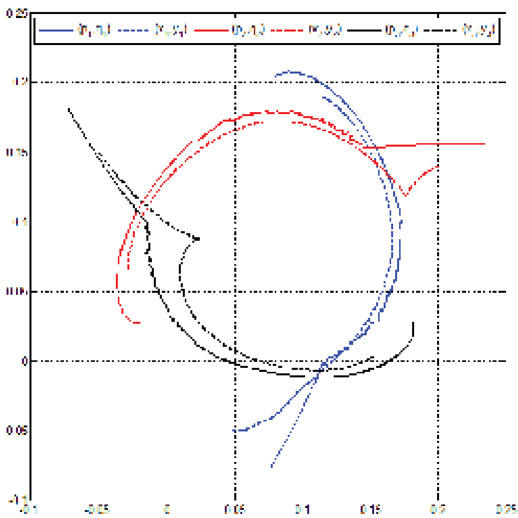
where $\tilde{\gamma}_i = \sum_{j\in N_i}\tilde{\gamma}_{ij}$ with $\tilde{\gamma}_{ij}$ similar to the case of point robots but related to coordinates α_i and $\tilde{V}_{ij}$ is a RPF also related to α_i. The dynamics of the coordinates α_i for the closed-loop system (32)-(36) is given by

$$\dot{\alpha}_i = -\frac{1}{2}k\frac{\partial\tilde{\gamma}_i}{\partial\alpha_i} - \sum_{j\in M_i}\frac{\partial\tilde{V}_{ij}}{\partial\alpha_i}, i = 1, ...n \tag{37}$$

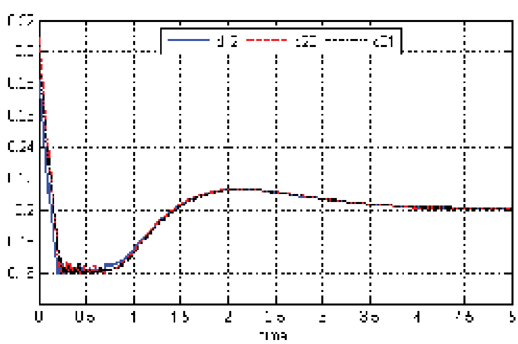
It is clear that the dynamics of coordinates α_i is the same than the case of point robots. Thus, the analysis of convergence and non-collision is reduced to the case of omnidirectional robots presented before.

Remark 1. *The control law (36) steers the coordinates α_i to a desired position. However, the angles θ_i remain uncontrolled. These angles do not converge to any specific value. Thus, the control law (36) is to be considered as a formation control without orientation.*

Fig. 9 shows a numerical simulation of the agents' trajectories and their relative distances of the closed-loop system (32)-(36) using the cyclic pursuit FG and the Khatib's RFP. In Fig. 9a, the continuous line represents the trajectories of coordinates (p_i, q_i) and the dashed line represents coordinates (x_i, y_i) of every robot. The value d_{ij} is the actual distance between agents i and j. The design parameters and the initial conditions are given by $k = 1$, $\eta = 2$, $d = 0.16$, $\ell = 0.038$, $[x_{10}, y_{10}, \theta_{10}] = [0.05, -0.05, -0.83]$, $[x_{20}, y_{20}, \theta_{20}] = [0.2, 0.14, 0.41]$, $[x_{30}, y_{30}, \theta_{30}] = [-0.05, 0.15, 2.2]$ and the desired formation is an equilateral triangle with side equal to 0.2. The agents converge to the desired formation avoiding collisions.



(a) Trajectories of the robots in plane



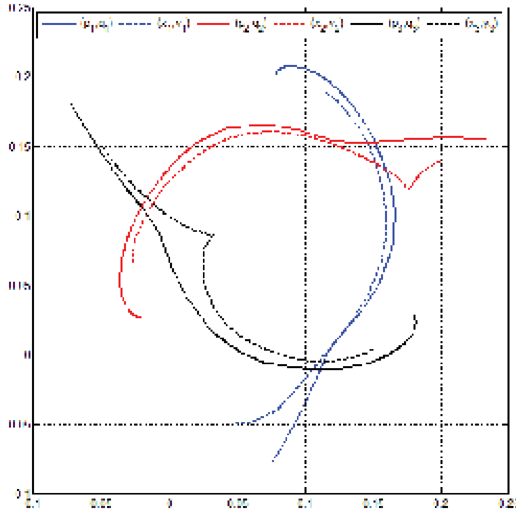
(b) Inter-robot distances

Fig. 9. Formation control with non-collision using the Khatib's RPF.

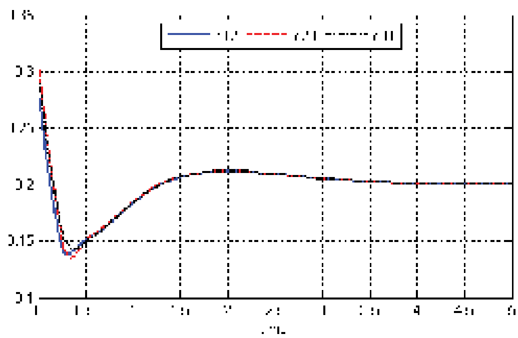
On the other hand, extending the non-collision strategy of the RVF based on an unstable scaled focus for the case of unicycles, we obtain

$$\begin{bmatrix} v_i \\ w_i \end{bmatrix} = A_i^{-1}(\theta_i) \left(-\frac{1}{2} k \frac{\partial \tilde{\gamma}_i}{\partial \alpha_i} - \sum_{j \in M_i} \tilde{\psi}_{ij} \right) \tag{38}$$

where $\tilde{\psi}_{ij}$ is similar to ψ_{ji} shown in (31) but related to coordinates α_i. Fig. 10 shows a numerical simulation for the closed-loop system (32)-(38) with the same parameters, desired formation and initial conditions than the previous case.



(a) Trajectories of the robots in plane



(b) Inter-robot distances

Fig. 10. Formation control with non-collision using the RVF.

To establish a comparison between the non-collision strategies, define an error performance index for coordinates α_i as follows:

$$J(e) = \frac{1}{t_f} \int_0^{t_f} \left(\sqrt{e_1^2 + e_2^2 + e_3^2} \right) dt \tag{39}$$

where $e_i = \|\alpha_i - \alpha_i^*\|$. Let $t_f = 5$, we obtain $J(e) = 0.1386$ for the case of the Khatib's RPF and $J(e) = 0.1173$ for the RVF strategy. Clearly, the the latter case presents the best results, which is reflected on less oscillations on the agents' trajectories while avoiding collisions.

8. Experimental work

This section presents some experiments of formation control with collision avoidance in an experimental setup consisting in three unicycle-type robots manufactured by Yujin (model: YSR-A) and a computer vision system composed by an UNIQ digital video camera (model: UF1000-CL) connected to an ARVOO video processor (model: Leonardo CL). The vision system captures and processes the position of two white circle marks placed on every robot (the marks represent the position of (x_i, y_i) and α_i) at 100 Hz rate. The position and orientation of each robot are obtained using this information . The images are processed in a Pentium4-based PC where the control actions u_i and w_i are also transformed into the desired angular velocities for the robot wheels using the parameters $\ell = 2.8$cm, $r = 2.2$cm and $L = 7.12$cm where r is the radius of the wheels and L is the distance between the two wheels. These commands are sent by a RF module to every robot.

Fig. 11 shows an experiment with Kathib's RPF strategy with $k = 0.2$, $\eta = 0.4$ and $d = 0.16$. The initial conditions (in meters and radians) are given by $[x_{10}, y_{10}, \theta_{10}] = [-0.0064, -0.2563, 1.5923]$, $[x_{20}, y_{20}, \theta_{20}] = [0.1813, 0.2136, -0.3955]$ and $[x_{30}, y_{30}, \theta_{30}] = [-0.2374, 0.1566, -1.1659]$. The desired formation is an equilateral triangle with side equal to 0.2.

Fig. 12 shows a second experiment with the RVF strategy using $k = 0.2$, $\eta = 1$ and $d = 0.16$. The desired formation is the same as in previous case. The initial conditions are given by $[x_{10}, y_{10}, \theta_{10}] = [0.0105, -0.2296, -0.8327]$, $[x_{20}, y_{20}, \theta_{20}] = [0.2047, 0.1445, 0.4030]$ and $[x_{30}, y_{30}, \theta_{30}] = [-0.2236, 0.1723, 2.1863]$. The simulation results are dashed lines whereas experimental results are continuous lines.

In both experiments, the control signals were normalized to

$$[\bar{v}_i, \bar{w}_i]^T = \frac{\mu}{\sqrt{\|F_i\|^2 + \varepsilon}} A_i^{-1}(\theta_i) F_i, i = 1, 2, 3 \tag{40}$$

where $\mu = 0.1$, $\varepsilon = 0.0001$, $F_i = -\frac{1}{2}k\frac{\partial \tilde{\gamma}_i}{\partial \alpha_i} - \sum_{j \in M_i} \frac{\partial \tilde{V}_{ij}}{\partial \alpha_i}$ for the first experiment with Kathib's RPF and $F_i = -\frac{1}{2}k\frac{\partial \tilde{\gamma}_i}{\partial \alpha_i} - \sum_{j \in M_i} \tilde{\psi}_{ij}$ for the second experiment with the RVF strategy. The normalization has two proposes. Firstly, to avoid actuator saturation for large values of $\|\alpha_i - \alpha_i^*\|$. Secondly, to compensate the adverse effects of friction and actuators' dead zone. We observe that the inter-agent distances converge to the desired value. However, the motion of coordinates α_i and (x_i, y_i) displays best performance in the case of the RVF strategy. Finally, fig. 13 shows the posture of the robots at final time recorded by the vision system in the second experiment. We observe that the front mark of every robot (coordinates α_i) converge to the desired formation.

9. Conclusion

Convergence to the desired formation and collision avoidance are the most important requirements on a formation control strategy. The analysis is complex because the control laws are decentralized considering the closed-loop behavior for any number of robots. Formation graphs are useful to describe the possible interaction between robots and provide mathematical tools for the analysis of the system. Although decentralized control strategies based on Artificial Potential Fields can be easily implemented, the complexity of the calculation of undesired equilibria remains an open problem.

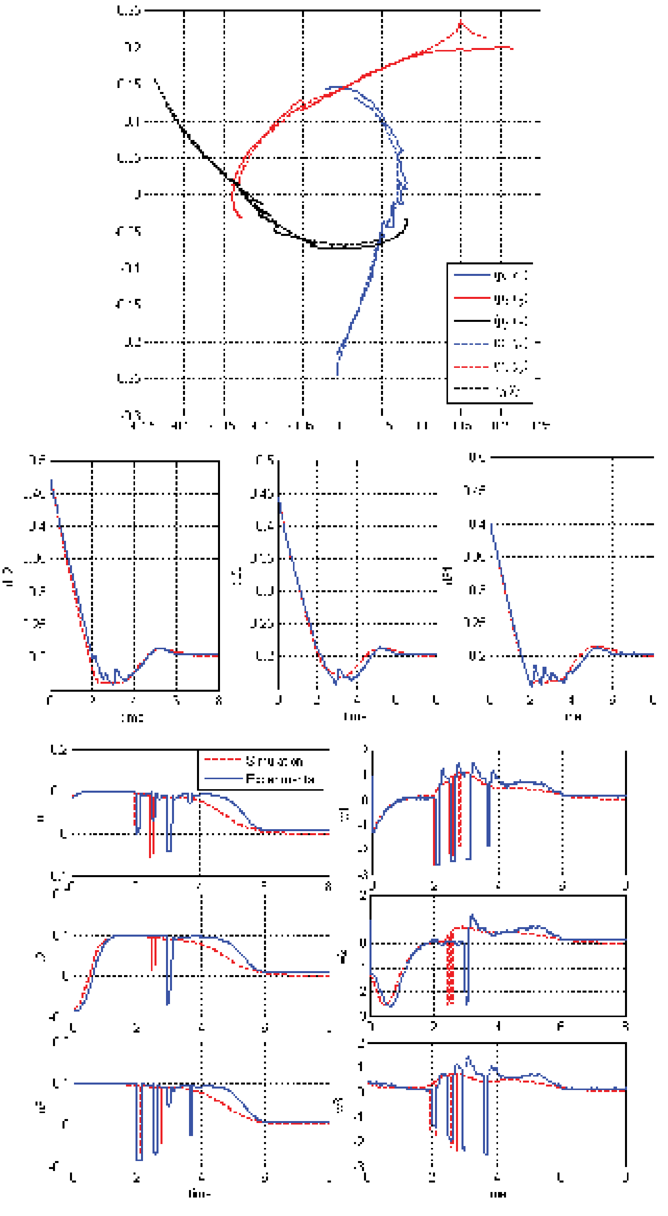


Fig. 11. Experiment 1 using the standard RPF of Khatib

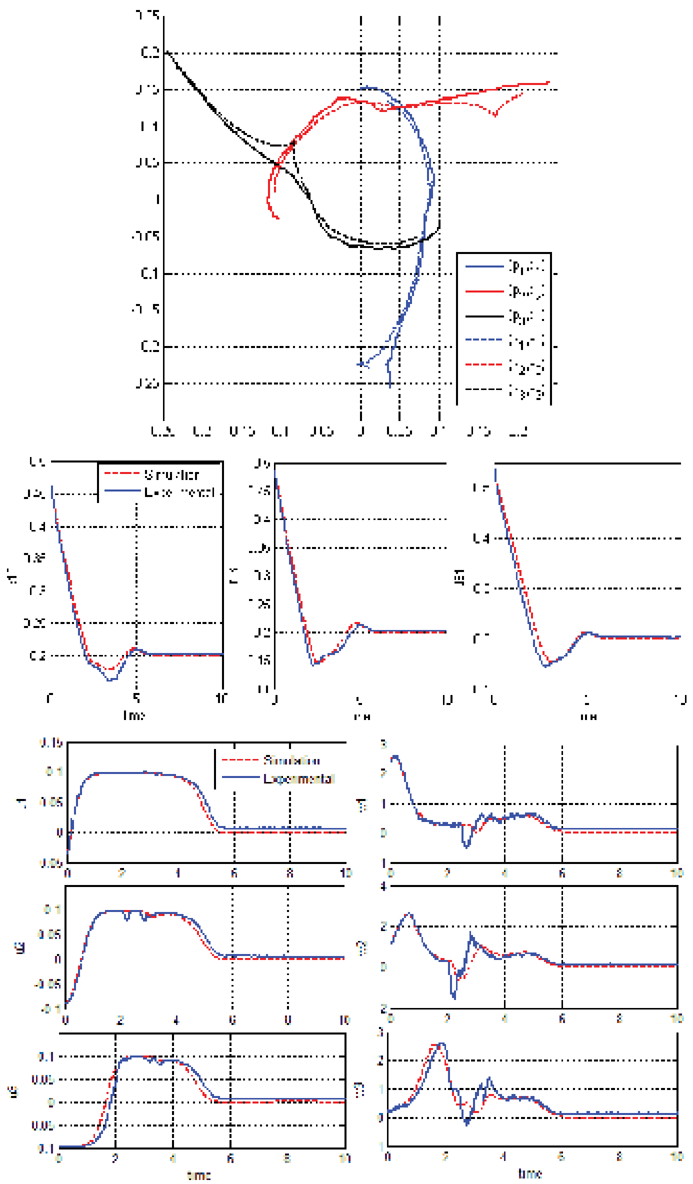


Fig. 12. Experiment 2 using the RVF strategy

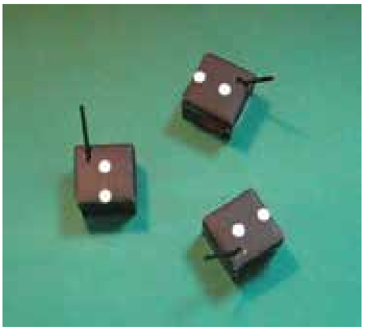


Fig. 13. Final positions of robots in the experiment 2

This chapter presents some alternatives to ensure convergence, modifying the standard design of potential functions. Also, we contribute to the state of art of formation control with the analysis of time invariance of the centroid of positions. This property is interesting because the dynamics of the team behavior remains centered on the position of the centroid, although every agent obeys a decentralized control strategy. Another contribution is a novel non-collision strategy based on RVF's instead of the repulsive forces from the negative gradient of a RPF. Numerical simulations and real-time experiments for the case of three unicycle-like robots show a better performance of the proposed strategies than the standard methodology.

10. References

Ando, H., Oasa, Y., Suzuki, I. & Yamashita, M. (1999). Distributed memoryless point convergence algorithm for mobile robots with limited visibility, *IEEE Transactions on Robotics and Automation* Vol. 15(No. 5): 818–828.

Arai, T., Pagello, E. & Parker, L. E. (2002). Guest editorial advances in multirobot systems, *IEEE Transactions on Robotics and Automation* Vol. 18(No. 5): 655–661.

Asahiro, Y., Asama, H., Suzuki, I. & Yamashita, M. (1999). Improvement of distributed control algorithms for robots carrying an object, *International Conference on Systems, Man, and Cybernetics*, IEEE, Tokyo, Japan, pp. 608–613.

Baillieul, J. & McCoy, L. (2007). The combinatorial graph theory of structured formations, *Conference on Decision and Control*, IEEE, New Orleans, USA, pp. 3609–3615.

Balch, T. & Arkin, R. (1998). Behavior-based formation control for multirobot teams, *IEEE Transactions on Robotics and Automation* Vol. 14(No. 3): 926–939.

Barnes, L., Fields, M. & Valavanis, K. (2007). Unmanned ground vehicle swarm formation control using potential fields, *Mediterranean Conference on Control and Automation*, IEEE, Athens, Greece, pp. 1–8.

Bell, H. (1972). Gerschgorin's theorem and the zeros of polynomials, *American Matematics* Vol. 3(No. 1): 292–295.

Brockett, R. (1983). Asymptotic stability and feedback stabilization, *in* R. Brockett, R. Millman & H. Sussmann (eds), *Differential Geometric Control Theory*, Birkhauser, Boston, USA, pp. 181–191.

Cao, Y. U., Fukunaga, A. & Kahng, A. (1997). Cooperative mobile robotics: Antecedents and directions, *Autonomous Robots* Vol. 4(No. 1): 7–27.

Cao, Z., Xie, L., Zhang, B., Wang, S. & Tan, M. (2003). Formation constrained multi-robot system in unknown environments, *International Conference on Robotics and Automation*, IEEE, Taipei, Taiwan, pp. 735–740.

Chen, Y. Q. & Wang, Z. (2005). Formation control: A review and a new consideration, *International Conference on Intelligent Robots and Systems*, IEEE/RSJ, Edmonton, Canada, pp. 3181–3186.

Das, A. & Fierro, R. (2003). Hybrid control of reconfigurable robot formations, *American Control Conference*, IEEE, Denver, USA, pp. 4607–4612.

Desai, J. (2002). A graph theoretic approach for modeling mobile robot team formations, *Journal of Robotic Systems* Vol. 19(No. 11): 511–525.

Desai, J., Kumar, V. & Ostrowski, J. (1998). Controlling formations of multiple mobile robots, *International Conference on Robotics and Automation*, IEEE, Leuven, Belgium, pp. 2864–2869.

Desai, J., Ostrowski, J. & Kumar, V. (2001). Modeling and control of formations of nonholonomic mobile robots, *IEEE Transactions on Robotics and Automation* Vol. 17(No. 6): 905–908.

Dimarogonas, D. & Kyriakopoulos, K. (2005). Formation control and collision avoidance for multi-agent systems and a connection between formation infeasibility and flocking behavior, *Conference on Decision and Control*, IEEE, Seville, Spain, pp. 84–89.

Dimarogonas, D. & Kyriakopoulos, K. (2006a). Distributed cooperative control and collision avoidance for multiple kinematic agents, *Conference on Decision and Control*, IEEE, San Diego, USA, pp. 721–726.

Dimarogonas, D. & Kyriakopoulos, K. (2006b). On the state agreement problem for multiple unicycles with varying communication links, *Conference on Decision and Control*, IEEE, San Diego, USA, pp. 4283–4288.

Dimarogonas, D., Kyriakopoulos, K. & Theodorakatos, D. (2006). Totally distributed motion control of sphere world multi-agent systems using decentralized navigation functions, *International Conference on Robotics and Automation*, IEEE, Orlando, USA, pp. 2430–2435.

Dimarogonas, D., Loizou, S., Kyriakopoulos, K. & Zavlanos, M. (2006). A feedback stabilization and collision avoidance scheme for multiple independent non-point agents, *Automatica* Vol. 42(No. 2): 229–243.

Do, K. (2006). Formation control of mobile agents using local potential functions, *American Control Conference*, IEEE, Minneapolis, USA, pp. 2148–2153.

Do, K. (2007). Formation tracking control of unicycle-type mobile robots, *International Conference on Robotics and Automation*, IEEE, Roma, Italy, pp. 2391–2396.

Egerstedt, M. & Hu, X. (2001). Formation constrained multi-agent control, *IEEE Transactions on Robotics and Automation* Vol. 17(No. 6): 947–951.

Fax, J. & Murray, R. (2004). Information flow and cooperative control of vehicle formations, *IEEE Transactions on Automatic Control* Vol. 49(No. 9): 1465–1476.

Feddema, J., Lewis, C. & Schoenwald, D. (2002). Decentralized control of cooperative robotic vehicles: Theory and application, *IEEE Transactions on Robotics and Automation* Vol. 18(No. 5): 852–864.

Francis, B., Broucke, M. & Lin, Z. (2004). Local control strategies for groups of mobile autonomous agents, *IEEE Transactions on Automatic Control* Vol. 49(No. 4): 622–629.

Ge, S. & Fua, C. (2005). Queues and artificial potential trenches for multirobot formations, *IEEE Transactions on Robotics* Vol. 21(No. 4): 646–656.

Gennaro, M. D. & Jadbabaie, A. (2006). Formation control for a cooperative multi-agent system using decentralized navigation functions, *American Control Conference*, IEEE, Minneapolis, USA.

Hendrickx, J., Yu, Y. C., Fidan, B. & Anderson, B. (2007). Rigidity and persistence of meta-formations, *Conference on Decision and Control*, IEEE, New Orleans, USA, pp. 5980–5985.

Hernandez-Martinez, E. & Aranda-Bricaire, E. (2008a). Non-collision conditions in formation control using a virtual leader strategy, *XIII CLCA and VI CAC*, IFAC, Merida, Venezuela, pp. 798–803.

Hernandez-Martinez, E. & Aranda-Bricaire, E. (2008b). Non-collision conditions in multi-agent robots formation using local potential functions, *International Conference on Robotics and Automation*, IEEE, Pasadena, USA, pp. 3776–3781.

Hernandez-Martinez, E. & Aranda-Bricaire, E. (2009a). Marching control of unicycles based on the leader-followers scheme, *35th Annual Conference of the IEEE Industrial Electronics Society*, IEEE, Porto, Portugal, pp. 2265–2270.

Hernandez-Martinez, E. & Aranda-Bricaire, E. (2009b). Multi-agent formation control with collision avoidance based on discontinuous vector fields, *35th Annual Conference of the IEEE Industrial Electronics Society*, IEEE, Porto, Portugal, pp. 2283 – 2288.

Hernandez-Martinez, E. & Aranda-Bricaire, E. (2010). Decentralized formation control of multi-agent robots systems based on formation graphs, *XIV Latinamerican Congress of Automatic Control and XIX ACCA Congress*, IFAC, Santiago, Chile, pp. 1–6.

Itkis, U. (1976). *Control Systems of Variable Structure*, Wiley.

Kim, D., Lee, H., Shin, S. & Suzuki, T. (2005). Local path planning based on new repulsive potential functions with angle distributions, *International Conference on Information Technology and Applications*, IEEE, Sydney, NSW, pp. 9–14.

Lafferriere, G., Caughman, J. & Williams, A. (2004). Graph theoretic methods in the stability of vehicle formations, *American Control Conference*, IEEE, Boston, USA, pp. 3729–3734.

Lee, M. & Park, M. (2003). Artificial potential field based path planning for mobile robots using a virtual obstacle concept, *Advanced Intelligent Mechatronics*, IEEE/RSJ, pp. 735–740.

Lee, M. & Park, M. (2004). Real-time path planning in unknown environment and a virtual hill concept to escape local minima, *Annual Conference of Industrial Electronics Society*, IEEE, Busan, Korea, pp. 2223–2228.

Leonard, N. E. & Fiorelli, E. (2001). Virtual leaders, artificial potentials and coordinated control of groups, *Conference on Decision and Control*, IEEE, Orlando, USA, pp. 2968–2973.

Li, Y. & Chen, X. (2005a). Extending the potential fields approach to avoid trapping situations, *International Conference on Intelligent Robots and Systems*, IEEE/RSJ, Edmonton, Canada, pp. 1386–1391.

Li, Y. & Chen, X. (2005b). Stability on multi-robot formation with dynamic interaction topologies, *International Conference on Intelligent Robots and Systems*, IEEE/RSJ, Edmonton, Canada, pp. 394–399.

Loizou, S. & Kyriakopoulos, K. (2002). Closed loop navigation for multiple holonomic vehicles, *International Conference on Intellingence Robots and Systems*, IEEE/RSJ, France, pp. 2861–2866.

Loizou, S., Tannert, H., Kumar, V. & Kyriakopoulos, K. J. (2003). Closed loop navigation for mobile agents in dynamic environments, *International Conference on Intelligent Robots and Systems*, IEEE/RSJ, Las Vegas, USA, pp. 3769–3774.

Loizou, S., Tannert, H. & Kyriakopoulos, K. (2003). Nonholonomic stabilization with collision avoidance for mobile robots, *International Conference on Intelligent Robots and Systems*, IEEE/RSJ, Las Vegas, USA, pp. 1220–1225.

Mallapragada, G., Chattopadhyay, I. & Ray, A. (2006). Autonomous navigation of mobile robots using optimal control of finite state automata, *Conference on Decision and Control*, IEEE, San Diego, USA, pp. 2400–2405.

Marshall, J., Broucke, M. & Francis, B. (2004). Formations of vehicles in cyclic pursuit, *IEEE Transactions on Automatic Control* Vol. 49(No. 11): 1963–1974.

Muhammad, A. & Egerstedt, M. (2004). Connectivity graphs as models of local interactions, *Conference on Decision and Control*, IEEE, Atlantis, Paradise Island, Bahamas, pp. 124–129.

Ogren, P. & amd N.E. Leonard, E. F. (2004). Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment, *IEEE Transactions on Automatic Control* Vol. 49(No. 8): 1292–1302.

Ogren, P. & Leonard, N. (2003). Obstacle avoidance in formation, *International Conference on Robotics and Automation*, IEEE, Taipei, Taiwan.

Olfati-Saber, R. & Murray, R. (2002). Distributed cooperative control of multiple vehicle formations using structural potential functions, *The 15th IFAC World Congress*, IFAC, Barcelona, Spain, pp. 2864–2869.

Olfati-Saber, R. & Murray, R. (2003). Agreement problems in networks with directed graphs and switching topology, *Conference on Decision and Control*, IEEE, Hawaii, USA, pp. 4126–4132.

Reynolds, C. (1987). Flocks, birds, and schools: A distributed behavioral model, *Computer Graphics* Vol. 21(No. 1): 25–34.

Rimon, E. & Koditschek, D. (1992). Exact robot navigation using artificial potential functions, *IEEE Transactions on Robotics and Automation* Vol. 5(No. 8): 501–518.

Schneider, F. & Wildermuth, D. (2005). Experimental comparison of a directed and a non-directed potential field approach to formation navigation, *International Symposium on Computational Intelligence in Robotics and Automation*, IEEE, pp. 21–26.

Spears, W., Spears, D., Hamann, J. & Heil, R. (2004). Distributed, physycs-based control of swarms of vehicles, *Autonomous Robots* Vol. 17(No. 2): 137–162.

Swaroop, D. & Hedrick, J. (1996). String stability of interconnected systems, *IEEE Transactions on Automatic Control* Vol. 41(No. 3): 349–357.

Tanner, H. (2004). Flocking with obstacle avoidance in switching networks of interconnected vehicles, *International Conference on Robotics and Automation*, IEEE, New Orleans, USA, pp. 3006–3011.

Tanner, H. & Kumar, A. (2005). Towards decentralization of mutli-robot navigation functions, *International Conference on Robotics and Automation*, IEEE, Barcelona, Spain, pp. 4132–4137.

Tanner, H., Kumar, V. & Pappas, G. (2002). Stability properties of interconnected vehicles, *International Symposium on Mathematical Theory of Networks and Systems*, IEEE/RSJ, France, pp. 394–399.

Tanner, H., Kumar, V. & Pappas, G. (2004). Leader-to-formation stability, *IEEE Transactions on Robotics and Automation* Vol. 20(No. 3): 443–455.

Yamaguchi, H. (2003). A distributed motion coordination strategy for multiple nonholonomic mobile robots in cooperative hunting operations, *Robotics and Autonomous Systems* Vol. 43(No. 1): 257–282.

Yao, J., Ordoñez, R. & Gazi, V. (2006). Swarm tracking using artificial potentials and sliding mode control, *Conference on Decision and Control*, IEEE, San Diego, USA, pp. 4670–4675.

7

Negotiation in Multi-Agent Environments

Dorin Militaru
Amiens School of Management
France

1. Introduction

Internet offers many choices of products, services and content. But the multitude of choices has altered the manner in which customers choose and buy products and services. Internet searching agents are tools allowing consumers to compare on-line Web-stores' prices for a product. In the field of electronic commerce, consumers can already benefit from the use of searching agents that automatically query on-line sellers' catalogues in order to gather information about available products. These agents have different degrees of sophistication and differ in their ability to process information. Intelligent agents in artificial intelligence are closely related to agents in economics. They can adapt to the needs of different users, it can learn new concepts and techniques, it can anticipate the needs of the user, and it can take initiative and make suggestions to the user. Negotiation is an important mechanism for resolving conflicts between individual or economic agents. Intelligent agents can provide services in filtering data, searching for information, online tutoring, and negotiation.

In this chapter, we present a study that investigates how costly information could affect the performance of several types of searching agents in electronic commerce environments. The existing agents base their search on a predefined list of Web-stores and, as such, they can be qualified as fixed-sample size searching agents. However, with the implementation of new Internet pricing schemes, this search rule evolve toward more flexible search methods allowing for an explicit trade-off between the communication costs and the product price. In this setting, the sequential optimal search rule is a possible alternative. Nevertheless, its adoption depends on its expected performance. The main goal is to analyze the relative performances of two types of search agents on a virtual market with costly information. The following section reviews several existing agents to highlight their flaws and strengths. This review aims to test the theory that agents are a viable alternative to search engines.

2. Overview of intelligent agent

During the last years, the remarkable growth of the Internet caused an increasing demand for more advanced tools capable to assist net-surfers in their search for useful information. In the field of electronic commerce, consumers can already benefit from the use of searching agents that automatically query on-line sellers' catalogues in order to gather information about available products. An example of simple searching agent is BargainFinder who gathers price information about books on behalf of her user. More recent and sophisticated searching agents, as Mpire [www.mpire.com], can screen a wide range of products (computer, software, cosmetics, wine, televisions, etc.) not only by considering their prices, but also by considering their quality attributes. However, the use of electronic agents is still in a nascent phase and, even if we already imagine a near future when consumers and companies will interact by their virtual counterparts (Guttman et al., 1998), it is not clear yet what such an electronic market will look like.

Information agents are agents that have access to at least one, and potentially many information sources. They are able to manipulate information obtained from these sources in order to answer queries posed by users and other information agents (Papazoglou et al., 1992). The information sources may be of many types (example, traditional databases). A number of studies have been made of information agents, including a theoretical study of how agents are able to incorporate information from different sources (Gruber, 1991), (Levy et al., 1994). Another important system in this area is called Carnot (Huhns et al., 1992), which allows pre-existing and heterogeneous database systems to work together to answer queries that are outside the scope of any of the individual databases. (Franklin et al., 1996) define an agent as an autonomous process running on a computer that is able to sense and react to its environment. This agent is able to run without interaction with the user and must therefore be able to make decisions about the environment and the realization of its goals. (Hermans, 1996) and (Jennings et al., 1996) outline several characteristics of software agents. They believe agents require social ability to interact with the user. The agent must be responsive and proactive so it can sense and react to its environment and the users needs. They must be temporally continuous, goal oriented, adaptive and they should be autonomous and able to collaborate with the user and other agents to perform tasks. (Negroponte, 1997) believes that agents are useful not because they can perform tasks a user could not perform on their own using other tools, but because they perform tasks, the user finds trivial or mundane. By delegating the task of information retrieval to the agent, the user is able to direct their attention to tasks that are more enjoyable or make better use of their time. This author suggests that increasing the amount of information that the user has access to on the Internet does not improve the Internet as an information resource. Rather, it makes the process of finding accurate reliable resources that match the user's information need even more difficult. He contends that users do not need more information. They need a relatively small amount of information that is concise, accurate and relevant to them. Therefore, a method of filtering the abundant information, so that only poignant information remains, is required.

Information management in agents is not restricted to the management of user queries. Agents also use a knowledge base to help them manage information, and create a model of the user's information needs. (Maes, 1997) identifies an approach to the design of agents that is "knowledge-based". It involves the agent accruing its knowledge base over time. (Maes, 1997), uses a machine learning approach to the design of agents. We notice that an agent can learn in several ways. Firstly, the agent may monitor user behavior and actions, with the aim of detecting patterns that it can emulate and automate. Secondly, the agent may seek advice from other sources or agents that provide the same service to their user and have more experience.

Negotiation is a critical activity in electronic environments. In general negotiation is defined as an iterative process which aims to achieve a mutually beneficial deal for the seller and buyer (Fisher et al. 1991). Negotiations can be done manually in electronic commerce, for example, by emails, but this is not time and cost-effective. There is a need for automated negotiation. For example, a negotiation process using two intelligent agents to negotiate a solution autonomously would be more efficient and objective. Software agents are currently being used for information retrieval and for offering recommendations such as finding product information, comparing product prices, and offering suggestions on product and services based on customer's interest and preferences (Murugesan, 1998).

Negotiation theory was first used in game theory. Negotiation in game theory is different from real life. The game software does not have to learn to make good decisions. The game has already all needed information for decision making. Still in game theory, we can find two good ideas. Firstly, in game theory, agents have knowledge of other agent negotiation and decision making algorithm. Secondly, game agents have all needed information for negotiating on a certain place and they can use common knowledge instead of its own. Auctioning is most popular and mostly used negotiation strategy. Usually agents are auctioning on goods or services. User creates agent with certain conditions, values, and bidding strategies and agent is bidding in auction instead of the user.

Negotiation Agents can be also used to improve regular person-to-person negotiation. InterNeg project is good example (www.InterNeg.org). These way agents can be used to search, sort, choose, or recommend some information, solution, but final decision is left to the users. Negotiation does not have to end-up with rejecting or accepting. The negotiation result can be data for next negotiation. In this case, agents work as a good secretary.

Automating negotiation for business to business trade has the potential to transform the way in which business is conducted, and improve negotiation trade transactions in terms of speed, efficiency, and quality of contract agreement. There have been recent research advances towards the construction of software capable of automating electronic commerce negotiation. However, there is still much work to be done for this to be achieved. Complicating the automation of trade is the context of ecommerce. Two key features of the electronic commerce trade context are identified by (Zwass, 1996) to be the ability to share and use business information and the evolution and maintenance of business relationships. Business information is the set of data, facts or opinions which directly relates to the conduct of the electronic commerce transaction. An example of this type of information is market information (market price). Another example is the participant information, what ones know about the other participants in the negotiation. Electronic marketplaces such as (Bogdanovych et al., 2004) facilitate ecommerce trade. Within this environmental context, the informational context is rich in diverse, dynamic, and easily accessible business information. A significant problem in automating negotiation in electronic commerce is accounting for the informational context that environments such as (Bogdanovych et al., 2004) provide. The consequence of accounting for the rich sources of information in automating negotiation is that the decisions made reflect those that are appropriate with regards to the environment. The challenge is how to enable automated negotiation software with this property.

With very few exceptions (Kephart et al., 1999), the research available on agent-based markets considered search costs as negligible in the analysis of market evolution. However, we believe that with the evolution of Internet pricing mechanisms and agent intermediation, the search costs will become an important aspect in modeling agent-based markets. Indeed, these two phenomena could change the basic rules of agent interactivity and market structure as we conceive of it today. It is clear that the fast spread of the Internet at global level balanced with bandwidth limited extensibility raises problems related to the Internet providers' quality of service. To avoid Internet congestion, the regulators consider the introduction of new pricing schemes based on the real use of network resources (Mcknight et al., 1997). If such pricing mechanisms will be adopted, the buying agents will cause to their users some "search cost" related to Internet resources usage for communication scopes. Even if this cost is low compared to the price of the product, it is clear that, in the long run, if the aim of the buying agent is to maximize her user's utility, she should consider the incurred search cost in her decisions. Unfortunately the searching rule currently used by the agents does not take these problems into account. Therefore it is important to analyze different other rules that could be used by agents in the future. Indeed, the existing agents base their search on a predefined list of commercial sites and they do not explicitly incorporate the costs associated to the realization of their objective. This kind of search is based on a fixed sample-size searching rule. In this paper, the agents using this rule are named FSS buying agents. However, the economic theory shows that, on a market with costly information, the optimal searching rule is a sequential rule based on a reservation price (Rothschild et al., 1974). The RP buying agents (using an optimal sequential searching rule) would represent a possible alternative to replace the FSS buying agents. Indeed, in presence of search costs, the consumer's objective is not only to minimize the product's purchase price, but also to minimize the communication costs.

In the present study, we compare the FSS buying agents and RP buying agents performances in term of total costs (product price plus total searching cost) paid by the consumers. We try to determinate if it is profitable for the existing searching agents to adopt an optimal sequential searching rule. For this purpose, we analyze a market where FSS buying agents and RP buying agents coexist. In our model, we assume that the search costs correspond to a constant unit cost of communication paid by the agents to have access to the price information of a Website. A lot of research in economics was concerned with game theoretic equilibrium of markets where buyers' search for price information is costly (Diamond, 1971), (Stiglitz, 1989), or where the buyers are in a situation of information asymmetry (Varian, 1980). Closer to our model is the work of (Stahl, 1989). However, while this author considered that fixed-sample-size searchers have complete information of all market prices, we limited their perceptiveness to a reduced set of sellers. In addition, we considered a unit demand and we introduced reservation values for buyers. The main reason for these modifications was that we wanted our model to be closer to the market scenario for electronic agents we have imagined. In this model, we showed that symmetric mixed-strategy equilibrium exists and we compare the performance of the FSS buying agents and RP buying agents. At the theoretical equilibrium, we show that the agents using a sequential searching rule allow consumers to pay lower total costs. On the existing Internet markets, the restrictive assumptions of a game theoretic analysis are very unlikely to be met. In order to relax them, we simulate a market where the sellers use a dynamic pricing strategy, and where the RP buying agents have access only to partial information when they determine their reservation price. The simulations confirm the good performance of the sequential searching rule and show that the agents could adopt it in the future.

Literature indicates that agents are a useful alternative to search engines because the user can delegate the information retrieval task to them. Autonomy, temporal continuity and adaptiveness are some of the aspects of agent functionality that enable an agent to work effectively on behalf of a user.

The chapter proceeds as follows. The market model for electronic agents that we propose and the game-theoretic equilibrium are presented in Sections 2 and 3. In Section 4, we compare the performances of the two types of searching agents on the market. In Section 5, we present and analyze the results of our simulations concerning the performance of the two searching rules in various configurations of the market. We present our conclusion in Section 6 together with some closing remarks on future developments.

3. Theoretical analysis of the agent-based market equilibrium

In this section, we describe an Internet agent-based market where agents representing companies or consumers transact a homogenous product (good or service). On the market, agents act on their users' behalf by trying either to maximize profits or to find "good deals".[1] In what follows, we will refer to agents owned by companies as "*selling agents*" and to agents owned by consumers as "*buying agents*". On the analyzed market, we assume that the number M of buying agents is significantly higher than the number N of selling agents ($M >> N$). The number of agents on the market is considered fix over time. As the commercialized product is homogenous, the N selling agents compete with each other on the market in terms of price. We suppose that each selling agent has the same constant unit production cost v and the same fix production cost FC. For convenience, these two production costs are considered to be nil.[2] Given that the reproduction costs are nil, each selling agent is able to satisfy at any time all the buying demands she receives. In our model, we completely ignore the possibility of price negotiation between buying agents and selling agents. The only permitted "bargaining" consists in selling agents making a take-it or leave-it offer to any buying agent who visits them.

On the considered market, all buying agents incur in their search process a positive cost c for each selling agent they visit but they differ by their search strategy. Thus, two types of buying agents coexist on the market: fixed-sample-size buying agents (FSS buying agents) and reservation-price buying agents (RP buying agents).[3] Each buying agent, whatever her type, has a unit demand constraint by a reservation value r. This reservation value is assumed to be equal among the buying population and should be interpreted, as the maximum-posted price of the selling agent a buying agent is willing to accept. The FSS buying agents correspond to the existing searching agents and use a predefined list of n transaction partners. These agents are in proportion w_1 on the market. A FSS buying agent will buy at the lowest price she was proposed, if lower than her reservation value r. We assume here that the sample-size n is exogenously given and is identical for all FSS buying agents. However, even if the list of selling agents is predefined, that list varies from one buying agent to another, which allows a posteriori for different buying agents to find

---

[1] Even if real users are an essential component for the existence of an agent-based market, the present study exclusively focuses on the strategic interaction between agents. Indeed, we assume that the agents are endowed with all relevant information they need to define their goals.

[2] Our analysis still holds for any positive value of these two parameters. Moreover, one can note that the assumption of zero production costs seems rather adequate on the Internet especially for products such as information goods (digital books, videos, etc.). Indeed, for these products the reproduction costs are nil (Varian, 2000).

[3] In economic theory, this two search rules are often qualified as "non-optimal" for the fixed-sample-size search rule and "optimal" for the reservation-price search rule (Rothschild, 1974).

different prices for the product. The remaining buying agents (proportion $w_2 = 1 - w_1$) are RP buying agents and proceed to a sequential search for transaction partners. If we consider *F(p)* to be the cumulative price dispersion on the market, the reservation-price rule used by the RP buying agent can be formalized as follows. When facing a current posted price *q*, the RP buying agent expected gain for searching once more is given by:

$$g(q) = \int_0^q (q - p)dF(p) = \int_0^q F(p)dp \tag{2}$$

where *q* is the current price and *p* is the posted price of the next visited selling agent. From an economic point of view, it is optimal to keep searching while the expected gain of an additional search exceeds the search cost (Rothschild, 1974). As a result, the reservation price *R* must satisfy: *g(R) = c*.

4. Market equilibrium

In order to compute the game theoretic equilibrium of the market model described in the previous section, we proceed to a strategic analysis of selling agents behavior on the market. With the previous market description, selling agents play a Nash non-cooperative game among themselves and a Stackelberg game against buying agents who take the prices they find as given. Because there is no pure-strategy equilibrium in our model (Varian, 1980), (Stahl, 1989), we analyzed symmetric mixed-strategy equilibrium with each selling agent randomly choosing a price with respect to the same equilibrium price distribution. We note by *f(p)* the equilibrium price distribution, and by *F(p)* the associated cumulative price distribution. Facing the same market price distribution and being characterized by the same set of parameters (search cost and reservation value), there is no reason for identical RP buying agents to solve their search problem in different manners. As a result, in equilibrium, all the RP buying agents choose the same reservation price *R*.

With this remark we can proceed to the computation of the market equilibrium. First, some preliminary characteristics of the Nash-Equilibrium probability distribution are established. Second, the optimal behavior of selling agents and the market equilibrium are computed. During these two steps the reservation price *R* is assumed exogenous which confers a conditional aspect to the analyzed market equilibrium. This aspect is emphasized by using the notation *F(p, R)* for the Nash-Equilibrium cumulative price distribution conditional on *R*. Finally, the existence of an endogenous reservation price *R* is established and an analytical expression for this price is computed.

4.1 Some preliminary results

Before computing the symmetric Nash equilibrium in mixed-strategy conditional on *R*, two properties of such a distribution need to be established, more precisely that it is atomless (Lemma 1.) and that it has an upper bound (Lemma 2.). Only intuitive proofs of these results will be given here. For more formalized explanations, see Stahl [11] and Varian [13].

Lemma 1. If *f*(*p*, *R*) is a Nash-Equilibrium distribution conditional on reservation price *R*, there can be no mass point[4] in this distribution, except at the lowest price.

[4] Recall that *p* is a mass point of a probability density function if there is positive probability concentrated at *p*.

Proof. Let's assume the probability distribution has a mass point at p^* (p^* other than the lowest price). Then, for any perceptiveness $n \geq 2$ of FSS buying agents, there is some positive probability that a buying agent visiting n selling agents only finds out about selling agents charging price p^*. In this particular scenario, if one of these selling agents lowers her price by an epsilon, she gains all the FSS buying agents as certain customers, thus increasing her profits. As a result, we can conclude that there cannot be a two-price equilibrium or any equilibrium price distribution with a mass point other than at the lowest price.

Lemma 2. If $F(p, R)$ is a Nash-Equilibrium distribution conditional on reservation price R, the maximal element of its support, denoted $p_{\max}$, is the minimum value between R and r: $p_{\max} = \min\{R, r\}$.

Proof. To prove this result two aspects need to be established: $p_{\max}$ can neither be higher, nor lower than $\min\{R, r\}$. Let's first consider that $p_{\max} < \min\{R, r\}$. Provided that the equilibrium distribution is atomless, when fixing a price $p_{\max}$, selling agents only win profits from RP buying agents. However, by slightly increasing their prices they lose no customers and they increase their profits. This situation can't be equilibrium. Let's now consider the opposite case where $p_{\max} > \min\{R, r\}$. If $r < R$ the proof is obvious as selling agents can only expect no sales by fixing a higher price than r, as no buying agent will pay more than her reservation value r for the product. If $r > R$, fixing the price $p_{\max}$ causes selling agents to lose all their customers: RP buying agents won't buy as the price is higher than their reservation price R, while for FSS buying agents the probability to find a lower price is one ($F(p, R)$ is atomless). So, the expected profits of the selling agent are nil. By lowering the maximal price to R, selling agents can expect to attract at least $1 / N$ RP buying agents, which gives them positive profits. With these remarks, one can conclude that $p_{\max} = \min\{R, r\}$.

4.2 Computation of the conditional mixed strategy equilibrium

The two properties of the probability distribution established earlier enable us to compute the equilibrium probability distribution conditional on R. As we showed that all the prices practiced on the market are lower than R (cf. Lemma 2.), in equilibrium, all the RP buying agents buy from the first store they visit. It follows that only FSS buying agents engage in active search activities. The condition for $f(p)$ to be an equilibrium probability distribution for the selling agent i ($i = 1, ..., N$) is that it should maximize her expected profits. In other terms, given that the other $N\text{-}1$ selling agents use mixed-strategies by randomly choosing a price according to $f(p)$, the selling agent i's profits by choosing any price from the support of $f(p)$ should be the same. Otherwise, she would be better off by choosing the price with the highest expected profits. For this to be true, for some positive value K, we need two conditions to be satisfied: (i). $\pi_i(p_i) \leq K$ for every possible price p and (ii). $\pi_i(p_i) = K$ for all the prices p effectively played in equilibrium.

By fixing a price of p_i while all other sellers use a mixed-strategy $f(p)$, the selling agent i's expected profits depend on the demand she will face: $\pi_i(p_i) = p_i D_i(p_i, p_{-i})$ where p_{-i} is the vector of prices chosen by all the sellers and $D_i(p_i, p_{-i}) = D_{i,w_1}(p_i, p_{-i}) + D_i$ where $D_{i,w_1}(p_i, p_{-i})$ is the expected demand from FSS buying agents and $D_{i,w_2}(p_i, p_{-i})$ is the expected demand from RP buying agents. As FSS buying agents have a n-perceptiveness, for such an agent to buy from selling agent i, two conditions need to be satisfied. First, selling agent i should be in the n-size sample, which happens with probability n / N. Second, p_i needs to be the lowest price between the n prices sampled, which happens with[5] probability $[1 - F(p_i)]^{n-1}$. As a result $D_{i, w_1}(p_i, p_{-i}) = w_1M(n / N)[1 - F(p_i)]^{n-1}$. As p_i is always lower than R and so are the other selling agents' prices on the market, selling agent i can expect that a $1 / N$ share of RP buying agents will buy from its store independently of the offered price.[6] The corresponding demand is: $D_{i, w_2}(p_i, p_{-i}) = w_2M(1 / N)$. It follows that:

$$\pi_i(p_i) = p_i\left[w_1M\frac{n}{N}[1-F(p_i,R)]^{n-1} + w_2M\frac{1}{N}\right] \tag{3.1}$$

In order for $F(p_i, R)$ to be a Nash-Equilibrium distribution, the selling agent i must expect equal profits by choosing any price with a non-negative support. In other words, we need the following condition to be true for all the prices p_i effectively played in equilibrium: $\pi_i(p_i) = \pi$. As for $p_i = p_{\max}$ we have $F(p_{\max}) = 1$, we can compute the value of π:

$$\pi = \frac{p_{\max}w_2M}{N} \tag{3.2}$$

Solving $\pi_i(p_i) = \pi$, we obtain:

$$F(p_i,R) = 1 - \left[\frac{w_2}{nw_1}\left(\frac{p_{\max}}{p_i} - 1\right)\right]^{\frac{1}{n-1}} \tag{3.3}$$

The density function $f(p_i, R)$ is the derivative of $F(p_i, R)$ with respect to p_i:

$$f(p_i,R) = \frac{w_2}{n(n-1)w_1}\left[\frac{w_2}{nw_1}\left(\frac{p_{\max}}{p} - 1\right)\right]^{-\frac{n-2}{n-1}}\frac{p_{\max}}{p^2} \tag{3.4}$$

In order for $F(p_i, R)$ to be a true probability distribution, it should only take positive values. So, we can compute the lowest bound of the support of f, $p_{\min}$, whose value is such that $F(p_{\min}, R) = 0$, which yields:

$$p_{\min}(R) = \frac{p_{\max}w_2}{nw_1 + w_2} \tag{3.5}$$

At this stage we have a complete characterization of the Nash Equilibrium distribution conditional on R.

4.3 Existence of an endogenous R

Until now, we have supposed R exogenously given. For our problem to be completely solved, we have to verify that there exists an endogenous reservation price R consistent with

[5] Given that there are no mass points in the probability distribution, we can ignore the possibility of a tie at a price of p_i.

[6] Given that at equilibrium all the prices will be lower than R, RP buying agents are equally likely to buy from any seller.

the previously determined distribution $F(p, R)$. As stated earlier, RP buying agents choose their reservation price to be the solution R^* of the following equation, provided that $R^* \leq r$:

$$g(R) = \int_{p_{min}(R)}^{R} F(p,R)dp = c \tag{3.6}$$

Note that RP buying agents will never choose a reservation price higher than their reservation value r. The condition (3.6) can be rewritten as:

$$\begin{aligned} H\left(p,R^*,c\right) &= 0 \quad \text{where} \\ H(p,R,c) &= \int_{p_{min}(R)}^{R} F(p,R)dp - c \end{aligned} \tag{3.7}$$

In order to facilitate the analysis, we first concentrate on the case where we have a solution $R^* \leq r$. It follows that $p_{max} = R^*$. Using the equations (3.3) and (3.5), the equation (3.7) becomes:

$$H(p,R^*,c) = \int_{\frac{w_2 R^*}{nw_1 + w_2}}^{R^*} \left[1 - \left[\frac{w_2}{nw_1} \left(\frac{R^*}{p} - 1 \right) \right]^{\frac{1}{n-1}} \right] dp - c \tag{3.8}$$

After computations, we obtain an analytical expression for R^*:

$$R^* = \frac{c}{1 - \alpha - \beta I_n} \tag{3.9}$$

where $\alpha = \dfrac{w_2}{nw_1 + w_2}$, $\beta = \left(\dfrac{w_2}{nw_1} \right)^{\frac{1}{n-1}}$ and $I_n = \displaystyle\int_0^{(1/\alpha)-1} \frac{t^{1/(n-1)}}{(t+1)^2} dt$

As the uniqueness of such a solution is obvious, a necessary condition for this R^* to be a consistent reservation price is that it should be positive:

$$1 - a - bI_n > 0 \tag{3.10}$$

Let's now consider what happens if the solution R^* is higher than r. In this case, the RP buying agents will always fix their reservation price at r and p_{max} will equal r. Therefore, under the condition (3.10), we can formally define a consistent reservation price R for RP buying agents: $R = \min\{R^*, r\}$. So, the complete description of the market equilibrium for selling agents behavior is given by the equations (3.3).

5. Compared expected performances of the two types of buying agents

In this section we compare the performance of RP buying agents versus FSS buying agents in term of total costs paid by the consumers. The consumer's total cost is obtained by summing the price paid for the product and the total searching cost incurred by the agent.

The expected total cost for a consumer who uses an FSS buying agent is given by the following expression:

$$CT_{FSS}(n) = cn + \overline{p}_n \tag{4.1}$$

where $CT_{FSS}(n)$ represents the expected total cost for a searching agent having a sample size n, $\overline{p}_n$ represents the expected price paid by the consumer and cn is the total search cost paid for using this agent. On the considered market, the expected price $\overline{p}_n$ paid by an FSS buying agent with a sample size N is given by:

$$\overline{p}_n = \int_{p_{min}}^{p_{max}} [1 - F(p)]^{n-1} nf(p)pdp \tag{4.2}$$

The expected total cost for this agent is:

$$CT_{FSS}(n) = cn + \int_{p_{min}}^{p_{max}} [1 - F(p)]^{n-1} nf(p)pdp \tag{4.3}$$

In order to determine the expected total cost of using an RP buying agent, we previously showed that at equilibrium such agents buy in the first store that they visit. Thus, an RP buying agent will pay only one search cost c. The expected total cost is then given by the following expression:

$$CT_{RP} = c + \int_{p_{min}}^{p_{max}} pf(p)dp \tag{4.4}$$

In order to compare the performance of these two types of searching agents, we must analyze the sign of the difference between their total costs:

$$\Delta CT(n) = CT_{FSS}(n) - CT_{RP} = c(n-1) + \int_{p_{min}}^{R} (1 - F(p))[(1 - F(p))^{n-1} - 1]dp \tag{4.5}$$

Replacing $F(p)$ by its analytical expression (cf. equation (3.3)), we can show that $\Delta CT(n)$ is positive for every sample size of FSS buying agents, every proportion of the two agent types on the market (w_1 and w_2) or every search cost value c. We can conclude that, at equilibrium, the RP buying agent always allow the consumers to pay lower total costs than those paid by with an FSS buying agent. The theoretical model shows that it is profitable for the searching agents to evolve toward the use of a sequential searching rule with reservation price in order to satisfy at best consumers needs. This result is coherent with the economic literature, which considers the RP rule to be optimal for searching on a market with costly information.

6. Proposed framework

The results obtained in the previous section concerning the superiority of the RP buying agents over the FSS buying agents' are strongly dependent on the assumptions made.

Indeed, as the RP buying agents are perfectly informed about the market structure and there is common knowledge of rationality, they are able to determine seller's mixed-strategy equilibrium and fix their reservation price according to this distribution. These assumptions are particularly restrictive on real market. Indeed, in practice it is unlikely for the RP buying agents to have a perfect knowledge about the market structure and sellers price their products according to the equilibrium's distribution probability previously established (in order to compute the equilibrium price distribution the sellers also need to be perfectly informed about the market structure). A more realistic analysis seems necessary, especially as some theoretical studies (Gastwirth, 1976) show that small anticipation error can considerably deteriorate the RP buying agents performance. Before concluding about the superiority of the RP buying agents' searching strategy over FSS buying agents in the context of a real market, it's important to study the impact of the introduction of partial information. Indeed, partial information about the market could increase considerably the RP buying agents total search cost whereas the fixed sample-size searching rule limits the FSS buying agents' search cost. According to other studies that try to relax the assumptions of the economic models (Kephart et al., 1999) in analyzing Internet markets, we simulate a market where the selling agents try to maximize their profit by using a dynamic pricing strategy requiring only partial information, and where RP buying agents determine their reservation price based on a partial degree of information on market's price distribution.

6.1 Sellers' pricing strategy and the RP buying agents' information

In our simulations, we consider that the selling agents use a dynamic pricing strategy based on the gradient heuristic. Sellers' incentive to adopt such pricing strategy on the Internet has already been presented in other research (Kephart et al., 1999). Indeed, this dynamic pricing strategy does not require information about the market structure and it allows the sellers to enjoy high profits by facilitating tacit collusions. During a given period, a selling agent using the gradient heuristic decide to reduce or to increase her price according to the observed variations of her profits following her last price modification. For example, if the preceding decision of the selling agent was to increase her price and she observed an increase in her profits, then she will reproduce the same action (price increase). In the opposite case, if the price modification lee to a reduction in her profits, she will do an opposite action, a price reduction. To define more precisely this selling agents' pricing strategy, it is necessary to fix two parameters: the amount Δ of the possible price variations and the announced price in the first period. In our simulations, the announced price, at the beginning, by a selling agent is chosen randomly in the interval $[0, r]$. With each price modification, the increment or the decrement Δ is selected randomly between values 1, 2, 3, 4 or 5 if the selling agent decides to increase her price or -1, -2, -3, -4 or -5 if the selling agent decides to decrease her price. We can note that this pricing strategy is easy to implement by a seller as it does not require any preliminary information neither on the competitors' behavior, nor on the demand.

As we have already mentioned, on traditional markets it is difficult for the RP buying agents to use the real price distribution in order to determine their reservation price. Within the framework of our simulations, we assume that the RP buying agents can calculate their reservation price by using a partial degree of information γ on the prices in a period. This information degree is assumed to be identical for all the RP buying agents evolving on the market, and can be interpreted as the number of prices an agent used to calculate her reservation price. For example, if $\gamma = 0{,}1$, the searching agents use 10% of the sellers' prices so us determine her reservation price.

6.2 Simulations

In our simulations, we fix the number of buying agents to M = 1000 and the numbers of selling agents to N = 20. We consider that the two types of searching agents are in equal proportion on the market $w_1 = 0{,}5$ and $w_2 = 0{,}5$ (i.e., 50% FSS buying agents and 50% RP buying agents). The buying agents' reservation value is equal to r = 50 and the communication cost is equal to c = 2. As we only want to study the performances of the two types of agents, the only parameters that we vary from one simulation to another are the RP buying agents' information degree γ and the FSS buying agents' sample size n. In our simulations, these two parameters can take the values γ = {0,1; 0,3; 0,5} and n = {2; 4; 6}. We simulate the market evolution on T = 30 000 periods and we consider that the selling agents can readjust their price rate at λ = 0,1 per period.

			Information RP (γ)		
			0,1	0,3	0,5
FSS simple-size (n)	2	FSS average price	16,54	19,10	27,60
		RP average price	11,05	8,93	11,50
	4	FSS average price	9,54	11,64	16,61
		RP average price	10,71	9,71	11,53
	6	FSS average price	5,84	7,99	11,97
		RP average price	9,49	9,39	11,47

Table 1. Average prices paid by the FSS and RP buying agents

			Information RP (γ)		
			0,1	0,3	0,5
FSS simple-size (n)	2	FSS search	2	2	2
		RP search	3,42	3,68	4,49
	4	FSS search	4	4	4
		RP search	3,42	3,72	4,24
	6	FSS search	6	6	6
		RP search	3,02	3,71	4,25

Table 2. Average search numbers for the FSS and RP buying agents

The results of our simulations in term of average paid prices, average number of searches carried out and average total costs for the two types of agent are presented in the tables below. We can notice that for the RP search agents the average number of searches is simply obtained by dividing the difference between the average price paid and the average total cost by the communication cost.

			Information RP (γ)		
			0,1	0,3	0,5
FSS simple-size (*n*)	2	Total cost FSS	20,54	23,10	31,60
		Total cost RP	17,89	16,30	20,49
	4	Total cost FSS	17,54	19,64	24,61
		Total cost RP	17,55	17,15	20,06
	6	Total cost FSS	17,84	19,99	23,97
		Total cost RP	15,90	16,82	19,93

Table 3. Average total costs for the FSS and RP buying agents

In term of average total costs, simulations show that RP buying agents' performance is quite good compared to FSS buying agents' performance. Moreover, this result still holds even if these agents have access to little information on market prices. However, we can notice that more RP agents have a significant degree of information on market prices, more their compared performance to FSS agents' is important. A somewhat counter intuitive result is that on increase in the RP buying agents' information degree leads to price raise on the market. This result is explained by the adaptability of the seller's strategy to the market configuration. Indeed, when the RP buying agents' information increases, the sellers are encouraged to specialize either on high prices, or on low prices, which leads to an increase in the prices dispersion on the market. In this case, FSS buying agents have a high probability to have high prices in their sample n, whereas RP buying agents search more. Nevertheless, when the FSS buying agents' sample size increases the latter finds lower prices. Our simulations confirm that RP buying agents can represent a good alternative for searching commercial information on the network.

7. Conclusion

In this study, we analyzed the possible evolution of the fixed sample-size searching rule used by the existing agents towards an optimal sequential searching rule with reservation price. Indeed, as it was underlined, a costly information search requires the agents to make a trade-off between communication costs and product prices. In order to study this possible evolution, we proposed a market model where FSS and RP buying agents coexist. A theoretical analysis of the market equilibrium showed that RP buying agents always allowed the consumers to pay lower total costs than FSS buying agents. Moreover, this result holds when some assumptions of the game theory model are relaxed by simulating

the dynamics of a market where the selling agents use a dynamic pricing strategy and where the RP buying agents can fix their reservation price only an the basis of partial information about the prices charged on the market. We can thus conclude that in the future the sequential searching rule could be a good alternative for searching agents.
This study has reviewed the main concepts and issues associated with the theory and practice of intelligent agents. It has drawn together a very wide range of material, and has hopefully provided an insight into what an agent is, how the notion of an agent can be formalized, how appropriate costly information search agent can be designed and implemented. The subject matter of this review is important because it is increasingly felt, both within academia and industry, that intelligent agents will be a key technology as computing systems become ever more distributed, interconnected, and open. In such environments, the ability of agents to autonomously plan and pursue their actions and goals, to cooperate, coordinate, and negotiate with others, and to respond flexibly and intelligently to dynamic and unpredictable situations will lead to significant improvements in the quality and sophistication of the software systems that can be conceived and implemented, and the application areas and problems which can be addressed.
Several extensions of our study are possible. First of all, it would be necessary to relax the assumption that the product sold on the market is an homogeneous good. Indeed, on the Internet, one of the most important dimensions of competition between the sellers is the quality of their products (Bakos et al., 1997), (Delong, 1998). Moreover, the Internet offers new differentiation opportunity for sellers, like personalization or bundling. Another extension is the introduction of a more complex transaction mechanism to model the negotiation between sellers and the searching agents. Negotiation is an important aspect in distribution of rare resources on the network. A third extension would be the introduction of an intermediary that will charge a fee to give agents access to his data-bases containing more relevant information about the market. The presence of this type of intermediation could evolve in the future (Bailey, 1997) and has the potential to influence agents' searching rule.

8. References

Bakos J.Y. (1997). Reducing Buyer Search Costs: *Implications For Electronic Marketplaces. Management Science*, Vol. 43, No. 12, pp. 1676-1692

Bailey J., Bakos, J.Y. (1997). An Exploratory Study Of The Emerging Role Of Electronic Intermediaries, *International Journal Of Electronic Commerce*, Vol. 1, No. 3, pp. 7-21

Bogdanovych A., Berger H., Simoff S., Sierra C. (2004). Ecommerce environments as 3d electronic institutions, *In Proceedings of the IADIS International e-Commerce Conference*, pp. 237–242, Lisbon, Portugal.

Campbell M., Gozinski A. (2001). An Information Agent Model, *Technical Report*, TR C01/19, Deakin University. Melbourne.

Cournot A.G. (1838). Recherches Sur Les Principes Mathematiques De La Theorie De La Richesse, *Hachette*, Paris

Delong J.B., Froomkin, A.M. (1998). The Next Economy?. In Hurley D., Kahin B. & H. Varian (Eds.), *Internet Publishing And Beyond: The Economics Of Digital Information And Intellectual Property*, Mit Press, Cambridge

Diamond P.A. (1971). A Model Of Price Adjustment, *Journal Of Economic Theory*, Vol. 3, No. 2-4, pp. 156-168

Fisher R., Ury W., Patton B. (1991). Getting to Yes: Negotiating Agreement Without Giving *In, 2nd ed. Boston, Houghton Mifflin*

Franklin S., Graesser A. (1996). Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, *Third International Workshop on Agent Theories*, Architectures, and Languages, Springer-Verlag.

Gastwirth J.L. (1976). On Probabilistic Consumer Search For Information. *Quarterly Journal Of Economics*, Vol. 90, No. 1, pp. 38-50

Gruber T. R. (1991). The role of common ontology in achieving sharable, reusable knowledge bases. In Fikes, R. and Sandewall, E., editors, *Proceedings of Knowledge Representation and Reasoning (KR&R-91)*. Morgan Kaufmann Publishers: San Mateo, CA

Guttman R., Moukas A., Maes P. (1998). Agent-Mediated Electronic Commerce: A Survey, *Knowledge Engineering Review*, Vol. 13, No. 2, pp. 147-159

Hermans B. (1996). Intelligent software agents on the internet: An inventory of currently offered functionality in the information society and a prediction of (near-) future developments, *Ph.D. Thesis*, Tilburg University. Tilburg.

Huhns M. N., Jacobs N., Ksiezyk T., Shen W. M., Singh M. P., and Cannata P. E. (1992). Integrating enterprise information models in Carnot. In *Proceedings of the International Conference on Intelligent and Cooperative Information Systems*, pages 32-42, Rotterdam, The Netherlands.

Jennings N., Wooldridge M. (1996). Software Agents, *IEE Review*, January, pp. 17-20.

Kephart J., Greenwald A. (1999). Shopbot And Pricebots, *Proceedings Of The International Joint Conferences On Artificial Intelligence*, Stockholm, Sweden

Levy A. Y., Sagiv Y., Srivastava D. (1994). Towards efficient information gathering agents. In Etzioni, O., editor, *Software Agents - Papers from the 1994 Spring Symposium (Technical Report SS-94-03)*, pages 64-70. AAAI Press.

Lieberman H. (1997). Autonomous Interface Agents, Henry Lieberman, Internet, *lieber.www.media.mit.edu/*

Maes P. (1997). Agents that Reduce Work and Information Overload, *Software Agents,* J. Bradshaw. AAAI, Menlo Park.

Mcknight L.W., Bailey J.P. (1997). Internet Economics, *The Mit Press*, Cambridge

Murugesan S., (1998). Intelligent Agents on the Internet and Web, *Proceedings of the IEEE Region IO Conference*, Tencon'98, New Delhi

Negroponte N. (1997). Agents: From Direct Manipulation to Delegation, *Software Agents.* J. Bradshaw. AAAI and MIT Press, Menlo Park.

Papazoglou M. P., Laufman S. C., Sellis T. K. (1992). An organizational framework for cooperating intelligent information systems. *Journal of Intelligent and Cooperative Information Systems*, 1(1):169-202.

Rothschild M. (1974). Searching For The Lowest Price When The Distribution Of Prices Is Unknown, *The Journal Of Political Economy*, Vol. 82, No. 4, pp. 689-711

Stahl D.O. (1989). Oligopolistic Pricing With Sequential Consumer Search, *American Economic Review*, Vol. 79, No. 4, pp. 700-712

Stiglitz J.E. (1989). Imperfect Information In The Product Market, *In Handbook Of Industrial Organization*, Schmalensee R. & R.D. Willig, North Holland, New-York, pp. 769-847

Varian H. (1980). A Model Of Sales, *American Economic Review*, Vol. 70, No. 4, pp. 651-659
Varian H. (2000). Buying, Sharing And Renting Information Goods, *Working Paper Series*, University of California at Berkeley.
Voorhees E. M. (1994). Software agents for information retrieval. In Etzioni, O., editor, *Software Agents - Papers from the 1994 Spring Symposium (Technical Report SS-94-03)*, pages 126-129. AAAI Press.
Zwass V. (1996). Electronic commerce: Structures and issues, *Journal of Electronic Commerce*, vol. 1(1), pp. 3–23

8

Reasoning about Resource-Sensitive Multi-Agents

Norihiro Kamide
Waseda Institute for Advanced Study, Waseda University,
1-6-1 Nishi Waseda, Shinjuku-ku, Tokyo 169-8050
Japan

1. Introduction

Aims. Formalizing common knowledge reasoning in multi-agent systems is of growing importance in Computer Science, Artificial Intelligence, Economics, Philosophy and Psychology. Obtaining a concrete logical foundation for common knowledge reasoning plays an important role for formal treatment and verification of multi-agent systems. For this reason, formalizing common knowledge reasoning is also a traditional issue for *multi-agent epistemic logics* (Fagin et al., 1995; Halpern & Moses, 1992; Lismont & Mongin, 1994; Meyer & van der Hoek, 1995). The aim of this paper is to formalize more fine-grained common knowledge reasoning by a new logical foundation based on Girard's linear logics.

Common knowledge. The notion of common knowledge was probably first introduced by Lewis (Lewis, 1969). This notion is briefly explained below. Let A be a fixed set of agents and α be an idea. Suppose that α belongs to the common knowledge of A, and i and j are some members of A. Then, we have the facts "both i and j know α", "i knows that j knows α" and "j knows that i knows α". Moreover, we also have the facts "i knows that j knows that i knows α", and so on. Then, these nesting structures develop an infinite hierarchy as a result.

Iterative interpretation. Suppose that the underlying multi-agent logic has the knowledge operators $\heartsuit_1, \heartsuit_2, ..., \heartsuit_n$, in which a formula $\heartsuit_i\alpha$ means "the agent i knows α." The common knowledge of a formula α is defined below. For any $m \geq 0$, an expression K^m means the set

$$\{\heartsuit_{i_1}\heartsuit_{i_2}\cdots\heartsuit_{i_m} \mid each\ \heartsuit_{i_t}\ is\ one\ of\ \heartsuit_1, ..., \heartsuit_n\ and\ i_t \neq i_{t+1}\ for\ all\ t = 1, ..., m-1\}.$$

When $m = 0$, $\heartsuit_{i_1}\heartsuit_{i_2}\cdots\heartsuit_{i_m}$ is interpreted as the null symbol. The common knowledge $\heartsuit_c\alpha$ of α is defined by using an infinitary conjunction $\bigwedge$ as the so-called *iterative interpretation of common knowledge*: $\heartsuit_c\alpha := \bigwedge\{\heartsuit\alpha \mid \heartsuit \in \bigcup_{m\in\omega} K^m\}$. Then, the formula $\heartsuit_c\alpha$ means "α is common knowledge of agents."

Common knowledge logics. *Common knowledge logics* (CKLs) are multi-agent epistemic logics with some knowledge and common knowledge operators (Fagin et al., 1995; Halpern & Moses, 1992; Lismont & Mongin, 1994; Meyer & van der Hoek, 1995). So far, CKLs have been studied based on classical logic (CL). On the other hand, CL is not so appropriate for expressing more fine-grained reasoning such as resource-sensitive, concurrency-centric and constructive reasoning. Thus, CKLs based on non-classical logics have been required for expressing such fine-grained reasoning.

Linear logics. Girard's *linear logics* (LLs) (Girard, 1987), which are most promising and useful non-classical logics in Computer Science, are logics that can naturally represent the concepts of "resource consumption" and "parallel execution" in concurrent systems. Applications of linear logics to programming languages have successfully been studied by many researchers (see e.g., (Miller, 2004) and the references therein). Combining LLs with some knowledge and common knowledge operators is thus a natural candidate for realizing an expressive and useful common knowledge logic. Indeed, *intuitionistic linear logic* (ILL) and *classical linear logic* (CLL) (Girard, 1987), which were introduced as refinements of intuitionistic logic (IL) and CL, respectively, are more expressive than IL and CL, respectively.

Multi-agent linear logics. A multi-agent epistemic linear logic with a common knowledge operator has not yet been proposed. A reason may be that to prove the cut-elimination and completeness theorems for such an extended multi-agent linear logic is difficult because of the complexity of the traditional setting of a common knowledge operator in sequent calculus. This paper is trying to overcome such a difficulty by introducing a new simple formulation of a *fixpoint operator*, which can be used as a common knowledge operator, and by using a phase semantic proof method. *Phase semantics*, which was originally introduced by Girard (Girard, 1987), is known to be a very useful Tarskian semantics for linear and other substructural logics. It was shown by Okada that the cut-elimination theorems for CLL and ILL can be proved by using the phase semantics (Okada, 1999; 2002). This paper uses Okada's method effectively to obtain the cut-elimination and completeness theorems for the proposed multi-agent linear logics.

New fixpoint operator. In the following, we explain the proposed formulation of fixpoint operator. The symbol ω is used to represent the set of natural numbers, and the symbol N is used to represent a fixed nonempty subset of ω. The symbol K is used to represent the set $\{\heartsuit_i \mid i \in N\}$ of modal operators, and the symbol K^* is used to represent the set of all words of finite length of the alphabet K. For example, $\{\iota\alpha \mid \iota \in K^*\}$ denotes the set $\{\heartsuit_{i_1} \cdots \heartsuit_{i_k}\alpha \mid i_1, ..., i_k \in N, k \in \omega\}$. Remark that K^* includes $\emptyset$ and hence $\{\iota\alpha \mid \iota \in K^*\}$ includes α. Greek lower-case letters ι and κ are used to represent any members of K^*. The characteristic inference rules for a fixpoint operator $\heartsuit_F$ are as follows:

$$\frac{\iota\kappa\alpha, \Gamma \Rightarrow \gamma}{\iota\heartsuit_F\alpha, \Gamma \Rightarrow \gamma}\ (\heartsuit_F\text{left}) \qquad \frac{\{\ \Gamma \Rightarrow \iota\kappa\alpha \mid \kappa \in K^*\ \}}{\Gamma \Rightarrow \iota\heartsuit_F\alpha}\ (\heartsuit_F\text{right}).$$

These inference rules imply the following axiom scheme: $\heartsuit_F\alpha \leftrightarrow \bigwedge\{\iota\alpha \mid \iota \in K^*\}$. Suppose that for any formula α, f_α is a mapping on the set of formulas such that $f_\alpha(x) := \bigwedge\{\heartsuit_i(x \wedge \alpha) \mid i \in \omega\}$. Then, $\heartsuit_F\alpha$ becomes a fixpoint (or fixed point) of f_α.

Interpretations of new fixpoint operator. The axiom scheme presented above just corresponds to the iterative interpretation of common knowledge. On the other hand, if we take the singleton $K := \{\heartsuit_0\}$, then we can understand $\heartsuit_0$ and $\heartsuit_F$ as the temporal operators X (next-time) and G (any-time), respectively, which are subsumed in *linear-time temporal logic* (LTL) (Emerson, 1990; Pnueli, 1977). The corresponding axiom scheme for the singleton case just represents the following axiom scheme for LTL: $\mathrm{G}\alpha \leftrightarrow \bigwedge\{\mathrm{X}^i\alpha \mid i \in \omega\}$ where $\mathrm{X}^i\alpha$ is defined inductively by $\mathrm{X}^0\alpha := \alpha$ and $\mathrm{X}^{i+1}\alpha := \mathrm{X}\mathrm{X}^i\alpha$. The fixpoint operator $\heartsuit_F$ is thus regarded as a natural generalization of both the any-time temporal operator and the common knowledge operator.

Present paper's results. The results of this paper are then summarized as follows. Two multi-agent versions MILL and MCLL of ILL and CLL, respectively, are introduced as Gentzen-type sequent calculi. MILL and MCLL have the fixpoint operator $\heartsuit_F$, which is naturally formalized based on the idea of iterative interpretation of common knowledge. The completeness theorems with respect to *modality-indexed phase semantics* for MILL and MCLL are proved by using Okada's phase semantic method. The cut-elimination theorems for MILL and MCLL are then simultaneously obtained by this method. Some related works are briefly surveyed.

2. Intuitionistic case

2.1 Sequent calculus

The language used in this section is introduced below. Let n be a fixed positive integer. Then, the symbol N is used to represent the set $\{1, 2, ..., n\}$ of indexes of modal operators. *Formulas* are constructed from propositional variables, $\mathbf{1}$ (multiplicative constant), $\top$, $\mathbf{0}$ (additive constants), $\to$ (implication), $\wedge$ (conjunction), $*$ (fusion), $\vee$ (disjunction), ! (of course), $\heartsuit_i$ ($i \in N$) (i-th modality) and $\heartsuit_{\mathrm{F}}$ (fixpoint modality). Remark that the symbols $\to, \wedge, *$ and $\vee$ are from (Troelstra, 1992), which are different from those in (Girard, 1987). Lower-case letters $p, q, ...$ are used to represent propositional variables, Greek lower-case letters $\alpha, \beta, ...$ are used to represent formulas, and Greek capital letters $\Gamma, \Delta, ...$ are used to represent finite (possibly empty) multisets of formulas. For any $\sharp \in \{!, \heartsuit_i, \heartsuit_{\mathrm{F}}\}$, an expression $\sharp\Gamma$ is used to denote the multiset $\{\sharp\gamma \mid \gamma \in \Gamma\}$. We write $A \equiv B$ to indicate the syntactical identity between A and B. An expression Γ^* means $\Gamma^* \equiv \gamma_1 * \cdots * \gamma_n$ if $\Gamma \equiv \{\gamma_1, \cdots, \gamma_n\}$ $(0 < n)$ and $\Gamma^* \equiv \mathbf{1}$ if $\Gamma \equiv \emptyset$. The symbol ω is used to represent the set of natural numbers. The symbol K is used to represent the set $\{\heartsuit_i \mid i \in N\}$, and the symbol K^* is used to represent the set of all words of finite length of the alphabet K. For example, $\{\iota\alpha \mid \iota \in K^*\}$ denotes the set $\{\heartsuit_{i_1} \cdots \heartsuit_{i_k}\alpha \mid i_1, ..., i_k \in N, k \in \omega\}$. Remark that K^* includes $\emptyset$ and hence $\{\iota\alpha \mid \iota \in K^*\}$ includes α. Greek lower-case letters $\iota, \iota_1, ..., \iota_n$ and κ are used to denote any members of K^*. A *two-sided intuitionistic sequent*, simply called a sequent, is an expression of the form $\Gamma \Rightarrow \gamma$ (the succedent of the sequent is not empty). It is assumed that the terminological conventions regarding sequents (e.g., antecedent, succedent etc.) are the usual ones. If a sequent S is provable in a sequent system L, then such a fact is denoted as $L \vdash S$ or $\vdash S$. The parentheses for $*$ is omitted since $*$ is associative, i.e., $\vdash \alpha * (\beta * \gamma) \Rightarrow (\alpha * \beta) * \gamma$ and $\vdash (\alpha * \beta) * \gamma \Rightarrow \alpha * (\beta * \gamma)$ for any formulas α, β and γ. A rule R of inference is said to be *admissible* in a sequent calculus L if the following condition is satisfied: for any instance

$$\frac{S_1 \cdots S_n}{S}$$

of R, if $L \vdash S_i$ for all i, then $L \vdash S$.

An *intuitionistic linear logic with fixpoint operator*, MILL, is introduced below.

Definition 2.1. *The initial sequents of* MILL *are of the form:*

$$\alpha \Rightarrow \alpha \qquad \Rightarrow \iota\mathbf{1} \qquad \Gamma \Rightarrow \iota\top \qquad \iota\mathbf{0}, \Gamma \Rightarrow \gamma.$$

The cut rule of MILL *is of the form:*

$$\frac{\Gamma \Rightarrow \alpha \quad \alpha, \Delta \Rightarrow \gamma}{\Gamma, \Delta \Rightarrow \gamma} \text{ (cut)}.$$

The logical inference rules of MILL *are of the form:*

$$\frac{\Gamma \Rightarrow \gamma}{\iota\mathbf{1}, \Gamma \Rightarrow \gamma} \text{ (}\mathbf{1}\text{we)}$$

$$\frac{\Gamma \Rightarrow \iota\alpha \quad \iota\beta, \Delta \Rightarrow \gamma}{\iota(\alpha \to \beta), \Gamma, \Delta \Rightarrow \gamma}\ (\to\text{left}) \quad \frac{\iota\alpha, \Gamma \Rightarrow \iota\beta}{\Gamma \Rightarrow \iota(\alpha \to \beta)}\ (\to\text{right})$$

$$\frac{\iota\alpha, \Gamma \Rightarrow \gamma}{\iota(\alpha \wedge \beta), \Gamma \Rightarrow \gamma}\ (\wedge\text{left1}) \quad \frac{\iota\beta, \Gamma \Rightarrow \gamma}{\iota(\alpha \wedge \beta), \Gamma \Rightarrow \gamma}\ (\wedge\text{left2}) \quad \frac{\Gamma \Rightarrow \iota\alpha \quad \Gamma \Rightarrow \iota\beta}{\Gamma \Rightarrow \iota(\alpha \wedge \beta)}\ (\wedge\text{right})$$

$$\frac{\iota\alpha, \iota\beta, \Gamma \Rightarrow \gamma}{\iota(\alpha * \beta), \Gamma \Rightarrow \gamma}\ (*\text{left}) \quad \frac{\Gamma \Rightarrow \iota\alpha \quad \Delta \Rightarrow \iota\beta}{\Gamma, \Delta \Rightarrow \iota(\alpha * \beta)}\ (*\text{right})$$

$$\frac{\iota\alpha, \Gamma \Rightarrow \gamma \quad \iota\beta, \Gamma \Rightarrow \gamma}{\iota(\alpha \vee \beta), \Gamma \Rightarrow \gamma}\ (\vee\text{left}) \quad \frac{\Gamma \Rightarrow \iota\alpha}{\Gamma \Rightarrow \iota(\alpha \vee \beta)}\ (\vee\text{right1}) \quad \frac{\Gamma \Rightarrow \iota\beta}{\Gamma \Rightarrow \iota(\alpha \vee \beta)}\ (\vee\text{right2})$$

$$\frac{\iota\alpha, \Gamma \Rightarrow \gamma}{\iota!\alpha, \Gamma \Rightarrow \gamma}\ (!\text{left}) \quad \frac{\iota_1!\gamma_1, ..., \iota_n!\gamma_n \Rightarrow \kappa\alpha}{\iota_1!\gamma_1, ..., \iota_n!\gamma_n \Rightarrow \kappa!\alpha}\ (!\text{right})$$

$$\frac{\iota!\alpha, \iota!\alpha, \Gamma \Rightarrow \gamma}{\iota!\alpha, \Gamma \Rightarrow \gamma}\ (!\text{co}) \quad \frac{\Gamma \Rightarrow \gamma}{\iota!\alpha, \Gamma \Rightarrow \gamma}\ (!\text{we})$$

$$\frac{\iota\kappa\alpha, \Gamma \Rightarrow \gamma}{\iota\heartsuit_{\mathrm{F}}\alpha, \Gamma \Rightarrow \gamma}\ (\heartsuit_{\mathrm{F}}\text{left}) \quad \frac{\{\ \Gamma \Rightarrow \iota\kappa\alpha\ \}_{\kappa \in K^*}}{\Gamma \Rightarrow \iota\heartsuit_{\mathrm{F}}\alpha}\ (\heartsuit_{\mathrm{F}}\text{right}).$$

Remark that ($\heartsuit_{\mathrm{F}}$right) has infinite premises, and that the cases for $\iota = \emptyset$ in MILL derive the usual inference rules for the intuitionistic linear logic. The rules ($\heartsuit_{\mathrm{F}}$left) and ($\heartsuit_{\mathrm{F}}$right) are intended to formalize an informal axiom scheme: $\heartsuit_{\mathrm{F}}\alpha \leftrightarrow \bigwedge\{\iota\alpha \mid \iota \in K^*\}$.
The following proposition is needed in the completeness proof.

Proposition 2.2. *The following rules are admissible in cut-free* MILL*: for any $i \in N$,*

$$\frac{\Gamma \Rightarrow \gamma}{\heartsuit_i\Gamma \Rightarrow \heartsuit_i\gamma}\ (\heartsuit_i\text{regu}) \quad \frac{\Gamma \Rightarrow \iota\heartsuit_{\mathrm{F}}\alpha}{\Gamma \Rightarrow \iota\kappa\alpha}\ (\heartsuit_{\mathrm{F}}\text{right}^{-1})$$

$$\frac{\Gamma \Rightarrow \iota(\alpha \to \beta)}{\Gamma, \iota\alpha \Rightarrow \iota\beta}\ (\to\text{right}^{-1}) \quad \frac{\iota(\alpha * \beta), \Gamma \Rightarrow \gamma}{\iota\alpha, \iota\beta, \Gamma \Rightarrow \gamma}\ (*\text{left}^{-1}).$$

An expression $\alpha \Leftrightarrow \beta$ is an abbreviation of $\alpha \Rightarrow \beta$ and $\beta \Rightarrow \alpha$.

Proposition 2.3. *The following sequents are provable in cut-free* MILL*: for any formulas α, β and any $i \in N$,*

1. $\heartsuit_i(\alpha \circ \beta) \Leftrightarrow \heartsuit_i\alpha \circ \heartsuit_i\beta$ *where* $\circ \in \{\to, \wedge, *, \vee\}$,
2. $\heartsuit_i!\alpha \Leftrightarrow !\heartsuit_i\alpha$,
3. $\heartsuit_{\mathrm{F}}\alpha \Rightarrow \heartsuit_i\alpha$,
4. $\heartsuit_{\mathrm{F}}\alpha \Rightarrow \alpha$,
5. $\heartsuit_{\mathrm{F}}\alpha \Rightarrow \heartsuit_i\heartsuit_{\mathrm{F}}\alpha$,
6. $\heartsuit_{\mathrm{F}}\alpha \Rightarrow \heartsuit_{\mathrm{F}}\heartsuit_{\mathrm{F}}\alpha$.

Note that a proof of MILL provides both an infinite width and an unbounded depth. Such a fact implies that obtaining a direct cut-elimination proof for MILL may be very difficult. Thus, to prove the cut-elimination theorem effectively, we need the phase semantic cut-elimination method proposed by Okada.

2.2 Phase semantics

We now define a phase semantics for MILL. The difference between such a semantics and the original semantics for the intuitionistic linear logic is only the definition of the valuations: whereas the original semantics for the intuitionistic linear logic has a valuation v, our semantics has an infinite number of modality-indexed valuations v^{ι} ($\iota \in K^*$), where $v^{\emptyset}$ is the same as v.

Definition 2.4. *An* intuitionistic phase space *is a structure* $\langle \mathbf{M}, cl, \mathrm{I}\rangle$ *satisfying the following conditions:*

1. $\mathbf{M} := \langle M, \cdot, 1\rangle$ *is a commutative monoid with the identity* 1,
2. cl *is an operation on the powerset* $P(M)$ *of* M *such that, for any* $X, Y \in P(M)$,
   C1: $X \subseteq cl(X)$,
   C2: $clcl(X) \subseteq cl(X)$,
   C3: $X \subseteq Y$ *implies* $cl(X) \subseteq cl(Y)$,
   C4: $cl(X) \circ cl(Y) \subseteq cl(X \circ Y)$
   where the operation $\circ$ *is defined as* $X \circ Y := \{x \cdot y \mid x \in X \text{ and } y \in Y\}$ *for any* $X, Y \in P(M)$ *(the operation* cl *is called here* closure operation*),*
3. I *is a submonoid of* M *such that* $cl\{x\} \subseteq cl\{x \cdot x\}$ *for any* $x \in \mathrm{I}$.

In order to obtain an interpretation of the logical constants and connectives, the corresponding constants and operations on $P(M)$ are defined below.

Definition 2.5. *Constants and operations on* $P(M)$ *are defined as follows: for any* $X, Y \in P(M)$,

1. $\dot{\mathbf{1}} := cl\{1\}$,
2. $\dot{\top} := M$,
3. $\dot{\mathbf{0}} := cl(\emptyset)$,
4. $X \dot{\rightarrow} Y := \{y \mid \forall x \in X\ (x \cdot y \in Y)\}$,
5. $X \dot{\wedge} Y := X \cap Y$,
6. $X \dot{\vee} Y := cl(X \cup Y)$,
7. $X \dot{*} Y := cl(X \circ Y)$,
8. $\dot{!}\, X := cl(X \cap \mathrm{I} \cap \dot{\mathbf{1}})$.

We define $D := \{X \in P(M) \mid X = cl(X)\}$. *Then,*

$\mathbf{D} := \langle D, \dot{\rightarrow}, \dot{*}, \dot{\wedge}, \dot{\vee}, \dot{!}, \dot{\mathbf{1}}, \dot{\top}, \dot{\mathbf{0}}\rangle$

is called an intuitionistic phase structure*.*

Remark that the following hold: for any $X, X', Y, Y', Z \in P(M)$,

1. $X \subseteq Y \dot{\rightarrow} Z$ iff $X \circ Y \subseteq Z$,
2. $X \subseteq X'$ and $Y \subseteq Y'$ imply $X \circ Y \subseteq X' \circ Y'$ and $X' \dot{\rightarrow} Y \subseteq X \dot{\rightarrow} Y'$.

Remark that D is closed under the operations $\dot{\rightarrow}, \dot{*}, \dot{\wedge}, \dot{\vee}, \dot{!}$ and $\bigcap$ (infinite meet), and that $\dot{\mathbf{1}}, \dot{\top}, \dot{\mathbf{0}} \in D$.

Definition 2.6. Modality-indexed valuations v^{ι} *for all* $\iota \in K^*$ *on an intuitionistic phase structure* $\mathbf{D} := \langle D, \dot{\rightarrow}, \dot{*}, \dot{\wedge}, \dot{\vee}, \dot{!}, \dot{\mathbf{1}}, \dot{\top}, \dot{\mathbf{0}}\rangle$ *are mappings from the set of all propositional variables to* D*. Then,* v^{ι} *for all* $\iota \in K^*$ *are extended to mappings from the set* Φ *of all formulas to* D *by:*

1. $v^\iota(\mathbf{1}) := \dot{\mathbf{1}}$,
2. $v^\iota(\top) := \dot{\top}$,
3. $v^\iota(\mathbf{0}) := \dot{\mathbf{0}}$,
4. $v^\iota(\alpha \wedge \beta) := v^\iota(\alpha) \dot{\wedge} v^\iota(\beta)$,
5. $v^\iota(\alpha \vee \beta) := v^\iota(\alpha) \dot{\vee} v^\iota(\beta)$,
6. $v^\iota(\alpha * \beta) := v^\iota(\alpha) \dot{*} v^\iota(\beta)$,
7. $v^\iota(\alpha \rightarrow \beta) := v^\iota(\alpha) \dot{\rightarrow} v^\iota(\beta)$,
8. $v^\iota(!\alpha) := \dot{!} v^\iota(\alpha)$,
9. $v^\iota(\heartsuit_i \alpha) := v^{\iota \heartsuit_i}(\alpha)$,
10. $v^\iota(\heartsuit_{\mathrm{F}} \alpha) := \bigcap_{\kappa \in K^*} v^{\iota\kappa}(\alpha)$.

Definition 2.7. *An* intuitionistic modality-indexed phase model *is a structure* $\langle \mathbf{D}, \{v^\iota\}_{\iota \in K^*} \rangle$ *such that* $\mathbf{D}$ *is an intuitionistic phase structure, and* $\{v^\iota\}_{\iota \in K^*}$ *is a class of modality-indexed valuations. A formula* α *is* true *in an intuitionistic modality-indexed phase model* $\langle \mathbf{D}, \{v^\iota\}_{\iota \in K^*} \rangle$ *if* $\dot{\mathbf{1}} \subseteq v^{\emptyset}(\alpha)$ *(or equivalently* $1 \in v^{\emptyset}(\alpha)$*) holds, and* valid *in an intuitionistic phase structure* $\mathbf{D}$ *if it is true for any modality-indexed valuations* $\{v^\iota\}_{\iota \in K^*}$ *on the intuitionistic phase structure. A sequent* $\alpha_1, \cdots, \alpha_n \Rightarrow \beta$ *(or* $\Rightarrow \beta$*) is* true *in an intuitionistic modality-indexed phase model* $\langle \mathbf{D}, \{v^\iota\}_{\iota \in K^*} \rangle$ *if the formula* $\alpha_1 * \cdots * \alpha_n \rightarrow \beta$ *(or* β*) is true in it, and valid in an intuitionistic phase structure if so is* $\alpha_1 * \cdots * \alpha_n \rightarrow \beta$ *(or* β*).*

The proof of the following theorem is straightforward.

Theorem 2.8 (Soundness). *If a sequent S is provable in* MILL*, then S is valid for any intuitionistic phase structures.*

2.3 Completeness and cut-elimination

In order to prove the strong completeness theorem, we will construct a canonical model. For the sake of clarity for the completeness proof, an expression $[\Gamma]$ is used to explicitly represent a multiset of formulas, i.e., $[\Gamma]$ and Γ are identical, but only the expressions are different.

Definition 2.9. *We define a commutative monoid* $\langle M, \cdot, 1 \rangle$ *as follows:*

1. $M := \{[\Gamma] \mid [\Gamma]$ *is a finite multiset of formulas*$\}$,
2. $[\Gamma] \cdot [\Delta] := [\Gamma, \Delta]$ *(the multiset union),*
3. $1 := [\ ]$ *(the empty multiset).*

We define the following: for any $\iota \in K^*$ *and any formula* α,

$$\|\alpha\|^\iota := \{[\Gamma] \mid \vdash_{cf} \Gamma \Rightarrow \iota\alpha\}$$

where $\vdash_{cf}$ *means "provable in cut-free* MILL*".*
We then define

$$D := \{X \mid X = \bigcap_{i \in I} \|\alpha_i\|^{\emptyset}\}$$

for an arbitrary (non-empty) indexing set I and an arbitrary formula α_i.
Then we define

$$cl(X) := \bigcap\{Y \in D \mid X \subseteq Y\}.$$

We define the following constants and operations on $P(M)$: for any $X, Y \in P(M)$,

1. $\dot{\mathbf{1}} := cl\{1\}$,
2. $\dot{\top} := M$,
3. $\dot{\mathbf{0}} := cl(\emptyset)$,
4. $X \dot{\rightarrow} Y := \{[\Delta] \mid \forall\, [\Gamma] \in X\ ([\Gamma, \Delta] \in Y)\}$,
5. $X \dot{\wedge} Y := X \cap Y$,
6. $X \dot{\vee} Y := cl(X \cup Y)$,
7. $X \dot{*} Y := cl(X \circ Y)$ *where* $X \circ Y := \{[\Gamma, \Delta] \mid [\Gamma] \in X$ *and* $[\Delta] \in Y\}$,
8. $\dot{!}X := cl(X \cap \mathrm{I} \cap \mathbf{1})$ *where*
   $\mathrm{I} := \{[\iota_1!\gamma_1, ..., \iota_n!\gamma_n] \mid \iota_1, ..., \iota_n \in K^*$ *and* $\gamma_1, ..., \gamma_n$ *: formulas*$\}$.

Modality-indexed valuations v^ι for all $\iota \in K^*$ are mappings from the set of all propositional variables to D such that

$v^\iota(p) := \|p\|^\iota$.

We have the following: for any $X, Y, Z \in P(M)$,

$X \circ Y \subseteq Z$ iff $X \subseteq Y \dot{\rightarrow} Z$.

Remark that D is closed under arbitrary $\bigcap$. Remark also that I is a monoid.
Moreover, we have to check the following fact.

Proposition 2.10. *For any $[\Sigma] \in \mathrm{I}$, $cl\{[\Sigma]\} \subseteq cl\{[\Sigma, \Sigma]\}$.*

Proof. Let $[\Sigma] \in \mathrm{I}$. Suppose $[\Sigma] \in cl\{[\Sigma]\}$, i.e., $[\Sigma] \in \bigcap\{Y \in D \mid \{[\Sigma]\} \subseteq Y\}$ iff $\forall W\ [W \in D$ and $\{[\Sigma]\} \subseteq W$ imply $[\Delta] \in W]$. We show $[\Delta] \in cl\{[\Sigma, \Sigma]\}$, i.e., $\forall W\ [W \in D$ and $\{[\Sigma, \Sigma]\} \subseteq W$ imply $[\Delta] \in W]$. To show this, suppose $W \in D$ and $\{[\Sigma, \Sigma] \subseteq W\}$, i.e., $\{[\Sigma, \Sigma]\} \subseteq W = \bigcap_{i \in I} \|\alpha_i\|^0 = \{\Pi \mid \forall i \in I\ (\vdash_{cf} \Pi \Rightarrow \alpha_i)\}$. This means $\forall i \in I\ (\vdash_{cf} \Sigma, \Sigma \Rightarrow \alpha_i)$. Moreover, $[\Sigma]$ is of the form $[\iota_1!\gamma_1, ..., \iota_k!\gamma_k]$ since $[\Sigma] \in \mathrm{I}$. Thus, we have:

$\forall i \in I\ (\vdash_{cf} \iota_1!\gamma_1, ..., \iota_k!\gamma_k, \iota_1!\gamma_1, ..., \iota_k!\gamma_k \Rightarrow \alpha_i)$,

and hence obtain:

$\forall i \in I\ (\vdash_{cf} \iota_1!\gamma_1, ..., \iota_k!\gamma_k \Rightarrow \alpha_i)$

by (!co). This means $\{[\iota_1!\gamma_1, ..., \iota_k!\gamma_k]\} \subseteq \bigcap_{i \in I} \|\alpha_i\|^0 = W$, i.e., $\{[\Sigma]\} \subseteq W$. Therefore we obtain $[\Delta] \in W$ by the hypothesis. ∎

Proposition 2.11. *The following hold: for any $\iota \in K^*$ and any formula α,*

1. $\|\heartsuit_i \alpha\|^\iota = \|\alpha\|^{\iota \heartsuit_i}$,
2. $\|\heartsuit_{\mathrm{F}} \alpha\|^\iota = \bigcap_{\kappa \in K^*} \|\alpha\|^{\iota\kappa}$.

Proof. (1) is obvious. (2) can be shown using the rules ($\heartsuit_{\mathrm{F}}$right) and ($\heartsuit_{\mathrm{F}}$right^{-1}), where ($\heartsuit_{\mathrm{F}}$right^{-1}) is admissible in cut-free MILL by Proposition 2.2. ∎

Lemma 2.12. *Let D be $\{X \mid X = \bigcap_{i \in I} \|\alpha_i\|^{\emptyset}\}$, and D_c be $\{X \in P(M) \mid X = cl(X)\}$. Then: $D = D_c$.*

Proof. First, we show $D_c \subseteq D$. Suppose $X \in D_c$. Then $X = cl(X) = \bigcap\{Y \in D \mid X \subseteq Y\} \in D$. Next, we show $D \subseteq D_c$. Suppose $X \in D$. We show $X \in D_c$, i.e., $X = \bigcap\{Y \in D \mid X \subseteq Y\}$. To show this, it is sufficient to prove that

1. $X \subseteq \{[\Gamma] \mid \forall\, W\, [W \in D \text{ and } X \subseteq W \text{ imply } [\Gamma] \in W]\}$,
2. $\{[\Gamma] \mid \forall\, W\, [W \in D \text{ and } X \subseteq W \text{ imply } [\Gamma] \in W]\} \subseteq X$.

First, we show (1). Suppose $[\Delta] \in X$ and assume $W \in D$ and $X \subseteq W$ for any W. Then we have $[\Delta] \in X \subseteq W$. Next we show (2). Suppose $[\Delta] \in \{[\Gamma] \mid \forall\, W\, [W \in D \text{ and } X \subseteq W \text{ imply } [\Gamma] \in W]\}$. By the assumption $X \in D$ and the fact that $X \subseteq X$, we have $[\Delta] \in X$. ∎

Lemma 2.13. *For any $X \subseteq M$ and any $Y \in D$, we have $X \dot{\rightarrow} Y \in D$.*

Proof. By using Proposition 2.2 for the admissibility of ($\rightarrow$right^{-1}) and ($*$left^{-1}) in cut-free MILL. ∎

Then, we can show the following.

Proposition 2.14. *The structure $\mathbf{D} := \langle D, \dot{\rightarrow}, \dot{*}, \dot{\wedge}, \dot{\vee}, \dot{!}, \dot{\mathbf{1}}, \dot{\top}, \dot{\mathbf{0}} \rangle$ defined in Definition 2.9 forms an intuitionistic phase structure for* MILL.

Proof. We can verify that D is closed under $\dot{\rightarrow}, \dot{*}, \dot{\wedge}, \dot{\vee}, \dot{!}$ and $\bigcap$. In particular, for $\dot{\rightarrow}$, we use Lemma 2.13. The fact $\dot{\mathbf{1}}, \dot{\top}, \dot{\mathbf{0}} \in D$ is obvious. We can verify that the conditions C1—C4 for closure operation hold for this structure. The conditions C1—C3 are obvious. We only show C4: $cl(X) \circ cl(Y) \subseteq cl(X \circ Y)$ for any $X, Y \in P(M)$. We have $X \circ Y \subseteq cl(X \circ Y)$ by the condition C1, and hence $X \subseteq Y \dot{\rightarrow} cl(X \circ Y)$. Moreover, by the condition C3, we have $cl(X) \subseteq cl(Y \dot{\rightarrow} cl(X \circ Y))$. Here, $cl(X \circ Y) \in D$ and by Lemma 2.13, we have $Y \dot{\rightarrow} cl(X \circ Y) \in D$. Thus, we obtain

$$cl(X) \subseteq cl(Y \dot{\rightarrow} cl(X \circ Y)) = Y \dot{\rightarrow} cl(X \circ Y)$$

by Lemma 2.12. Therefore we obtain (*): $cl(X) \circ Y \subseteq cl(X \circ Y)$ for any $X, Y \in P(M)$. By applying the fact (*) twice, Lemma 2.12 and the commutativity of $\circ$, we have

$$cl(X) \circ cl(Y) \subseteq cl(cl(X) \circ Y) \subseteq cl(cl(X \circ Y)) = cl(X \circ Y).$$

∎

We then have a modified version of the key lemma of Okada (Okada, 2002).

Lemma 2.15. *For any $\iota \in K^*$ and any formula α,*

$$[\iota\alpha] \in v^{\iota}(\alpha) \subseteq \|\alpha\|^{\iota}.$$

Proof. By induction on the complexity of α. We show only some critical cases.
(Case $\alpha \equiv \heartsuit_i\beta$): By induction hypothesis, we have $[\iota\heartsuit_i\beta] \in v^{\iota\heartsuit_i}(\beta) \subseteq \|\beta\|^{\iota\heartsuit_i}$, i.e., $[\iota(\heartsuit_i\beta)] \in v^{\iota}(\heartsuit_i\beta) \subseteq \|\heartsuit_i\beta\|^{\iota}$ by Proposition 2.11.
(Case $\alpha \equiv \heartsuit_{\mathrm{F}}\beta$): We show $[\iota\heartsuit_{\mathrm{F}}\beta] \in v^{\iota}(\heartsuit_{\mathrm{F}}\beta) \subseteq \|\heartsuit_{\mathrm{F}}\beta\|^{\iota}$. First, we show $[\iota\heartsuit_{\mathrm{F}}\beta] \in v^{\iota}(\heartsuit_{\mathrm{F}}\beta)$, i.e., $[\iota\heartsuit_{\mathrm{F}}\beta] \in \bigcap_{\kappa \in K^*} v^{\iota\kappa}(\beta)$ iff $\forall\kappa \in K^*\ ([\iota\heartsuit_{\mathrm{F}}\beta] \in v^{\iota\kappa}(\beta))$. Since $v^{\iota\kappa}(\beta) \in D$, we have $v^{\iota\kappa}(\beta) = \bigcap_{k \in I} \|\delta_k\|^{\emptyset} = \{[\Gamma] \mid \forall k \in I(\vdash_{cf} \Gamma \Rightarrow \delta_k)\}$. Thus, $\forall\kappa \in K^*\ ([\iota\heartsuit_{\mathrm{F}}\beta] \in v^{\iota\kappa}(\beta))$ means (*): $\forall k \in I(\vdash_{cf} \iota\heartsuit_{\mathrm{F}}\beta \Rightarrow \delta_k)$. On the other hand, by induction hypothesis, we have $\forall\kappa \in K^*\ ([\iota\kappa\beta] \in v^{\iota\kappa}(\beta))$, i.e., (**): $\forall k \in I\ \forall\kappa \in K^*\ (\vdash_{cf} \iota\kappa\beta \Rightarrow \delta_k)$. By applying ($\heartsuit_{\mathrm{F}}$left) to (**), we obtain (*).

Next we show $v^{\iota}(\heartsuit_{\mathrm{F}}\beta) \subseteq \|\heartsuit_{\mathrm{F}}\beta\|^{\iota}$. Suppose $[\Gamma] \in v^{\iota}(\heartsuit_{\mathrm{F}}\beta)$, i.e., $[\Gamma] \in \bigcap_{\kappa \in K^*} v^{\iota\kappa}(\beta)$. We show $[\Gamma] \in \|\heartsuit_{\mathrm{F}}\beta\|^{\iota}$, i.e., $\vdash_{cf} \Gamma \Rightarrow \iota\heartsuit_{\mathrm{F}}\beta$. By induction hypothesis, we have $v^{\iota\kappa}(\beta) \subseteq \|\beta\|^{\iota\kappa}$. Thus, we obtain $[\Gamma] \in \bigcap_{\kappa \in K^*} v^{\iota\kappa}(\beta) \subseteq \bigcap_{\kappa \in K^*} \|\beta\|^{\iota\kappa}$, and hence $[\Gamma] \in \bigcap_{\kappa \in K^*} \|\beta\|^{\iota\kappa}$, i.e., $\forall \kappa \in K^*$ $([\Gamma] \in \|\beta\|^{\iota\kappa})$ iff $\forall \kappa \in K^*$ $(\vdash_{cf} \Gamma \Rightarrow \iota\kappa\beta)$. By applying ($\heartsuit_{\mathrm{F}}$right) to this, we obtain $\vdash_{cf} \Gamma \Rightarrow \iota\heartsuit_{\mathrm{F}}\beta$. ∎

Theorem 2.16 (Strong completeness)**.** *If a sequent S is valid for any intuitionistic phase structures, then S is provable in cut-free* MILL.

Proof. Using Lemma 2.15, we can obtain this theorem as follows. Let $\Gamma \Rightarrow \gamma$ be S, and α be $\Gamma^* \rightarrow \gamma$. If formula α is true, then $[\,] \in v^{\emptyset}(\alpha)$. On the other hand $v^{\iota}(\alpha) \subseteq \|\alpha\|^{\iota}$ for any $\iota \in K^*$, and hence $[\,] \in \|\alpha\|^{\emptyset}$, which means "$\Rightarrow \alpha$ is provable in cut-free MILL". ∎

Theorem 2.17 (Cut-elimination)**.** *The rule* (cut) *is admissible in cut-free* MILL.

Proof. If a sequent S is provable in MILL, then S is valid by Theorem 2.8 (Soundness). By Theorem 2.16 (Strong completeness), S is provable in cut-free MILL. ∎

3. Classical case

3.1 Sequent calculus

The language used in this section is introduced below. *Formulas* are constructed from propositional variables, $\mathbf{1}, \perp$ (multiplicative constants), $\mathbf{0}$, $\top$ (additive constants *), $\wedge$ (conjunction), $*$ (fusion), $\vee$ (disjunction), $+$ (fission), $\cdot^{\perp}$ (negation), ! (of course), ? (why not), $\heartsuit_i$ $(i \in N)$ (i-th modality), $\heartsuit_{\mathrm{F}}$ (fixpoint modality) and $\heartsuit_{\mathrm{D}}$ (co-fixpoint modality). The notational conventions are almost the same as that in the previous section. For example, for any $\sharp \in \{!, ?, \heartsuit_i, \heartsuit_{\mathrm{F}}, \heartsuit_{\mathrm{D}}\}$, an expression $\sharp\Gamma$ is used to denote the multiset $\{\sharp\gamma \mid \gamma \in \Gamma\}$. A *classical one-sided sequent*, simply a sequent, is an expression of the form $\vdash \Gamma$. An expression $\alpha \leftrightarrow \beta$ is used to represent the fact that both $\vdash \alpha^{\perp}, \beta$ and $\vdash \alpha, \beta^{\perp}$ are provable. In the one-sided calculi discussed here, the De Morgan duality is assumed, i.e., the following laws and the replacement (or substitution) theorem are assumed: $\mathbf{1}^{\perp} \leftrightarrow \perp$, $\perp^{\perp} \leftrightarrow \mathbf{1}$, $\top^{\perp} \leftrightarrow \mathbf{0}$, $\mathbf{0}^{\perp} \leftrightarrow \top$, $\alpha^{\perp\perp} \leftrightarrow \alpha$, $(\alpha \wedge \beta)^{\perp} \leftrightarrow \alpha^{\perp} \vee \beta^{\perp}$, $(\alpha \vee \beta)^{\perp} \leftrightarrow \alpha^{\perp} \wedge \beta^{\perp}$, $(\alpha * \beta)^{\perp} \leftrightarrow \alpha^{\perp} + \beta^{\perp}$, $(\alpha + \beta)^{\perp} \leftrightarrow \alpha^{\perp} * \beta^{\perp}$, $(!\alpha)^{\perp} \leftrightarrow ?(\alpha^{\perp})$, $(?\alpha)^{\perp} \leftrightarrow !(\alpha^{\perp})$, $(\heartsuit_i\alpha)^{\perp} \leftrightarrow \heartsuit_i(\alpha^{\perp})$, $(\heartsuit_{\mathrm{F}}\alpha)^{\perp} \leftrightarrow \heartsuit_{\mathrm{D}}(\alpha^{\perp})$ and $(\heartsuit_{\mathrm{D}}\alpha)^{\perp} \leftrightarrow \heartsuit_{\mathrm{F}}(\alpha^{\perp})$.
A *classical linear logic with fixpoint operator*, MCLL, is introduced below.

Definition 3.1. *The initial sequents of* MCLL *are of the form:*

$$\vdash \alpha, \alpha^{\perp} \qquad \vdash \iota\mathbf{1} \qquad \vdash \Gamma, \iota\top.$$

The cut rule of MCLL *is of the form:*

$$\frac{\vdash \Gamma, \iota\alpha \quad \vdash \Delta, \iota(\alpha^{\top})}{\vdash \Gamma, \Delta}\ (\text{cut}).$$

*Although $\mathbf{0}$ is not appeared explicitly in the one-sided calculi discussed in this paper, it is used as an abbreviation of $\top^{\perp}$.

The logical inference rules of MCLL *are of the form:*

$$\frac{\vdash \Gamma}{\vdash \Gamma, \iota\bot}\ (\bot)\quad \frac{\vdash \Gamma, \iota\alpha \quad \vdash \Delta, \iota\beta}{\vdash \Gamma, \Delta, \iota(\alpha * \beta)}\ (*)\quad \frac{\vdash \Gamma, \iota\alpha, \iota\beta}{\vdash \Gamma, \iota(\alpha + \beta)}\ (+)$$

$$\frac{\vdash \Gamma, \iota\alpha \quad \vdash \Gamma, \iota\beta}{\vdash \Gamma, \iota(\alpha \wedge \beta)}\ (\wedge)\quad \frac{\vdash \Gamma, \iota\alpha}{\vdash \Gamma, \iota(\alpha \vee \beta)}\ (\vee 1)\quad \frac{\vdash \Gamma, \iota\beta}{\vdash \Gamma, \iota(\alpha \vee \beta)}\ (\vee 2)$$

$$\frac{\vdash \iota_1?\gamma_1, ..., \iota_n?\gamma_n, \kappa\alpha}{\vdash \iota_1?\gamma_1, ..., \iota_n?\gamma_n, \kappa!\alpha}\ (!)\quad \frac{\vdash \Gamma, \iota\alpha}{\vdash \Gamma, \iota?\alpha}\ (?)$$

$$\frac{\vdash \Gamma, \iota?\alpha, \iota?\alpha}{\vdash \Gamma, \iota?\alpha}\ (?\mathrm{co})\quad \frac{\vdash \Gamma}{\vdash \Gamma, \iota?\alpha}\ (?\mathrm{we})$$

$$\frac{\{\ \vdash \Gamma, \iota\kappa\alpha\ \}_{\kappa \in K^*}}{\vdash \Gamma, \iota\heartsuit_{\mathrm{F}}\alpha}\ (\heartsuit_{\mathrm{F}})\quad \frac{\vdash \Gamma, \iota\kappa\alpha}{\vdash \Gamma, \iota\heartsuit_{\mathrm{D}}\alpha}\ (\heartsuit_{\mathrm{D}}).$$

Note that the following conditions hold for MCLL: for any $i \in N$ and any formulas α and β, $\heartsuit_i(\alpha \circ \beta) \leftrightarrow (\heartsuit_i\alpha) \circ (\heartsuit_i\beta)$ where $\circ \in \{\wedge, \vee, *, +\}$, $(\heartsuit_i\alpha)^{\bot} \leftrightarrow \heartsuit_i(\alpha^{\bot})$ and $\heartsuit_i(\sharp\alpha) \leftrightarrow \sharp(\heartsuit_i\alpha)$ where $\sharp \in \{!, ?\}$.

3.2 Phase semantics

We now define a phase semantics for MCLL. The difference between such semantics and the original semantics is only the definition of the valuations.

Definition 3.2. *Let $\langle M, \cdot, 1\rangle$ be a commutative monoid with the unit 1. If $X, Y \subseteq M$, we define $X \circ Y := \{x \cdot y \mid x \in X \text{ and } y \in Y\}$. A* phase spase *is a structure $\langle M, \hat{\bot}, \hat{I}\rangle$ where $\hat{\bot}$ is a fixed subset of M, and $\hat{I} := \{x \in M \mid x \cdot x = x\} \cap \hat{\bot}^{\hat{\bot}}$. For $X \subseteq M$, we define $X^{\hat{\bot}} := \{y \mid \forall x \in X\ (x \cdot y \in \hat{\bot})\}$. $X\ (\subseteq M)$ is called a* fact *if $X^{\hat{\bot}\hat{\bot}} = X$. The set of facts is denoted by D_M.*

Remark that the operation $\circ$ is commutative and assosiative, and has the monotonicity property w.r.t. $\circ$: $X_1 \subseteq Y_1$ and $X_2 \subseteq Y_2$ imply $X_1 \circ X_2 \subseteq Y_1 \circ Y_2$ for any $X_1, X_2, Y_1, Y_2\ (\subseteq M)$.

Proposition 3.3. *Let $X, Y \subseteq M$. Then:*

1. $X \subseteq Y^{\hat{\bot}}$ *iff* $X \circ Y \subseteq \hat{\bot}$,
2. *if* $X \subseteq Y$ *then* $X \circ Y^{\hat{\bot}} \subseteq \hat{\bot}$,
3. $X \circ X^{\hat{\bot}} \subseteq \hat{\bot}$,
4. *if* $X \subseteq Y$ *then* $Y^{\hat{\bot}} \subseteq X^{\hat{\bot}}$,
5. *if* $X \subseteq Y$ *then* $X^{\hat{\bot}\hat{\bot}} \subseteq Y^{\hat{\bot}\hat{\bot}}$,
6. $X \subseteq X^{\hat{\bot}\hat{\bot}}$,
7. $(X^{\hat{\bot}\hat{\bot}})^{\hat{\bot}\hat{\bot}} \subseteq X^{\hat{\bot}\hat{\bot}}$,
8. $X^{\hat{\bot}\hat{\bot}} \circ Y^{\hat{\bot}\hat{\bot}} \subseteq (X \circ Y)^{\hat{\bot}\hat{\bot}}$,
9. $x \in X^{\hat{\bot}}$ *iff* $\{x\} \circ X \subseteq \hat{\bot}$,
10. *if* $X \circ Y \subseteq \hat{\bot}$ *then* $X \circ Y^{\hat{\bot}\hat{\bot}} \subseteq \hat{\bot}$.

Note that $\cdot^{\hat{\bot}\hat{\bot}}$ is a closure operator similar to *cl* discussed in the previous section.

Proposition 3.4. *Let $X, Y \subseteq M$. Then:*

1. $X^{\hat{\perp}}$ *is a fact,*
2. $X^{\hat{\perp}\hat{\perp}}$ *is the smallest fact that includes X,*
3. *if X and Y are facts, then so is $X \cap Y$,*
4. *if X_i for all $i \in \omega$ are facts, then so is $\bigcap_{i \in \omega} X_i$.*

Definition 3.5. *Let $A, B \subseteq M$. We define the following operators and constants:*

1. $\hat{\perp} := \{1\}^{\hat{\perp}}$,
2. $\hat{\mathbf{1}} := \hat{\perp}^{\hat{\perp}} = \{1\}^{\hat{\perp}\hat{\perp}}$,
3. $\hat{\top} := M = \emptyset^{\hat{\perp}}$,
4. $\hat{\mathbf{0}} := \hat{\top}^{\hat{\perp}} = M^{\hat{\perp}} = \emptyset^{\hat{\perp}\hat{\perp}}$,
5. $A \mathbin{\hat{\wedge}} B := A \cap B$,
6. $A \mathbin{\hat{\vee}} B := (A \cup B)^{\hat{\perp}\hat{\perp}}$,
7. $A \mathbin{\hat{*}} B := (A \circ B)^{\hat{\perp}\hat{\perp}}$,
8. $A \mathbin{\hat{+}} B := (A^{\hat{\perp}} \circ B^{\hat{\perp}})^{\hat{\perp}}$,
9. $\hat{!}A := (A \cap \hat{I})^{\hat{\perp}\hat{\perp}}$,
10. $\hat{?}A := (A^{\hat{\perp}} \cap \hat{I})^{\hat{\perp}}$.

We can show that, by Proposition 3.4, the constants defined above are facts and the operators defined above are closed under D_M.

Definition 3.6. Modality-indexed valuations ϕ^ι *for all $\iota \in K^*$ on a phase space $\langle M, \hat{\perp}, \hat{I} \rangle$ are mappings which assign a fact to each propositional variables. Each modality-indexed valuation ϕ^ι ($\iota \in K^*$) can be extended to a mapping $\cdot^\iota$ ($\iota \in K^*$) from the set Φ of all formulas to D_M by:*

1. $p^\iota := \phi^\iota(p)$ *for any propositional variable p,*
2. $\perp^\iota := \hat{\perp}$,
3. $\mathbf{1}^\iota := \hat{\mathbf{1}}$,
4. $\top^\iota := \hat{\top}$,
5. $\mathbf{0}^\iota := \hat{\mathbf{0}}$,
6. $(\alpha^\perp)^\iota := (\alpha^\iota)^{\hat{\perp}}$,
7. $(\alpha \wedge \beta)^\iota := \alpha^\iota \mathbin{\hat{\wedge}} \beta^\iota$,
8. $(\alpha \vee \beta)^\iota := \alpha^\iota \mathbin{\hat{\vee}} \beta^\iota$,
9. $(\alpha * \beta)^\iota := \alpha^\iota \mathbin{\hat{*}} \beta^\iota$,
10. $(\alpha + \beta)^\iota := \alpha^\iota \mathbin{\hat{+}} \beta^\iota$,
11. $(!\alpha)^\iota := \hat{!}(\alpha^\iota)$,
12. $(?\alpha)^\iota := \hat{?}(\alpha^\iota)$,
13. $(\heartsuit_i \alpha)^\iota := \alpha^{\iota \heartsuit_i}$,

14. $(\heartsuit_{\mathrm{F}}\alpha)^{\iota} := \bigcap_{\kappa \in K^*} \alpha^{\iota\kappa}$,

15. $(\heartsuit_{\mathrm{D}}\alpha)^{\iota} := (\bigcup_{\kappa \in K^*} \alpha^{\iota\kappa})^{\hat{\perp}\hat{\perp}}$.

We call the values α^{ι} $(\iota \in K^*)$ *the* inner-values *of* α $(\in \Phi)$.

Definition 3.7. $\langle M, \hat{\perp}, \hat{I}, \{\phi^{\iota}\}_{\iota \in K^*}\rangle$ *is a* modality-indexed phase model *if* $\langle M, \hat{\perp}, \hat{I}\rangle$ *is a phase space and* ϕ^{ι} $(\iota \in K^*)$ *are modality-indexed valuations on* $\langle M, \hat{\perp}, \hat{I}\rangle$. *A sequent* $\vdash \alpha$ *is* true *in a modality-indexed phase model* $\langle M, \hat{\perp}, \hat{I}, \{\phi^{\iota}\}_{\iota \in K^*}\rangle$ *if* $\alpha^{\emptyset\hat{\perp}} \subseteq \hat{\perp}$ *(or equivalently* $1 \in \alpha^{\emptyset}$*), and* valid *in a phase space* $\langle M, \hat{\perp}, \hat{I}\rangle$ *if it is true for any modality-indexed valuations* ϕ^{ι} $(\iota \in K^*)$ *on the phase space. A sequent* $\vdash \alpha_1, \cdots, \alpha_n$ *is* true *in a modality-indexed phase model* $\langle M, \hat{\perp}, \hat{I}, \{\phi^{\iota}\}_{\iota \in K^*}\rangle$ *if* $\vdash \alpha_1 + \cdots + \alpha_n$ *is true in the model, and* valid *in a phase space* $\langle M, \hat{\perp}, \hat{I}\rangle$ *if it is true for any modality-indexed valuations* ϕ^{ι} $(\iota \in K^*)$ *on the phase space.*

Theorem 3.8 (Soundness). *If a sequent S is provable in* MCLL*, then S is valid for any phase space.*

Proof. By induction on the length of the proof P of S. For example, if the last rule of inference in P is of the form:

$$\frac{\vdash \Gamma'}{\vdash \Gamma}$$

where $\Gamma \equiv \{\alpha_1, \cdots, \alpha_n\}$ and $\Gamma' \equiv \{\alpha'_1, \cdots, \alpha'_n\}$, then we show that $(\alpha'_1 + \cdots + \alpha'_n)^{\emptyset\hat{\perp}} \subseteq \hat{\perp}$ implies $(\alpha_1 + \cdots + \alpha_n)^{\emptyset\hat{\perp}} \subseteq \hat{\perp}$, i.e. $(\alpha_1'^{\emptyset\hat{\perp}} \circ \cdots \circ \alpha_n'^{\emptyset\hat{\perp}})^{\hat{\perp}\hat{\perp}} \subseteq \hat{\perp}$ implies $(\alpha_1^{\emptyset\hat{\perp}} \circ \cdots \circ \alpha_n^{\emptyset\hat{\perp}})^{\hat{\perp}\hat{\perp}} \subseteq \hat{\perp}$. To show this, it is enough to prove that $\alpha_1'^{\emptyset\hat{\perp}} \circ \cdots \circ \alpha_n'^{\emptyset\hat{\perp}} \subseteq \hat{\perp}$ implies $\alpha_1^{\emptyset\hat{\perp}} \circ \cdots \circ \alpha_n^{\emptyset\hat{\perp}} \subseteq \hat{\perp}$, since we have Proposition 3.3 (10) and (6). $\Gamma^{\emptyset\hat{\perp}}$ denotes $\hat{\perp}$ if Γ is empty, and $\Gamma^{\emptyset\hat{\perp}}$ denotes $\gamma_1^{\emptyset\hat{\perp}} \circ \cdots \circ \gamma_n^{\emptyset\hat{\perp}}$ if $\Gamma \equiv \{\gamma_1, \cdots, \gamma_n\}$. In the proof, we will sometimes use the properties in Propositions 3.3 and 3.4 implicitly. Here we show only the following case.
Case ($\heartsuit_{\mathrm{F}}$): The last inference of P is of the form:

$$\frac{\{\vdash \Gamma, \iota\kappa\alpha\}_{\kappa \in K^*}}{\vdash \Gamma, \iota\heartsuit_{\mathrm{F}}\alpha}\ (\heartsuit_{\mathrm{F}}).$$

Suppose $\forall\kappa \in K^*\ [\Gamma^{\emptyset\hat{\perp}} \circ (\iota\kappa\alpha)^{\emptyset\hat{\perp}} \subseteq \hat{\perp}]$, i.e., $\forall\kappa \in K^*\ [\Gamma^{\emptyset\hat{\perp}} \circ (\alpha^{\iota\kappa})^{\hat{\perp}} \subseteq \hat{\perp}]$. Then we obtain (*): $\forall\kappa \in K^*\ [\Gamma^{\emptyset\hat{\perp}} \subseteq \alpha^{\iota\kappa}]$ by Proposition 3.3 (1). We show $\Gamma^{\emptyset\hat{\perp}} \circ (\iota\heartsuit_{\mathrm{F}}\alpha)^{\emptyset\hat{\perp}} \subseteq \hat{\perp}$, i.e., $\Gamma^{\emptyset\hat{\perp}} \circ (\bigcap_{\kappa \in K^*} \alpha^{\iota\kappa})^{\hat{\perp}} \subseteq \hat{\perp}$. This is equivalent to $\Gamma^{\emptyset\hat{\perp}} \subseteq \bigcap_{\kappa \in K^*} \alpha^{\iota\kappa}$ by Proposition 3.3 (1), and hence we show this below. Suppose $x \in \Gamma^{\emptyset\hat{\perp}}$. Then we obtain $\forall\kappa \in K^*\ [x \in \alpha^{\iota\kappa}]$ by (*). This means $x \in \bigcap_{\kappa \in K^*} \alpha^{\iota\kappa}$. ∎

3.3 Completeness and cut-elimination

Next, we consider the strong completeness theorem for MCLL. In order to prove this theorem, we have to construct a canonical model.

Definition 3.9. *We construct a canonical modality-indexed phase model* $\langle M, \hat{\perp}, \hat{I}, \{\phi^{\iota}\}_{\iota \in K^*}\rangle$. *Here M is the set of all multisets of formulas where multiple occurrence of a formula of the form* $\iota?\alpha$ *in the multisets counts only once.* $\langle M, \cdot, 1\rangle$ *is a commutavive monoid where* $\Delta \cdot \Gamma := \Delta \cup \Gamma$ *(the multiset union) for all* $\Delta, \Gamma \in M$, *and* 1 $(\in M)$ *is* $\emptyset$ *(the empty multiset). For any formula* α, *we define*

$[\alpha]_\iota := \{\Delta \mid \vdash_{cf} \Delta, \iota\alpha\}$ *where* $\vdash_{cf} \Delta, \iota\alpha$ *means that* $\vdash \Delta, \iota\alpha$ *is cut-free provable. We call* $[\alpha]_\iota$ $(\iota \in K^*)$ *the* outer-values *of* α $(\in \Phi)$*. We define* $\hat{\bot} := [\bot]_\emptyset = \{\Delta \mid \vdash_{cf} \Delta\}$*.* $\hat{I}$ *is defined as* $\{[\iota_1?\gamma_1, ..., \iota_n?\gamma_n] \mid [\gamma_1, ..., \gamma_n] \in M, \iota_1, ..., \iota_n \in K^*\}$*. The modality-indexed valuations* ϕ^ι $(\iota \in K^*)$ *are defined as* $\phi^\iota(p) := [p]_\iota$ *for any propositional variable* p*.*

Proposition 3.10. *Let* $\hat{I}$ *be* $\{[\iota_1?\gamma_1, ..., \iota_n?\gamma_n] \mid [\gamma_1, ..., \gamma_n] \in M, \iota_1, ..., \iota_n \in K^*\}$*, and* $\dot{I}$ *be* $\{\Delta \in M \mid \Delta \cup \Delta = \Delta\} \cap \hat{\bot}^{\hat{\bot}}$*. Then:* $\hat{I} = \dot{I}$*.*

We then have the following.

Proposition 3.11. $\langle M, \hat{\bot}, \hat{I}\rangle$ *defined in Definition 3.9 is a phase space.*

To prove the completeness theorem, we must prove some lemmas which are analogous to the lemmas established by Okada in (Okada, 1999).

Lemma 3.12. *For any* $\iota \in K^*$ *and any formula* α*, if* $\alpha^\iota \subseteq [\alpha]_\iota$ *then* $\{\iota\alpha\} \in \alpha^{\iota\hat{\bot}}$*.*

Lemma 3.13. *Let* α *be any formula. Then:*

1. $[\alpha]_\iota^{\hat{\bot}\hat{\bot}} = [\alpha]_\iota$,
2. $[\heartsuit_i\alpha]_\iota = [\alpha]_{\iota\heartsuit_i}$,
3. $[\heartsuit_{\rm F}\alpha]_\iota = \bigcap_{\kappa\in K^*} [\alpha]_{\iota\kappa}$,
4. $[\heartsuit_{\rm D}\alpha]_\iota = \bigcup_{\kappa\in K^*} [\alpha]_{\iota\kappa}$.

Proof. We show only (4).
(4): First, we show $[\heartsuit_{\rm D}\alpha]_\iota \subseteq \bigcup_{\kappa\in K^*} [\alpha]_{\iota\kappa}$. To show this, we use the fact that the following rule is admissible in cut-free MCLL:

$$\frac{\vdash \Gamma, \iota\heartsuit_{\rm D}\alpha}{\vdash \Gamma, \iota\alpha}\ (\heartsuit_{\rm D}^{-1}).$$

Suppose $\Gamma \in [\heartsuit_{\rm D}\alpha]_\iota$, i.e., $\vdash_{cf} \Gamma, \iota\heartsuit_{\rm D}\alpha$. We show $\Gamma \in \bigcup_{\kappa\in K^*} [\alpha]_{\iota\kappa}$, i.e., $\exists\kappa \in K^*$ $(\vdash_{cf} \Gamma, \iota\kappa\alpha)$. By applying the rule $(\heartsuit_{\rm D}^{-1})$ to the hypothesis $\vdash_{cf} \Gamma, \iota\heartsuit_{\rm D}\alpha$, we obtain $\vdash_{cf} \Gamma, \iota\alpha$. Next, we show $\bigcup_{\kappa\in K^*} [\alpha]_{\iota\kappa} \subseteq [\heartsuit_{\rm D}\alpha]_\iota$. Suppose $\Gamma \in \bigcup_{\kappa\in K^*} [\alpha]_{\iota\kappa}$, i.e., $\exists\kappa \in K^*$ $(\vdash_{cf} \Gamma, \iota\kappa\alpha)$. By applying the rule $(\heartsuit_{\rm D})$ to this, we obtain $\vdash_{cf} \Gamma, \iota\heartsuit_{\rm D}\alpha$, i.e., $\Gamma \in [\heartsuit_{\rm D}\alpha]_\iota$. ∎

Using Lemmas 3.12 and 3.13, we can prove the following main lemma.

Lemma 3.14. *For any formula* α *and any* $\iota \in K^*$*,* $\alpha^\iota \subseteq [\alpha]_\iota$*.*

Proof. By induction on the complexity of α.
• Base step: Obvious by the definitions.
• Induction step: We show some cases. Other cases are almost the same as those in (Okada, 1999).
(Case $\alpha \equiv \heartsuit_i\beta$): Suppose $\Gamma \in (\heartsuit_i\beta)^\iota$, i.e., $\Gamma \in \beta^{\iota\heartsuit_i}$. Then we have $\Gamma \in \beta^{\iota\heartsuit_i} \subseteq [\beta]_{\iota\heartsuit_i}$ by induction hypothesis, and hence obtain $\vdash_{cf} \Gamma, \iota\heartsuit_i\beta$, i.e., $\vdash_{cf} \Gamma, \iota(\heartsuit_i\beta)$. Therefore $\Gamma \in [\heartsuit_i\beta]_\iota$.

(Case $\alpha \equiv \heartsuit_{\mathrm{F}}\beta$): Suppose $\Gamma \in (\heartsuit_{\mathrm{F}}\beta)^{\iota}$, i.e., $\Gamma \in \bigcap_{\kappa \in K^*} \beta^{\iota\kappa}$. Then we have $\Gamma \in \bigcap_{\kappa \in K^*} \beta^{\iota\kappa} \subseteq \bigcap_{\kappa \in K^*} [\beta]_{\iota\kappa}$ by induction hypothesis, and hence obtain $\forall\kappa \in K^*$ ($\Gamma \in [\beta]_{\iota\kappa}$), i.e., $\{\vdash_{cf} \Gamma, \iota\kappa\beta\}_{\kappa \in K^*}$. By applying the rule ($\heartsuit_{\mathrm{F}}$) to this, we obtain $\vdash_{cf} \Gamma, \iota\heartsuit_{\mathrm{F}}\beta$. Therefore $\Gamma \in [\heartsuit_{\mathrm{F}}\beta]_{\iota}$.
(Case $\alpha \equiv \heartsuit_{\mathrm{D}}\beta$): We will show:

$$(\heartsuit_{\mathrm{D}}\beta)^{\iota} =_{df} (\bigcup_{\kappa \in K^*} \beta^{\iota\kappa})^{\hat{\perp}\hat{\perp}} \subseteq (\bigcup_{\kappa \in K^*} [\beta]_{\iota\kappa})^{\hat{\perp}\hat{\perp}} \subseteq [\heartsuit_{\mathrm{D}}\beta]_{\iota}.$$

For this, $(\bigcup_{\kappa \in K^*} \beta^{\iota\kappa})^{\hat{\perp}\hat{\perp}} \subseteq (\bigcup_{\kappa \in K^*} [\beta]_{\iota\kappa})^{\hat{\perp}\hat{\perp}}$ can be proved by the induction hypothesis $\beta^{\iota\kappa} \subseteq [\beta]_{\iota\kappa}$ and Proposition 3.3 (5), i.e., the monotonicity of $\cdot^{\hat{\perp}\hat{\perp}}$. Next, we prove $(\bigcup_{\kappa \in K^*} [\beta]_{\iota\kappa})^{\hat{\perp}\hat{\perp}} \subseteq [\heartsuit_{\mathrm{D}}\beta]_{\iota}$. By Lemma 3.13 (4), we have $\bigcup_{\kappa \in K^*} [\beta]_{\iota\kappa} \subseteq [\heartsuit_{\mathrm{D}}\beta]_{\iota}$. Moreover, by Proposition 3.3 (5), i.e., the monotonicity of $\cdot^{\hat{\perp}\hat{\perp}}$, we obtain $(\bigcup_{\kappa \in K^*} [\beta]_{\iota\kappa})^{\hat{\perp}\hat{\perp}} \subseteq [\heartsuit_{\mathrm{D}}\beta]_{\iota}^{\hat{\perp}\hat{\perp}}$. By Lemma 3.13 (1), we have $[\heartsuit_{\mathrm{D}}\beta]_{\iota}^{\hat{\perp}\hat{\perp}} = [\heartsuit_{\mathrm{D}}\beta]_{\iota}$. Therefore $(\bigcup_{\kappa \in K^*} [\beta]_{\iota\kappa})^{\hat{\perp}\hat{\perp}} \subseteq [\heartsuit_{\mathrm{D}}\beta]_{\iota}$. ∎

Theorem 3.15 (Strong completeness)**.** *If a sequent S is valid for any phase space, then S is provable in cut-free* MCLL.

Proof. Lemma 3.14 implies this theorem as follows. Let $\vdash \alpha_1, ..., \alpha_n$ be S, and $\vdash \alpha$ be $\vdash \alpha_0 + \cdots + \alpha_n$. If $\vdash \alpha$ is true, then $\emptyset \in \alpha^{\emptyset}$. On the other hand, we have $\alpha^{\iota} \subseteq [\alpha]_{\iota}$ for any $\iota \in K^*$, and hence obtain $\emptyset \in [\alpha]_{\emptyset}$. This means that $\vdash \alpha$ (i.e., S) is provable in cut-free MCLL. ∎

Theorem 3.16 (Cut-elimination)**.** *The rule* (cut) *is admissible in cut-free* MCLL*.*

Proof. If a sequent S is provable in MCLL, then S is valid by Theorem 3.8 (Soundness). By Theorem 3.15 (Strong completeness), S is provable in cut-free MCLL. ∎

4. Related works

It is known that LLs are useful for formalizing and analyzing multi-agent systems (Harland & Winikoff, 2001; 2002; Pham & Harland, 2007). In (Harland & Winikoff, 2001), LLs were used by Harland and Winikoff as a basis for BDI (Belief, Desire, Intention)-style agent systems. In (Harland & Winikoff, 2002), the notion of negotiation in multi-agent systems was discussed by Harland and Winikoff based on LLs. As mentioned in (Harland & Winikoff, 2002), the resource-sensitive character of LLs is more appropriate for handling agent negotiation. In (Pham & Harland, 2007), a temporal version of LLs was used by Pham and Harland as a basis for flexible agent interactions.
In order to directly express agents' knowledge, some *epistemic linear logics*, which have some knowledge operators, have been introduced and studied by several researchers (Baltag et al., 2007; Kamide, 2006). Epistemic linear and affine logics, which are formulated as Hilbert-style axiomatizations, were proposed by Kamide (Kamide, 2006). The completeness theorems with respect to Kripke semantics for these epistemic logics were shown by him. A resource-sensitive dynamic epistemic logic, which is based on a sequent calculus for a non-commutative ILL with some program operators, was introduced by Baltag et al. (Baltag et al., 2007). The completeness theorem with respect to epistemic quantales for this logic was proved by them.

A fixed point linear logic μMALL$^=$ which has some least and greatest fixed point operators was introduced and studied by Baelde and Miller (Baelde, 2009; Baelde & Miller, 2007). The logic μMALL$^=$ enjoys cut-elimination and has a complete focused proof system. μMALL$^=$, also called μMALL, was motivated to offer a natural framework for reasoning about automata (Baelde, 2009). The least fixed point operator μ in μMALL$^=$ is formalized using the following inference rule:

$$\frac{\vdash \Gamma, B(\mu B)\mathbf{t}}{\vdash \Gamma, \mu B\mathbf{t}}\ (\mu)$$

where B represents a formula abstracted over a predicate and terms, and $\mathbf{t}$ represents a vector of terms. Compared with $\heartsuit_{\mathrm{F}}$ in MCLL, the operator μ in μMALL$^=$ does not use an infinitary rule.

Some linear logics with some additional modal operators have been proposed by some researchers. For example, *(linear-time) temporal linear logics*, which are roughly regarded as special cases of fixpoint linear logics, were studied by Kanovich and Ito (Kanovich, 1997) and by Kamide (Kamide, 2010).

5. Acknowledgments

This research was supported by the Japanese Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Young Scientists (B) 20700015.

6. References

Baelde, D. (2009). On the proof theory of regular fixed points, Proceedings of the 18th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2009), *Lecture Notes in Artificial Intelligence* 5607, pp. 93–107.

Baelde, D. & Miller, D. (2007). Least and greatest fixed points in linear logic, Proceedings of the 14th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2007), *Lecture Notes in Artificial Intelligence* 4790, pp. 92–106.

Baltag, A., Coecke, B. & Sadrzadeh, M. (2007). Epistemic actions as resources, *Journal of Logic and Computation* 17 (3), pp. 555–585.

Emerson, E.A. (1990). Temporal and modal logic, In Handbook of Theoretical Computer Science, Formal Models and Semantics (B), Jan van Leeuwen (Ed.), pp. 995-1072, Elsevier and MIT Press.

Fagin, R., Halpern, J.Y., Moses, Y. & Vardi, M.Y. (1995). Reasoning about knowledge, MIT Press.

Girard, J.-Y. (1987). Linear logic, *Theoretical Computer Science* 50, pp. 1–102.

Halpern, J.H. & Moses, Y. (1992). A guide to completeness and complexity for modal logics of knowledge and beliefs, *Artificial Intelligence* 54, pp. 319–379.

Harland, J. & Winikoff, M. (2001). Agents via mixed-mode computation in linear logic: A proposal, Proceedings of the ICLP'01 Workshop on Computational Logic in Multi-Agent Systems (CLIMA-2).

Harland, J. & Winikoff, M. (2002). Agent negotiation as proof search in linear logic, Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 938–939.

Kamide, N. (2006). Linear and affine logics with temporal, spatial and epistemic operators, *Theoretical Computer Science* 353 (1–3), pp. 165–207.

Kamide, N. (2010). Phase semantics for linear-time formalism, *Logic Journal of the IGPL* (to appear).

Kanovich, Max I. & Ito, T. (1997). Temporal linear logic specifications for concurrent processes (extended abstract), Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science (LICS), pp. 48-57.

Lewis, D.K. (1969). Convention: A philosophical study, Harvard University Press, 1969.

Lismont, L. & Mongin, P. (1994). On the logic of common belief and common knowledge, *Theory and Decision* 37, pp. 75–106.

Meyer, J.-J. & van der Hoek, W. (1995). Epistemic logic for AI and Computer Science, Cambridge Tracts in Theoretical Computer Science 41, Cambridge University Press.

Miller, D. (2004). An overview of linear logic programming, T. Ehrhard, J.-Y. Girard, P. Ruet and P. Scott (Eds.), London Mathematical Society Lecture Note Series 316 (Linear logic in computer science), pp. 119–150, Cambridge University Press.

Okada, M. (1999). Phase semantic cut-elimination and normalization proofs of first- and higher-order linear logic, *Theoretical Computer Science* 227, pp. 333–396.

Okada, M. (2002). A uniform semantic proof for cut-elimination and completeness of various first and higher order logics, *Theoretical Computer Science* 281, pp. 471–498.

Pham, D.Q. & Harland, J. (2007). Temporal linear logic as a basis for flexible agnt interactions, Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), 28.

Pnueli, A. (1997). The temporal logic of programs, Proceedings of the 18th IEEE Symposium on Foundations of Computer Science, pp. 46-57.

Troelstra, A.S. (1992). Lectures on linear logic, CSLI Lecture Notes, Vol. 29, Stanford, CA: CSLI.

9

Fast Visual Tracking of Mobile Agents *

Andelko Katalenic[1], Ivica Draganjac[2], Alan Mutka[3] and Stjepan Bogdan[4]

[1]*Eindhoven University of Technology*

[2,3,4]*Faculty of Electrical Engineering and Computing University of Zagreb*

[1]*The Netherlands*

[2,3,4]*Croatia*

1. Introduction

Control and coordination of agents in multi-agent systems is demanding and complex task. Applications like search and rescue Jennings et al. (1997), mapping of unknown environments or just simple moving in formation demands gathering a lot of information from surrounding. A local sensor-based information is preferred in large formations due to observability constraints. On the other hand, local information processing requires sophisticated and expensive hardware that would manage to gather and process sensor information in real-time. Furthermore, refined sensors, such as a camera, entail time consuming algorithms. Generally, it is difficult to provide satisfactory results by implementation of such algorithms in real-time. For this reason immense research efforts have been put in development of fast and simple image analysis methods. This is specifically noticeable in the field of robot vision, a discipline that strives to solve problems such as robot localization and tracking, formation control, obstacle avoidance, grasping and so on. Visual tracking of robots in formation is usually based on visual markers. Algorithms for detection of predefined shapes or colors are simple enough to be executed even on low-cost embedded computers in real-time. In Chiem & Cervera (2004) pose estimation based on the tracking of color regions attached to the robot is presented. The position of the leader robot is estimated at video rate of 25 frames per second. The main disadvantage of the proposed method is marker position on the robot, i.e. a marker can be recognized only from particular angle. Authors in Cruz et al. (2007) accomplish robot identification and localization by using visual tags arranged on the back of each robot on a 3D-truncated octagonal-shaped structure. Each face of visual tag has a code that provides the vehicle's ID as well as the position of the face in the 3D-visual marker. This information allows a vision sensor to identify the vehicle and estimate its pose relative to the sensor coordinate system. A robot formation control strategy based on visual pose estimation is presented in Renaud et al. (2004). Robots visual perception is enhanced by the control of a motorized zoom, which gives the follower robot a large field of view and improves leader detection. A position-based visual servo control strategy for leader-follower formation control of unmanned ground vehicles is proposed in Dani et al. (2009). The relative pose and the relative velocity are obtained using a geometric pose estimation technique and a nonlinear velocity estimation strategy. The geometric pose estimation technique Gans (2008)

*This article is extended version of the paper originally presented at The 4th IEEE Conference on Industrial Electronics and Applications (ICIEA 2009)

uses Euclidean homography relationships and a single known geometric length on a single following object.
Visual detection of an object without usage of additional markers might be very difficult. In some cases, it is important to recognize and localize an object that has to be picked up or manipulated (stationary object), while in other cases, it is paramount to identify and pinpoint an object that needs to be followed (object in motion). Object detection algorithms are usually based on matching captured images with stored object models. Tomono Tomono (2006) proposed a high density indoor map-based visual navigation system on the basis of on-line recognition and shape reconstruction of 3D objects, using stored object models. However, the implemented Scale-invariant feature transform (SIFT) extraction is compute-intensive, and a real-time or even super-real-time processing capability is required to provide satisfactory results.
Currently, the fastest object detection is provided by cascades of classifiers based on Haar-like features Viola & Jones (2001). Apart from low execution times, the statistical nature of an this algorithm based on Haarlike features is robust to motion blur and changing lighting conditions because those features represent basic elements of the picture including edges, lines and dots. The cascades are trained using *AdaBoost* learning algorithm Matas & Sochman (Oct. 2008) and can be calculated very fast using *integral image*. In Choi (2006) real time on-road vehicle detection with optical flow and Haar-like feature is presented. The algorithm detects vehicles moving in the same direction up to 88% of accuracy. Fast vision-based pedestrian detection using Haar-like features is presented in Monteiro et al. (2006). Result shows that system can detect pedestrian at 17 frames per second on 1.7 GHz processor with pedestrian detection rate up to 90% of accuracy. Thorough review of image processing methods in visual navigation for mobile robots is given in Bonin-Font et al. (2008).
Additional estimators are used in order to increase robustness and reliability of a system based on vision system in the loop. A vision-based range estimator for leader-follower based on Immersion and Invariance is presented in Morbidi et al. (2010). The proposed reduced-order nonlinear observer is simple to implement, easy to tune and achieves global asymptotical convergence of the observation error to zero. The extended and unscented Kalman filters have been used in Mariottini et al. (2007) and Mariottini et al. (2005) to estimate the robots' relative distance.
In this paper we present an implementation of an algorithm for fast real-time visual tracking and localization in multi-agent system comprised of Wifibot mobile platforms Lab (2005). Mobile platforms, equipped with IP cameras, are moving in formation that is established and maintained by using a visual feedback. Leader agent is followed by follower agent which uses only images captured by its camera as feedback information.
The paper is organized as follows. In section 2 algorithm for fast object detection, using Haar-like features and integral image has been introduced followed by the description of strong classifiers and cascades of classifiers. Section 3 introduces the method for the platform position and orientation detection, using five differently trained cascades of classifiers. In section 4, the laboratory setup for platform tracking, comprised of two Wifibot platforms is described Design of Extended Kalman Filter is presented in section 5, while experimental results on platform localization are given in section 6.

2. Fast visual detection algorithm

2.1 Features and integral image

The algorithm for visual detection and localization of mobile agents, described in this article, is based on the method called "rapid object detection using boosted cascade of simple features", introduced by Viola and Jones Viola & Jones (2001). This algorithm classifies images based on the value of simple upright features made up of white and gray rectangles. The value of proposed features is defined as the difference between the sum of the image pixels within white and gray regions. In order to improve the efficiency of the algorithm, Leinhart and Maydt Lienhart & Maydt (2002) have extended the basic set of haar-like features by the set of 45° rotated rectangles. The extended set of features that has been used in the proposed object detection system is shown in Fig 1.
The value of a feature can be calculated by

$$feature_I = \sum_{i \epsilon I = \{1,...,N\}} \omega_i \cdot RecSum(r_i), \tag{1}$$

where $RecSum(r_i)$ denotes pixel sum within i-th feature rectangle, $\omega_i \epsilon \mathbf{R}$ represent the weights with opposite signs, used for compensation of the differences in area size between two rectangles, and N is the number of rectangles the feature is composed of. In Lienhart & Maydt (2002) a method for fast calculation value of upright and rotated features at all scales at constant time has been proposed. Authors proposed an intermediate representation for the image called the *integral image* Viola & Jones (2001). The method is based on two auxiliary images - *Summed Area Table* $SAT(x, y)$ for upright rectangles and *Rotated Summed Area Table* $RSAT(x, y)$ for rotated rectangles. $SAT(x, y)$, defined by

$$SAT(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y'), \tag{2}$$

can be calculated with one pass over all pixels in the image and it represents the sum of the pixels of the upright rectangle with top left corner at $(0, 0)$ and bottom right corner at (x, y).

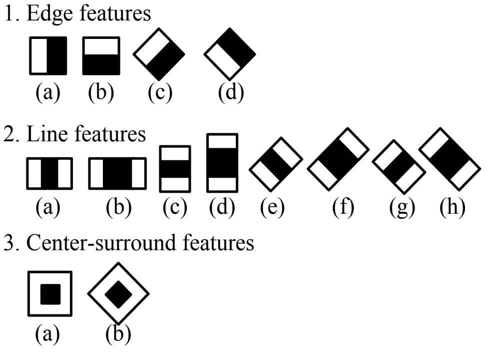


Fig. 1. Feature prototypes of simple Haar-like and center-surround features used for the platform detection

On the other hand, for the calculation of $RSAT(x, y)$, defined by

$$RSAT(x,y) = \sum_{x' \leq x, x' \leq x - |y - y'|} I(x', y'), \tag{3}$$

two passes over all pixels in the image are needed. $RSAT(x, y)$ gives the sum of pixels within rectangle rotated for 45° having the most right corner at (x, y). Once equations (2) and (3) have been calculated, only four lookup tables and three arithmetic operations are required to determine pixel sum of any upright or 45° rotated rectangle. Accordingly, the difference between two rectangle sums can be determined in maximally eight lookup tables and six arithmetic operations.

2.2 Simple and strong classifiers

Having defined the feature set, the training set of positive images (containing the object of interest) and the training set of negative images (containing no object), it is necessary to construct a classification function (classifier) that separates positive from negative images. A classifier can consist of one or more features. A classifier consisting of only one feature is also called a week classifier and it is defined by

$$h_j = \begin{cases} 1 & if \quad p_j f_j(x) < p_j \theta_j \\ 0 & otherwise \end{cases} \tag{4}$$

where f_j is a feature, θ_j a threshold, p_j parity indicating the direction of the inequality sign and x represents the image being classified. Because of their low accuracy, which is often within the range of 50-80%, single-featured classifiers are not very useful in practice. Much better detection performance provides a *strong* classifier that is based on a set of features selected in training procedure using *AdaBoost* algorithm described in Matas & Sochman (Oct. 2008) and Viola & Jones (2001). It trains a desired number of weak classifiers for the given image and forms a more complex classifier made as a linear combination of them, namely:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases} \tag{5}$$

Since the detection performance is often strict, it results in a strong classifier, which in order to be executed, requires computing a large number of features. This significantly increases the computational time and makes it inapplicable in real time problems.

2.3 Cascade of classifiers

This problem can be avoided, by construction of a *cascade* of classifiers providing an increased detection performance Viola & Jones (2001). Block diagram of the detection cascade is shown in Fig. 2. The cascade comprises several interconnected stages of classifiers. The detection process starts at first stage that is trained to detect almost all objects of interest and to eliminate as much as possible non-object patterns, i.e. classifiers that detect distinctive features of the object that are often describable using only few of the features from the used feature set (Fig. 1). The role of this classifier is to reduce the number of locations where the further evaluation must be performed. Every next stage consists of more complex classifier, trained to eliminate negative patterns being admitted through previous stage. If the object of interest has been detected at one stage, the detection process continues at the next stage. Otherwise the sub-window being checked is classified as a non-object and discarded immediately. The overall outcome of cascade is positive only in case the object has been detected by all stages.

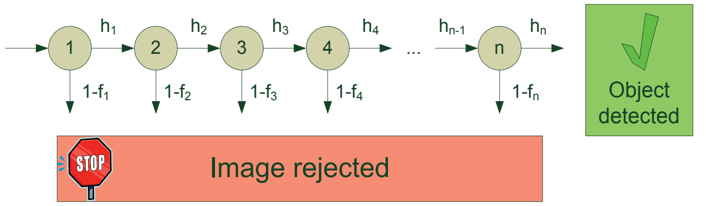


Fig. 2. Block diagram of the detection cascade with n stages

The use of a cascade significantly reduces computation time since the most patterns being tested are negative and thus discarded at lower stages composed of simpler classifiers, thus providing that the evaluation of higher stages occurs only on object-like regions. Every stage of cascade is characterized by hit rate h and false alarm rate f. Accordingly, the overall hit rate of cascade is h^n, and false alarm rate f^n, where n represents the number of stages. Overall, using several stages that have very high hit rate and relatively low false alarm rate results in a cascade with a high overall hit rate and a very low overall false alarm rate, since both overall values are obtained by multiplying individual rates.

3. Platform localization

3.1 Platform detection

Robot localization method presented in this paper identifies the platform location and orientation with respect to the camera, by image analysis using different cascades of classifiers. The main idea is to train several cascades in a way that each cascade is trained on a set of images taken from a specific angle of view. Thereafter, the platform detection is performed by applying all cascades on each frame captured from a Wifibot camera installed on a platform. Information about platform position and orientation can be then obtained by analyzing outputs of all cascades, as they depend on platform orientation. For example, in case detection is performed on a frame containing Wifibot platform with sideways view, it is expected that the best detection results would be obtained by cascade trained to detect platform from this angle of view. It should be noted that the resolution of orientation detection depends on the number of cascades being applied - higher resolution requires more cascades. However, the final number of cascades has to be selected as a trade off between the desired resolution and available computation time.

Since there are no significant differences in the front and the back view of the platform, as well as in the left and the right view, the algorithm for calculation of the angle of view has two solutions, which makes the orientation detection procedure more complicated. This problem has been solved through coloring Wifibot batteries in different colors, taking into account that the grayscale representations of both colors remain unchanged, so these colors do not influence grayscale images used in training process. Additional orientation information in form of different battery colors is obtained in the following way. The method utilizes the fact that an opposite platform view results in an opposite battery order with respect to the left image side. For example, considering that the left battery is colored red and the right battery green, their order in front platform view will be red - green, while in back view it will be green - red. Therefore, locations of batteries in an image has to be detected first, which is accomplished by detecting their colors. Having known the battery locations, their order with respect to the left image side becomes also known. In case of sideways platform view,

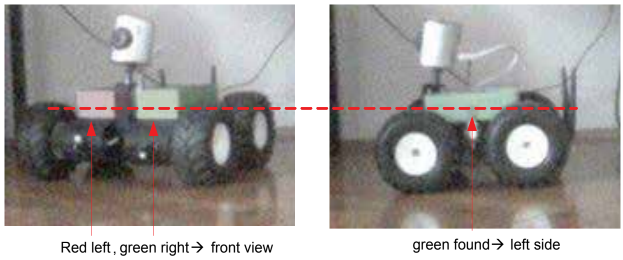
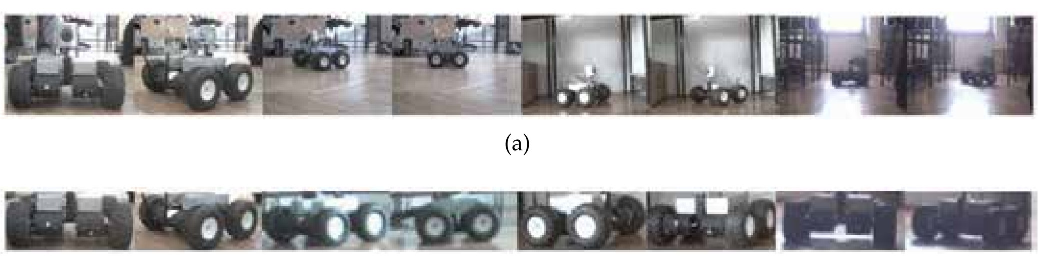


Fig. 3. Solving of the platform symmetry problem

only one battery color is detected depending on which side the platform is seen from. Need for battery color detection makes no significant influence on the overall algorithm execution time. Described strategy for solving the problem of angle of view ambiguity is demonstrated in Fig. 3.

3.2 Cascades training

Careful selection of images needed for training cascades is of special importance for accurate detection of the predefined object. In an effort to obtain a desired system robustness, 110 images containing Wifibot platform under different lighting conditions have been used (Fig. 4(a)). All images were taken by a camera mounted on the platform. During image collection process, the platform was located at different distances and orientations with respect to the platform being photographed. Once the set of positives has been created, only those image regions containing the Wifibot platform have been extracted (Fig. 4(b)). Dimension of extracted region was set to 50x22 pixels. The number of positives has been increased to 4000 by deriving new images characterized by different contrast and rotation angle as shown in Fig. 5. The set of negative images comprised of 5000 samples without Wifibot platform.
The number of cascades used in the system has been set to five. For that reason, the base of positive images has been eye-checked and sorted to five new subsets, each of them containing images of the platform taken from the specific angle of view. Due to the platform symmetry, the process of positive image sorting has been done according to two main angles of view - front-back and sideways. Accordingly, two cascades of classifiers were trained for detection of the platform from this two directions - *Wifi_Front_Back* and *Wifi_Side*. In order to enhance the quality of the orientation detection, two additional cascades of classifiers that provide



(a)

(b)

Fig. 4. Examples of positives, a) images taken by Wifibot camera, b) positive samples created from images

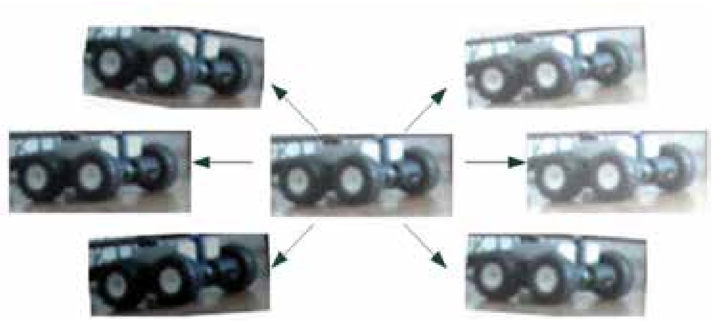


Fig. 5. Deriving of additional positives

detection of the platform from more specified angles were used. One of them, named *Wifi_FB_Extreme* is intended for detection of the platform from strictly front-back view, while *Wifi_Side_Extreme* detects the platform from strictly side view. The last cascade - *Wifi_All* has been trained for the platform detection from any angle of view, thus providing more robust detection of the platform location in an image. Angles of view covered by described cascades are shown in Fig. 6. Training of all cascades has been performed using *HaarTraining* application from *OpenCV* library *OpenCV On-line documentation* (Oct. 2008).

3.3 Searching method

Searching for the platform in current frame is carried out using resizable *searching window* that defines the image area being checked. Three different searching windows can be seen in Fig. 7. When platform detection is executed within these three windows, the platform will be detected only within red-colored searching window. The sides ratio of a searching window is always kept at constant value defined by sides ratio of classifiers used in cascades. Process of searching for the platform starts within the largest searching window that satisfies defined sides ratio and is initially located at the top left image corner. After all five cascades of classifiers have been applied to this image area, the detection continues within other image areas as the searching window shifts through the image until it reaches the right bottom image

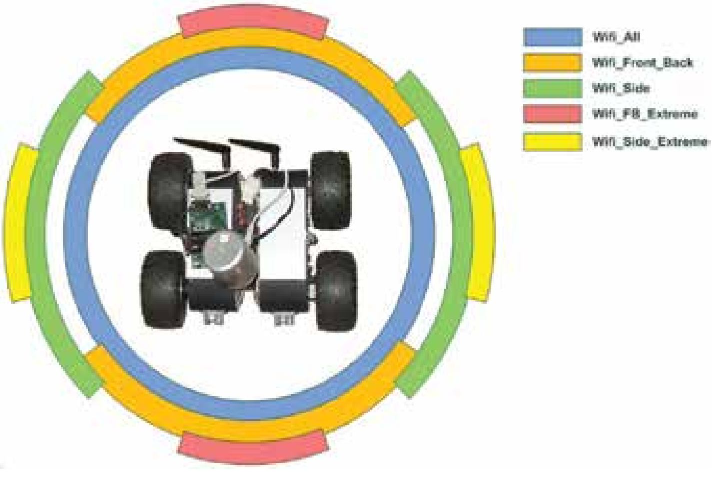


Fig. 6. Angles of view covered by five differently trained cascades

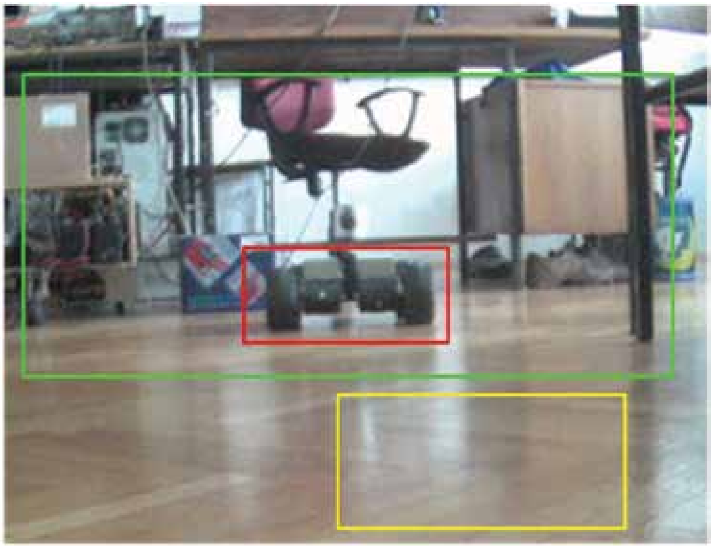


Fig. 7. Three different searching windows

corner. When the whole image has been checked, searching window is resized by predefined *scale factor* and the described procedure is repeated until the searching window reaches predefined *minimum size*. The searching algorithm is very fast, since a feature calculation based on integral image has been used. Described searching method, involving the whole image checking, is used only at the beginning of detection process in order to detect initial platform position. Once detected, the platform is being tracked by searching window which size can vary within 20% of the size of the window that detected the platform in previous frame. A scale factor, used for window resize, has been set to 0.95. Furthermore, this searching window is applied only to a narrow image area close to the last detected platform location. In this way, the speed of searching algorithm has been additionally increased. The algorithm was able to process 15 frames per second on 3.0 GHz Pentium IV processor with 1GB RAM.
Fig. 8 shows the screen of an application developed for validation of a single cascade using static images. After the platform detection has finished, this application highlights all searching windows the platform was detected within. As can be seen, usually there are more than one hits, closely located one to each other, thus called *neighbors*. When the number of neighbors is greater than the predefined *min_neighbours* parameter, the overall detection result is positive, and the platform location in image is highlighted by the representative searching window (blue rectangle in Fig. 8). Generally, all five cascades of classifiers are used in searching procedure which results in five different numbers of neighbors, one for each cascade. Fig. 9 shows the number of neighbors obtained by applying all five cascades to live video stream containing a static sideways view of a Wifibot platform. As can be noticed, each cascade has detected a certain number of neighbors. Moreover those numbers vary as the time is running, due to variations in lightning conditions and camera noise. As expected, the most neighbors have been detected by *Wifi_Side* and *Wifi_Side_Extreme* cascades. Decision making part of described orientation detection system selects a cascade that has detected the most neighbors in current frame. In case the number of neighbors is grater then the predefined *min_neighbours* parameter, selected cascade indicates the angle of view, i.e. current platform orientation with respect to the camera, according to the map shown in Fig. 6. There is one

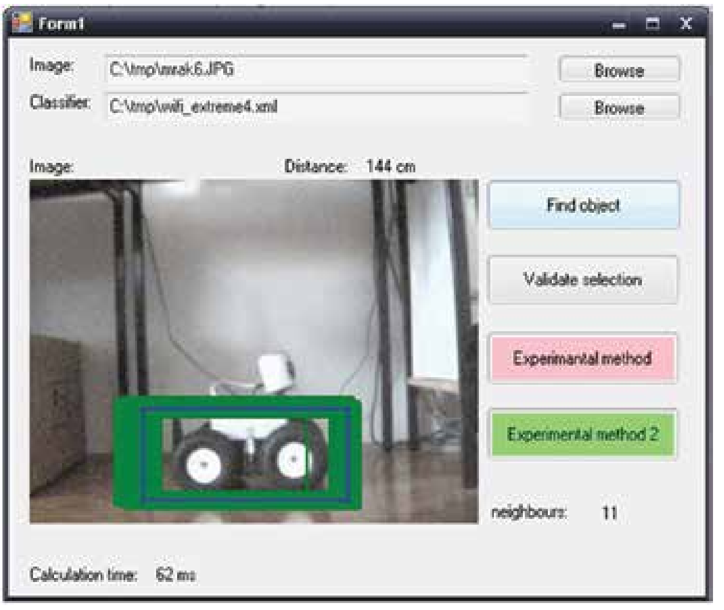


Fig. 8. Windows application for cascade validation on static images

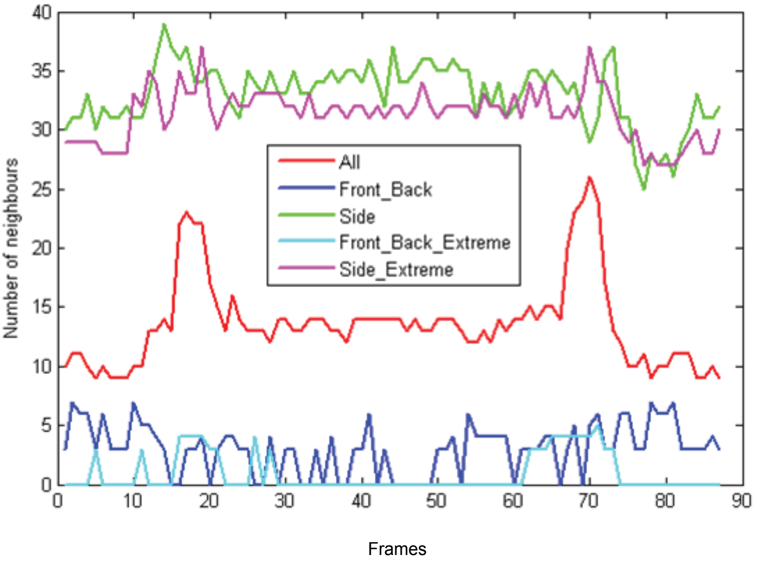
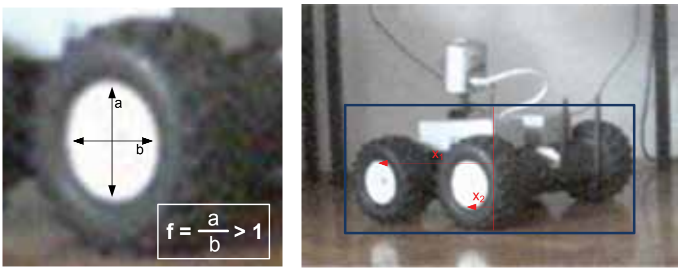


Fig. 9. Number of neighbors obtained by each cascade on a live video stream containing a sideways view of a Wifibot platform

supplementary condition selected cascade must satisfy in order to be declared as a hit - the percentage of its hits in the last ten frames must be greater than the experimentally adjusted parameter *min_percentage*. This condition provides a smooth transition in cascades outputs as the platform rotates.



(a) (b)

Fig. 10. a) axis length ratio, b) wheel distances from the platform center

In order to enhance the quality of orientation detection performed by cascades, an additional image analysis based on the calculation of wheel parameters has been applied. These parameters describe two white elliptic regions within platform wheels, which can be easily detected within positive searching window. One parameter is axis length ratio f (Fig. 10(a)), and the other two parameters $x1$ and $x2$ represent the wheel distances from the platform center (Fig. 10(b)). The angle of view is unambiguously defined by these parameters, and thus can be precisely reconstructed.

4. Platform tracking setup

Described visual-based system for platform localization has been tested on laboratory setup comprised of two Wifibot platforms - leader and follower. The follower's task is to track the remotely controllable leader that performs an arbitrary trajectory, while keeping the predefined inter-distance. The designed tracking control system has been divided into two parts - platform inter-distance control subsystem and subsystem for control of follower orientation with respect to the leader. The subsystem for follower orientation control provides actions needed for follower to face the leader, which is accomplished when the leader is located in the center of image captured by the follower. The distance between the leader and the follower is obtained from the leader size in current frame. An increase in the platform inter-distance causes the decrease in the platform size in an image, and vice versa. This dependency has been experimentally obtained. Once cascades detect position of the platform a distance from camera is calculated based on the width of the frame determined from neighbors (blue rectangle in Figure 8). A function of distance with respect to width is depicted in Fig. 11. Blue dots represent measurements while red line is interpolation of the following form:

$$d(w) = 50850 \cdot e^{-0.07287 \cdot w} + 311.2 \cdot e^{-0.005298 \cdot w} \tag{6}$$

It is evident that relation between width of an object and its distance from a camera is not linear which is characteristic of human eye as well.

The difference between current and desired platform inter-distance causes the follower forward/backward motion, while the leader's displacement from the image center causes the follower rotation until the leader appears in the image center. Block diagram of the designed control system, containing two described subsystems, is shown in Fig. 12. The control

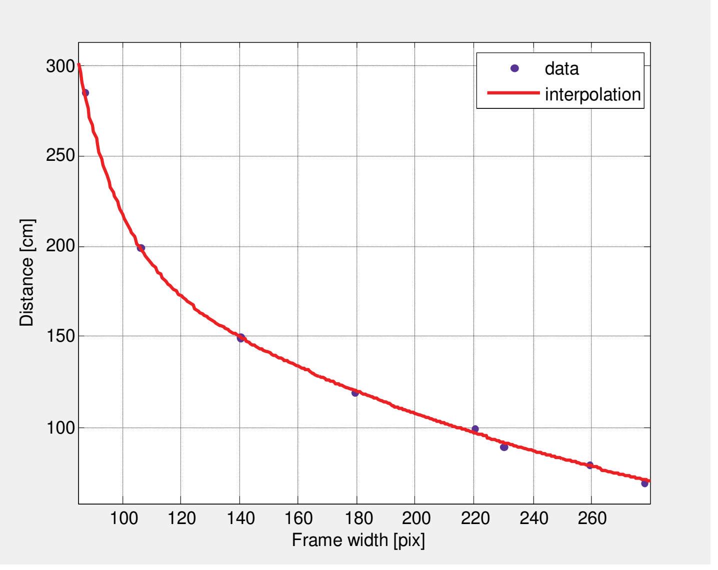


Fig. 11. Distance from camera with respect to the frame width

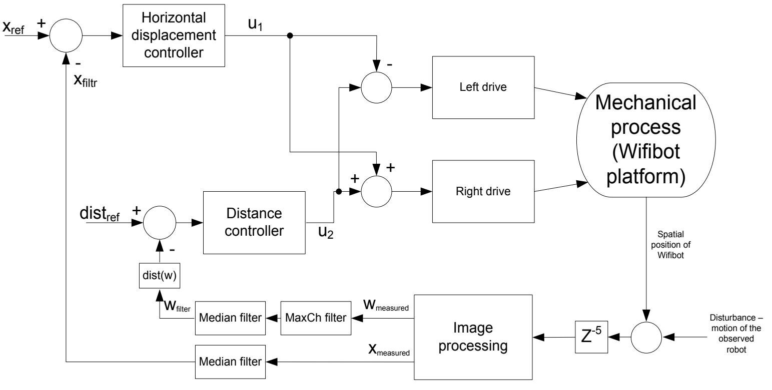


Fig. 12. Block diagram of the system for platform following

algorithm with sample time of Td=50 ms, has been executed on 3.0 GHz Pentium IV processor, while the camera images and commands for platform motion have been transmitted trough the wireless network established between PC and access point embedded in the platform. Due to the noise in angle and distance measurement (Fig. 13) Median and MaxChange filters have been added in the feedback loop. For correct design of tracking controllers an additional delay of 250 ms (5x*Td*), caused by image capture and processing by the camera on board of the platform, is introduced in the feedback loop. Proportional controllers have been used for both - distance and orientation control. It is important to note that the horizontal displacement estimation of a dynamical object using low angle of view and low resolution camera is difficult due to high noise amplification, specially for distant objects that appear small compared to the

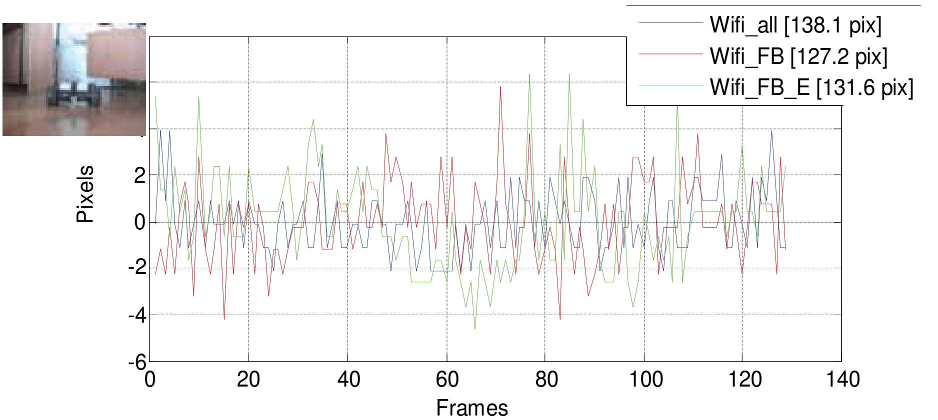


Fig. 13. Noise in calculated distance (*wavg*=133 pix, *d*=160 cm)

picture frame and the noise amplitude of only one pixel represents a very high deviation in the physical displacement of the object.
The estimate of this relation is written in (7).

$$\frac{\Delta x}{\Delta pix} \approx \frac{37}{w}\text{cm} \tag{7}$$

This phenomena can also be observed in the Fig. 11 where the curve tangent becomes very steep for large distances. Also, detecting horizontally moving objects, which are close to the camera, tends to be difficult since they fully appear on the picture frame just for a fraction of time which is comparable to the camera delay. Because of that the usage of unfiltered signals from the vision system in the conventional control schemes generates big overshoots and chattering in controlled signals when tracking very distant or very close objects. Model based filtering is a solution to this problem.

5. Kalman filter design

5.1 Tracked agent dynamics

Synthesis of a control scheme based on the measurements from a visual tracking system used in an environment with changing lighting conditions and obstacles, which very often cover parts of the tracked object, becomes very difficult due to often unavailability of sensor data. Furthermore, the available data is often noisy due to low camera resolution and stochastic nature of the used detection scheme. For this reason, a good estimator and filter is needed in the control loop. In this application, we have used a common technique, namely an *extended Kalman filter* Mohinder S. Grewal (2008). Since no information other than vision is available for the tracked agent, it's distance and horizontal displacement has been modeled by an integrator with an unknown gain, i.e.:

$$\begin{bmatrix} \Theta(k) \\ v_D(k) \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Theta(k-1) \\ v_D(k-1) \end{bmatrix} \tag{8}$$

where Θ represents the object distance D or horizontal displacement x, v_D the object speed in either horizontal or vertical direction and T the discretization period of the control scheme. Also, inverses of (6) and (7) have been used as state measurement functions.

5.2 Host agent dynamics

In the multi-agent systems, both the tracked agent and the host agent are in motion, meaning the host agent motion also influences the tracked agents frame position making the model (8) imprecise. That is why model (8) has to be updated with the host motion profile which in the case of the Wifibot is directly available from the incremental encoders mounted in the wheels. Availability of the direct wheel speed measurements eliminates the need for accurate host robot dynamical model. With host object motion taken into account, the agent distance is modeled with:

$$\begin{bmatrix} D(k) \\ v_D(k) \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} D(k-1) \\ v_D(k-1) \end{bmatrix} + \begin{bmatrix} -2 \cdot T \\ 0 \end{bmatrix} \cdot \bar{N}_{enc}(k), \tag{9}$$

where $\bar{N}_{enc}(k)$ represents a mean value of left and right incremental encoder outputs, namely: $\bar{N}_{enc} = 0.5 \cdot (N_{encL} + N_{encR})$. Furthermore, making a complete dynamical model of the tracked object's horizontal displacement requires knowledge of the host robot dimensions together with left and right incremental encoder outputs.

From Fig. 14 we have: $R = \sqrt{\left(\frac{d}{2}\right)^2 + a^2}$, which gives a relation for the angular velocity of the host agent:

$$\omega = K \cdot \frac{N_{encR} - N_{encL}}{R} \cdot \frac{d}{2R} = \frac{d}{R^2} \cdot \Delta N_{enc}, \tag{10}$$

where ΔN_{enc} represents the difference between the encoder outputs and K is the encoder gain. The small angle approximation $\dot{x} \approx D \cdot \omega \approx \frac{D \cdot d}{R^2} \cdot \Delta N_{enc}$ is then used to obtain the final tracked agent horizontal displacement model:

$$\begin{bmatrix} x(k) \\ v_x(k) \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k-1) \\ v_x(k-1) \end{bmatrix} + \begin{bmatrix} \frac{D \cdot d}{R^2} \cdot T \\ 0 \end{bmatrix} \cdot \Delta N_{enc}(k) \tag{11}$$

Extended Kalman filters based on dynamical models (9) and (11) with inverses of (6) and (7) as state measurement functions are then used as filters to improve signal to noise ratio of the vision system and as the estimators of the tracked object position when there is no data available.The experimental data follows in the next section.

6. Experimental results

Performance of the proposed localization and tracking method has been tested on the experimental setup comprised of two Wifibot platforms in leader-follower formation. First experiment has been conducted on static platform in order to investigate quality of all five cascades.Results obtained by the experiment are presented in Fig. 15(a). Four different postures (angles of view) of Wifibot platform on three different distances have been examined. For each posture and distance (12 cases) 90 frames have been processed (15 fps). In case of front-back angle of view (images 1, 2 and 3) results are very good. Two cascades, *Wifi_All* and *Wifi_Front_Back*, have 100% accuracy. Slight deviation can be noticed with cascade *Wifi_FB_Extreme* in case platform is far away from the camera; only 5.6% accuracy is achieved. However, this result is also useful since *Wifi_Side_Extreme* for the same case is 0%. If one compares *Wifi_Front_Back* with *Wifi_Side* and *Wifi_FB_Extreme* with *Wifi_Side_Extreme*, then for all three images *Wifi_Front_Back* and *Wifi_FB_Extreme* cascades are dominant (presented gray in Fig. 15(a)). For side view (images 4, 5 and 6) cascades *Wifi_Side* and *Wifi_Side_Extreme* are dominant with 100% accuracy which is exceptionally good result. Results obtained

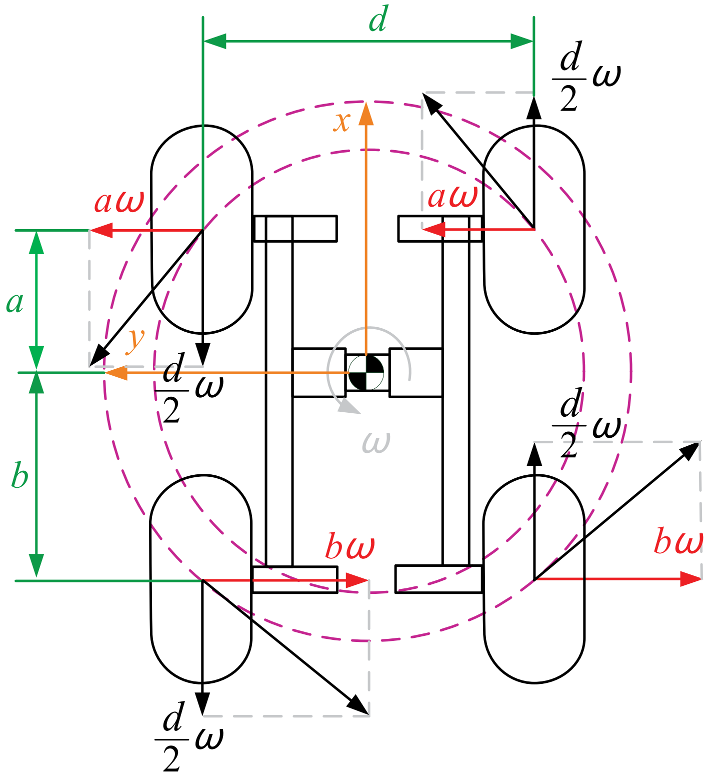
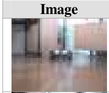
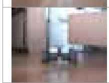
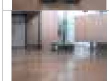
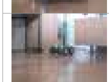
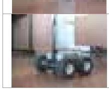


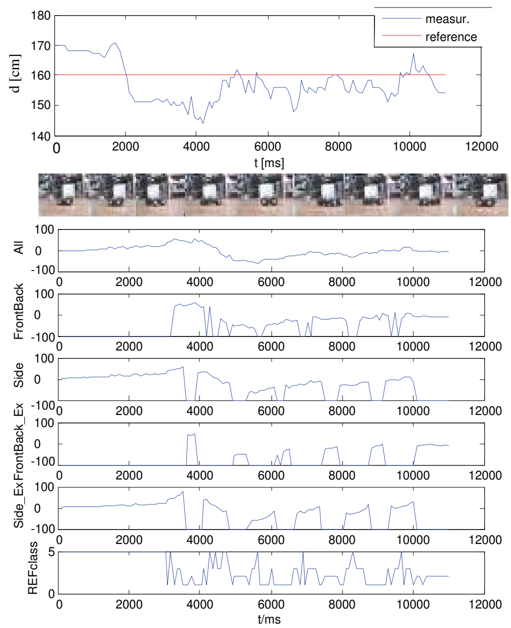
Fig. 14. Wheel positions and dimension of the Wifibot robot ($a = b$)

for conditions when angle of view was somewhere between front-back and side view are demonstrated on images 7 to 12. Although situation is not as clear as for previous examples, in most cases *Wifi_Front_Back* and *Wifi_Side_Extreme* cascades are dominant, which demonstrates that platform is inclined.

Second set of experiments have been performed on moving platform. The leader executes a circular movement (small radius) while follower has been commanded to remain at distance of 160 cm. Results, depicted in Fig. 15(b), are comprised of measured distance (top of the figure), displacement of the leader from the center of image (calculated by cascades) and reference cascade (bottom of the figure). As may be seen the follower keeps the distance error within ś10 cm which is less than 10% of the set point value. In the same time horizontal displacement control algorithm rotates the follower so that leader remains within limits of ś50 pixels from the center of image. Depending on the angle of view, particular cascade detects the leader and calculates its displacement: cascade *Wifi_All* is active all the time during experiment, *Wifi_Side* and *Wifi_Side_Extreme* detect the leader at the beginning of experiment as it moves sideways, while at the same time *Wifi_Front_Back* and *Wifi_FB_Extreme* remain inactive. In 3^{th} second of experiment the leader starts to turn which is immediately recognized by *Wifi_Front_Back* cascade that becomes active together with *Wifi_Side* and *Wifi_Side_Extreme*. Activation and deactivation of other cascades can be easily tracked throughout experiment.

Image	*ALL*	*FB*	*SIDE*	*FB_E*	*SD_E*	*No.*
	1.0	1.0	0.0	0.056	0.0	1
	1.0	1.0	0.0	1.0	0.0	2
	1.0	1.0	0.056	1.0	0.0	3
	1.0	0.26	1.0	0.0	1.0	4
	0.68	0	1.0	0.0	1.0	5
	1.0	0.39	1.0	0.011	1.0	6
	1.0	0.95	0.0	0.0	0.0	7
	1.0	1.0	0.32	0.0	0.033	8
	1.0	1.0	0.0	0.0	0.38	9
	0.95	0.31	0.088	0.0	0.011	10
	0.99	0.32	0.24	0.0	0.39	11
	0.93	0.23	1.0	0.0	1.0	12

(a) Classification results (static images)



(b) Tracking of a circular movement of the leader

Fig. 15. Experimental results

Diagram *REFclass* shows which cascade is dominant at particular moment: 1 = *Wifi_All*, 2 = *Wifi_Front_Back*, 3 = *Wifi_Side*, 4 = *Wifi_Front_Back*, and 5 = *Wifi_Side_Extreme*. The video clips displaying described experiments can be downloaded from http://flrcg.rasip.fer.hr/movies/. Third set of experiments have been performed to test the proposed extended Kalman filter. Figures 16 and 17 show tracked agents horizontal displacement and distance signals with and without using a Kalman filter.

Covariance matrices are set to $P_0 = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, Q = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$ and $R = 10$. An improvement of tracking performance can be seen, specially for agents which are distant from the camera and which appear small in the detected picture. In the final setup, maximal change and median filters from the control loop Fig. 12 have been replaced with the proposed Kalman filters.

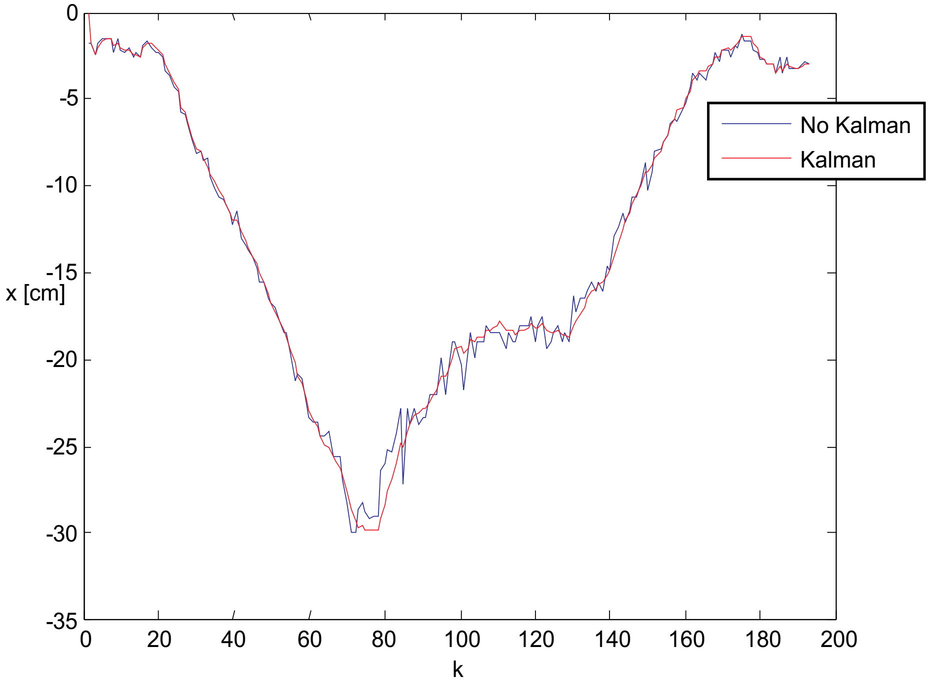


Fig. 16. Horizontal displacement of the tracked agent at distances shown in Fig. 17

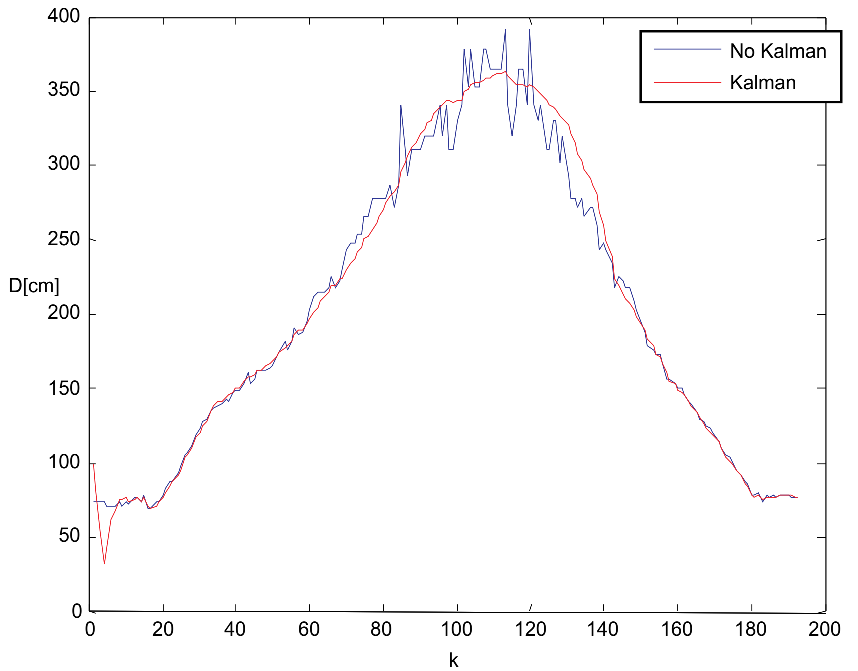


Fig. 17. Distance of the tracked agent

7. Conclusion

This paper presents localization and tracking strategy that is based on an extremely rapid method for visual detection of an object, thus providing a real time solution suitable for the design of multi-agent control schemes. The agents tracking and localization is carried out through five differently trained cascades of classifiers that process images captured by cameras mounted on agents. A training of 5 cascades, described herein, has employed ŞpositiveŤ and ŞnegativeŤ images, i.e. images with and without an object that should be recognized. Once trained, the cascade have been used for classification process (based on Haar-like features), that enables each agent to determine its relative position and orientation with respect to all other agents performing tasks in its field of view. Performance of the proposed method has been demonstrated on a laboratory setup composed of two mobile robot platforms. Experimental results demonstrate that overall accuracy of recognition is very good (in some case even 100%), thus providing accurate calculation of distance between the leader and the follower. Although acceptable, results of calculation of displacement from the center of image are reticent, and can be improved by using model based estimation, i.e. proposed extended Kalman filters. Further improvements can be made with the usage of a higher image resolution camera. Future work will be toward improvement of robustness of the proposed method with respect to various environment conditions (light intensity, shadows, etc).

8. References

Bonin-Font, F., Ortiz, A. & Oliver, G. (2008). Visual naviation for mobile robots: A survey, *Journal of Intellient & Robotic Systems* 53.

Chiem, S. Y. & Cervera, E. (2004). Vision-based robot formations with bezier trajectories, *Proceedings of Intelligent Autonomous Systems* pp. 191Ű–198.

Choi, J. (2006). Real time on-road vehicle detection with optical flows and haar-like feature detector, *a final report of a course CS543 Computer Vision* .

Cruz, D., Mcclintock, J., Perteet, B., Orqueda, O. A. A., Cao, Y. & Fierro, R. (2007). Decentralized cooperative control - a multivehicle platform for research in networked embedded systems, *Control Systems Magazine, IEEE* 27(3): 58–78.
URL: *http://dx.doi.org/10.1109/MCS.2007.365004*

Dani, A. P., Gans, N. & Dixon, W. E. (2009). Position-based visual servo control of leader-follower formation using image-based relative pose and relative velocity estimation, *ACC'09: Proceedings of the 2009 conference on American Control Conference*, IEEE Press, Piscataway, NJ, USA, pp. 5271–5276.

Gans, N.R.; Dani, A. D. W. (2008). Visual servoing to an arbitrary pose with respect to an object given a single known length, *ACC'08: Proceedings of the 2008 conference on American Control Conference*, pp. 1261 – 1267.

Jennings, J., Whelan, G. & Evans, W. (1997). Cooperative search and rescue with a team of mobile robots, *Advanced Robotics, 1997. ICAR '97. Proceedings., 8th International Conference on* pp. 193–200.

Lab, W. (2005). Wifibotsc datasheet, *Hardware Manual* .

Lienhart, R. & Maydt, J. (2002). An extended set of haar-like features for rapid object detection, Vol. 1, pp. I–900–I–903.
URL: *http://dx.doi.org/10.1109/ICIP.2002.1038171*

Mariottini, G. L., Prattichizzo, D. & Daniilidis, K. (2005). Vision-based localization of leader-follower formations, *IEEE Conference on Decision and Control and European Control Conference*, pp. 635–640.

Mariottini, G., Morbidi, F., Prattichizzo, D., Pappas, G. & Daniilidis, K. (2007). Leader-follower formations: Uncalibrated vision-based localization and control.

Matas, J. & Sochman, J. (Oct. 2008). Adaboost.
URL: *http://www.robots.ox.ac.uk/ az/lectures/cv/adaboost_matas*

Mohinder S. Grewal, A. P. A. (2008). Theory and practice using matlab 3rd edition, *A John Wiley & sons, Inc* .

Monteiro, G., Peixoto, P. & Nunes, U. (2006). Vision-based pedestrian detecion using haar-like features, *Robotica* .

Morbidi, F., Mariottini, G. L. & Prattichizzo, D. (2010). Brief paper: Observer design via immersion and invariance for vision-based leader-follower formation control, *Automatica* 46(1): 148–154.

OpenCV On-line documentation (Oct. 2008).
URL: *http://opencvlibrary.sourceforge.net/*

Renaud, P., Cervera, E. & Martiner, P. (2004). Towards a reliable vision-based mobile robot formation control, *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on* 4: 3176–3181.

Tomono, M. (2006). 3-d object map building using dense object models with sift-based recognition features, *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* pp. 1885–1890.

Viola, P. & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features, Vol. 1, pp. I–511–I–518.
URL: *http://dx.doi.org/10.1109/CVPR.2001.990517*

10

How Computer Networks Can Become Smart

Ricardo Bagnasco and Joan Serrat
Universitat Politécnica de Catalunya
Spain

1. Introduction

In the past, computers had a simpler life, few applications had network capabilities and users only wanted to connect to Internet in few specific situations as to check emails, view some mostly plain-text webpage or to play few games with their friends using a "not very wide" bandwidth. But several aspects have changed nowadays, we have better connections, more content available on Internet (the biggest computer network ever known), there are new devices able to connect allowing the presence of more network elements than before and, in addition, the rapid advances in real-time supported applications that are expected, make it difficult to manage them as has been done. Today's network applications require extensive human involvement in management but we need a more independent (smart) network so research community has been developing lot of work in the last few years trying to ease this issue and to introduce autonomic behaviour in computer and telecommunication networks[1]. Manage a network is a complex task where the purpose is to control the behaviour of their elements in order to fulfil some high level goals. The automation of networks and systems thus far has been applied to simplified problems. Typically, we can find some works about improving routing[2]; configuring a network element to better support some application; or controlling the compliance of contracts between peers. Among the different enabling technologies for automate network communications the policy based management is one of the most representatives. This paradigm allows segregating the rules that govern the behaviour of the managed system from the functionality provided by the system itself (Boutaba & Xiao 2002). This is particularly useful in environments where it is necessary to dynamically change goals and add new services or resources. The most well developed implementations of this paradigm rely on pre-programmed rules based on logic. Nevertheless, to be called autonomic, a system must show a degree of flexibility to self adapts to changes and this is hardly achievable by means of static policies. Then, it emerges the need of learning and to figure out the mechanism that will lead to new policies.

Another important aspect of a network is that it has high and low level goals so it would be important to understand the whole vertical structure from sending/receiving bits till more complex application tasks in order to combine and enforce the best action choice. One

---

[1] Our focus is on computer networks but the concepts we mention here are valid for both kind of networks and probably they will merge in a near future.

[2] Routing: how to select paths to send network traffic.

useful methodology to address these challenges is using multi-agent derived frameworks, especially those that acknowledge the existence of other cooperative/competitive agents in the same environment. In this chapter we are going to study the most important characteristics of a network environment and we will show recent approaches which will establish the basis of the future advances we are going to experience in our communications.

There are many different topics and theories around this subject ongoing today so it is quite difficult to present all of them in a comprehensive and coherent manner still covering enough depth. We adopted here a tutorial-like organization where we introduce the concepts, frameworks and general theories and then present topics in the specific domain of computer networks. The remainder of this chapter is organized as follows: first, we explore the dimensions of a network that will affect any proposed solution; then we continue with the modelling techniques and math involved in current approaches to achieve the goal of a self-managed network; later we analyze typical scenarios and most interesting functions that would be important to manage in an autonomic fashion and finally, recognizing that to enable full automation of the networks and systems there are still many unsolved problems and open issues, we are going to mention at the end of this chapter some hot topics waiting for solutions coming from future research.

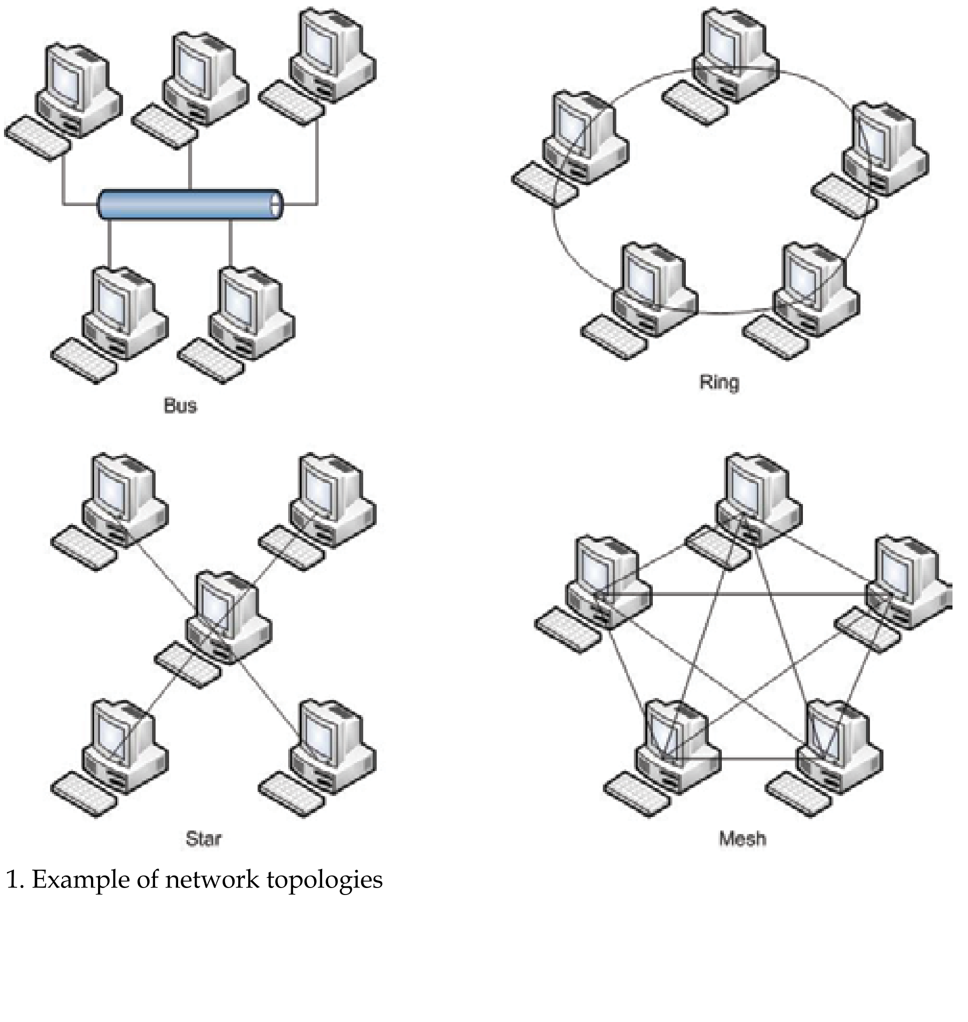


Fig. 1. Example of network topologies

2. Dimensions that affect the solution in a network environment

Besides the functional aspects, there are several features of the network itself that can influence the solution design. We should be aware of the scenario's characteristics in order to choose wisely how to approach the situation. Here we will enumerate some network related orthogonal dimensions that could be used to define the taxonomy of those scenarios.

- Network element characteristics:
  - Internal Resources: memory, processing power (CPU), storage, etc;
  - Relevant external features as screen size if it is a video device;
  - Technologies, protocols and "languages" it can understand.
- General network characteristics:
  - Size: from two to near infinite. It affects the scalability of the algorithm.
  - Topology affects the way agents can communicate (see fig.1):
    - Bus: they all share same line of communication;
    - Ring: they are connected in a circular sequential way;
    - Star: there is one element that concentrate the communications;
    - Mesh: they are all fully connected.
  - Organization: Affects the coordination.
    - Brief history of management in centralized networks:
      To manage a centralized network the Simple Network Management Protocol (SNMP) was developed in response to the growth of Internet devices. It is a protocol for retrieving information, with the use of Management Information Base semantics (MIB[3]) is possible to define what can be communicated. This framework and protocol have been widely implemented in commercial products, and even become the de facto standard for network management. Due to the maturity of SNMP, many device manufacturers include support for this protocol in their products, and many SNMP management solutions exist off-the-shelf for common operating systems and programming languages. However it is still needed to write the management code by control system programmers. This was not considered sophisticated enough to cope with the approach undertaken by the International Standardization Organization (ISO) within the Open Systems Interconnection (OSI) framework. Because of that, the telecommunication community defined a common set of standards which could apply to both worlds. Later these standards were used for telecommunications management network (TMN) defining five functional areas called FCAPS (Fault, Configuration, Accounting, Performance and Security Management). Nowadays they are working on a newer model to replace the aging TMN. The next solution was Common Information Model (CIM) plus eXtensible Markup Language (XML), this solution provides open schema to describe objects and also enables application interoperability without APIs. Within a single organization a hierarchically structured distributed management system can be appropriate. Operators can manage the automated managers effectively. The management system can be partitioned

[3] MIB: it is a database that contains information about each device in a network. It is used by network protocols to control performance and productivity.

functionally with dedicated manager components performing configuration, fault or security management.
As in any organization, the centralized approach is excellent in order to coordinate and to achieve goals.

- Peer-to-Peer (P2P): Decentralizing the network.
  The centralized approach is not so good when we scale up the amount of integrants. The number of emitting nodes producing information can easily exceed a certain critical value making the centralized server unavailable. Moreover if dynamic information is stored it would be important to update a huge database and to manage it properly. The P2P approach adds flexibility, scalability and survival in the presence of some problems that would be not offered by any centralized approach. On the other hand we loose the potential to achieve fast agreements and hinder the coordination in order to fulfill the network goals because now we need to conciliate and interact more than in the previous approach. Depending on the organization we can find a centralized initially and then decentralized P2P protocol with a central server performing some operations at the beginning of the relation (for example: Napster), or a purely decentralized with no central coordination activity (for instance Freenet) or partially centralized with some nodes acting as supernodes with local central indexes (for example Kazaa). The content can be distributed in a structured way, where network topology is tightly controlled and files are placed in precise locations (Chord, CAN, PAST, Pastry) or loosely structured where locations are affected by routing hints but are not completely specified (Freenet) or unstructured where data is distributed randomly over peers and broadcasting mechanisms are used for searching (Napster, KaZaa). Some common problems in P2P environment are: how to discover resources; how to stimulate the sharing; how to select from different sources; how to assure the presence of all the pieces; and how adapt to change in the location distribution where both agents and targets are moving.
- Hybrid: the best approach is probably a hybrid solution combining the advantages of both architectures in accordance to the particular situation involved. P2P may be useful between cooperating distributed managers but a hierarchy can provide a single point of access to the network and allow a better balance between independent coordination of the network elements and the need of administrative control. In summary, P2P is better to survive while centralized is better to achieve goals.

3. Modelling and learning techniques

We can define a good model as the one that allows us to express in an efficient way the characteristics that are important for the functionality we want to offer. There are several theories and models that describe network scenarios but actually they are somehow incompatible so we are still looking for a unified theory (if it exists).
It is a common agreement that solution complexity depends on the adopted modelling, therefore the designer must be careful about choices he makes because they will have a big impact in the future system. That implies he should have a good understanding of both the task and the network environment. Probably the first decision to be made is to choose the

model's scope, some researchers decide to limit it to the network system while others also include the final users and/or the context of execution. The next step is to define the control variables that will be our inputs and which will be the measures to be considered as outputs. The last task is to choose the best technique to find the solution to the situation. Define an architecture is a common way to face the modelling task so there has been several tries to design one that could assure the functionality of the system in a network environment. IBM defined the autonomic computing architecture (IBM, 2001) similar to the agent model from artificial intelligence, with sensors to capture the state, actuators to affect the environment and an inner loop to reason about. The term "autonomic" comes from the autonomic nervous system that acts as the primary conduit of self-regulation and control in human bodies. Later, Motorolla redefined it to some extent to include explicitly the compatibility with legacy devices and applications, creating the FOCALE architecture (Strassner et al., 2006). For the "choose a technique to build a solution" task there are basically two tools: logic and maths. With logic we can express our knowledge about the world and with math we can classify, predict and express a preference between options. In particular two branches of math, utility based (there is a function to maximize) and economical approach (mostly market based and game theory) are common models used by research community.

We are looking for methods that allow learning by improving the performance so we need a measure to quantify the degree of optimality achieved, because of that we are going to focus on mathematical techniques. The simplest expression of a process model is some function giving us an output from the inputs in the form of *f (input_1,… , input_i) = (output_1,… , output_j).* If we have an analytic model for the system we can use it and try to solve the equations. But if we do not know the relationships between inputs and outputs we should select a method that is capable of dealing with uncertainty such as Bayesian networks that learns them. We are interested in the kind of tasks that are so complex that there is no such analytical model or it is hard to solve it so creates the need of new kind of solutions. Learning then becomes a mean to build that function when we lack of tools to solve the maths involved or we have no perfect knowledge about the relationships between variables. Among several possible kind of tasks to be learned (classification, regression, prediction, etc) we will mention the research in several areas but explaining better the planning one. The task to be learned will be how to choose actions in order to follow the best path of functioning during the life cycle of the system.

There are basically three main approaches to learning: supervised, unsupervised and reward-based (also known as reinforcement learning). In supervised learning the critic provides the correct output; in unsupervised learning no feedback is provided at all; in reward-based the critic provides a quality assessment (reward) of the learner's output. The complexity of the interactions of multiple network elements make supervised learning not easy to apply to the problem because they assume a supervisor that can provide the elements with many different samples of correct answers which is not feasible in network scenarios. We are dealing with dynamic environments in several ways; sources of changes can come from variations in network resources, user requirements (including the pattern of user requests), service provision and network operational contexts (Tianfield, 2003). Therefore, the large majority of research papers in this field have used reward-based methods as those modelled by a Markov Decision Process because they have the appealing characteristic of allowing the agents to learn themselves in an unknown dynamic environment directly from its experiences. Other great advantage of this model is that do not need to memorize all the process, it assumes that the consequences of choices only depends on the previous state (not all history, just the previous state matters). In this section we will mention some important characteristics of Markov Process and few variants that make different assumption about the environment. Then we will analyze the dynamics that emerges depending on if we are in a cooperative or competitive environment and we will mention few approaches that already exist to give some insights about the current development stating pros and cons.

3.1 From single-agent to multiagents modelling

If we are in a situation where we ignore the exact dynamic but it is known that it is statistically stationary (probabilities of transition between states do not vary) we can in average learn to act optimally. Borrowing ideas from the stochastic adaptive control theory has showed to be of great help in network management so we will start briefly mentioning some variants of the single-agent approach of Markov based techniques. Generally we can say that value-search algorithms (like q-learning) have been more extensively investigated than policy search ones in the domain of telecommunications.
Modelling the situation as a Markov Chain means that we assume that the probabilities of state transitions are fixed and that we do need to memorize all the past because the transitions only depend on the previous state. The application of that model to processes is called Markov Decision Process (MDP). In its simplest version it assumes perfect information but there are two extensions to MDP, one that it is useful when we can not see all the parameters of the real states but some others related, called Hidden Markov Model. It models the situation where states are not directly observable. And the second variant called Partially Observable Markov Decision Process (POMDP) that allows modelling the uncertainty or error in the perception of the world where states are partially observable.
Returning to MDP, the most well developed solution in machine learning is Reinforcement Learning (RL) where the agent learns by trial-and-error interaction with its dynamic environment (Sutton & Barto, 1998). At each time step, the agent perceives the state of the environment and takes an action which causes the environment transit into a new state. The agent receives a scalar reward signal that evaluates the quality of this transition but not explicit feedback on its performance, the goal is to maximize the reward of the process. In the case of single agent RL there are good algorithms with convergence properties available.
The sparse nature of many networks seems to indicate that multiagent techniques are a good option to design and build complete solutions in these scenarios. A multiagent system (Weiss, 1999) is defined as a group of autonomous, interacting entities sharing a common environment. A specific work on multi-agent systems for automated network management founded in (Lavinal et al., 2006) where it adapts domain specific management models to traditional agency concept and describes precise agent interactions. Not only assign roles to agents (e.g. managed element, manager) but also all agent-to-agent interactions are typed according to their roles and task dependencies. Other basic approach to multiagents learning is to model the situation with a hierarchical technique where the situation is seen as several independent tasks with the particularity that the integration of each best policy is also the global best option. It seems good but when the task is not easily decomposed hierarchical then we need some different approach. The most extended solution in the machine learning literature is called Multiagent Reinforcement Learning (MARL). This modelling is an extension from the single agent reinforcement learning so it does not require exact knowledge of the system and assumes the environment is constantly changing and

hence requires constant adaptation. Also it has the same drawbacks: the learning process may be slow; a large volume of consistent training experience is required; and it may not be able to capture complex multi-variable dependencies in the environment. Furthermore, challenges not presented in the single-agent version appear in MARL like the need of coordination, scalability of algorithms, nonstationarity of the learning problem (because all agents are learning at the same time) and the specification of a learning goal. The last topic is related to the fact that in single agent version we had only one reward but now we could have many involved. If we were in a fully cooperative environment we could add the rewards and maximize the sum. But in a competitive environment we need a different approach and a new kind of goals appears, as to arrive to a Nash equilibrium[4] or convergence to a stationary policy as stability measure. However some concerns have been raised against its usefulness because the link between convergence to equilibrium and performance is unclear; in fact sometimes such equilibrium corresponds to suboptimal team behaviour. To find the right goals in MARL algorithms (convergence, rationality, stability and/or adaptation) is still an open field of research.

The generalization of the MDP framework to the multiagent case is called stochastic game, where it appears the joint action set combining all the individual actions. As it is stated in the survey of MARL techniques done by (Busoniu et al., 2008) the simplest version of a stochastic game is called static (stateless) game where rewards depend only on the joint actions. Mostly analyzed with game theoretic focus, often the scenario is reduced to two competitive agents with two actions in zero-sum[5] or general-sum games in order to limit and control the complexities involved.

The taxonomy of MARL algorithms depends on which dimensions we choose to classify them. We can focus on:

- The degree of cooperation/competition;
- How each agent manage the existence of others (unaware of them, tracking their behaviour, modelling the opponent);
- The need of information, sometimes agents need to observe the actions of other agents and in other algorithms they need to see in addition their rewards;
- Homogeneity of the agent's learning algorithms (the same algorithm must be used by all the agents) vs heterogeneity (other agents can use other learning algorithms);
- The origin of the algorithms (Game Theory; Temporal Difference –RL-; Direct Policy Search) but it is important to notice that there are many approaches that actually mix these techniques.

The coordination is another topic itself, to explicit coordinate agent's policies there are mechanisms based on social conventions, roles and communication that could be used in cooperative or competitive environments. In the case of social conventions and roles, the goal is to restrict the action choices of the agents. Social conventions impose an order between elections of actions. It dictates how the agents should choose their action in a coordination game in order to reach equilibrium. Social conventions assume that an agent can compute all equilibrium in a game before choosing a single one. And, to reduce the size

---

[4] A joint strategy such that each individual strategy is a best response to the others, no agent can benefit by changing its strategy as long as all other agent keep their strategies constant. The idea of using Nash is to avoid the learner being exploited by other agents.

[5] A situation in which a participant's gain or loss is exactly balanced by the losses or gains of the others participants. One individual does better at another's expense.

of action set in order to reduce the expense of the calculation, it is useful to assign roles to agents (some of the actions are deactivated). The idea of roles is to reduce the problem to a game where it is easier to find the equilibrium by means of reducing the size of action sets. But, if we have too many agents we still need a method to reduce the amount of calculation and that is the utility of extra structures as the coordination graph appeared in (Guestrin et al., 2002). It allows the decomposition of a coordination game into several smaller sub-games that are easier to solve. One of the assumptions here is that the global payoff function can be written as a linear combination of many local payoff functions (each involving few agents).

Another important issue in multiagents is communication; we can define direct or indirect communication. Examples of the first include shared blackboards, signalling and message-passing (hard-coded or learned methods). Indirect communication methods involve the implicit transfer of information through modification of the world environment. For example leaving a trail or providing hints through the placement of objects in the environment. Much inspiration here comes from insects' use of pheromones.

In addition, in complex environments it is not realistic to assume complete information then ignorance needs to be taken into account. Ignorance could lead to incomplete or imperfect information. Incomplete information means that the element does not know the goals of the other elements or how much do they value their goals. Collaboration is a possible mechanism to overcome this. Imperfect means that the information about the other's actions is unknown in some respect. The ignorance could manifest in the form of: uncertain information; missing information; or indistinguishable information. Other sources could be measurements with errors or unreliable communication. Bayesian networks and mathematical models as POMDP have been used with some degree of success to try to fix this problem because they explicitly model the ignorance but on the other hand they are difficult to use and do not scale well. Actually it could be even more complicated, because in POMDP there is no good solution for planning with an infinite horizon. The problem of finding optimal policies in POMDP is not trivial, actually is PSPACE-complete (Papadimitriou & Tsitsiklis, 1987) and it becomes NEXP-complete for decentralized POMDPs (Bernstein et al., 2000).

Finding an exact solution with all those problems mentioned above is still infeasible so sometimes it is used some metaheuristic (biological inspired are probably the most well known ones), they are approximate algorithms of the search process (Alba, 2005), this means that they are no guaranteed to find a globally optimum solution and that, given the same search space, they may arrive at a different solution each time are run. But because they work empirically with some decent performance they are of particular interest for networks where the size of the search space over which learning is performed may grow exponentially.

The multi-agent learning area is still in development and each algorithm makes its own assumptions in order to cope with specific options so the solution designer must find the algorithm that fits better whereas the researchers should develop new more advanced mechanisms with less assumptions.

3.2 Cooperative or competitive learning?

The presence of other agents introduces the question: are they friends or enemies? We will focus first on some approaches that assume the cooperation and good behaviour of all the agents (friends) in the system because they are the most developed algorithms. This is not a complete survey but it tries to illustrate the state of current research.

In the extreme of fully cooperative algorithms all the agents have the same reward function and the goal is to maximize it. If a central approach is taken then the task is reduced to a MDP.
In (Tan, 1993) they extend Q-Learning to multi-agent learning using joint state-action values. This approach is very intensive in the communication of states and actions (every step) and do not scale well. A similar approach is to let the agents exchange information like in the Sparse cooperative Q-learning (Kok & Vlassis, 2006). It is a modification of the reinforcement learning approach which allows the components not only learn from environmental feedbacks but also from the experiences of neighbouring components. Global optimization is thus tackled in a distributed manner based on the topology of a coordination graph. Maximization is done by solving simpler local maximizations and aggregating their solutions. Other approach is to restrict each agent to use only the information received from its immediate neighbours to update its estimates of the world state (as a contra it could result in long latency and inconsistent views among agents). In MARL there are basically states, actions and rewards so the approaches differ in what the agents share and what is private (what is social and what is personal). The cost of communications should also be taken into account, a framework to reason about it can be found in (Pynadath & Tambe, 2002) with the name of communicative multiagent team decision problem. Another option to reduce complexity is to try to reduce the universe of possible policies. To achieve it we can use a different level of abstraction that allows to construct near-optimal policies as in (Boutilier & Dearden, 1994) or to use explicitly a task structure to coordinate just at the high level of composed policies as in (Makar et al., 2001).
It is extremely difficult to find some algorithm in the literature which is guaranteed to converge to global optimum even in the reduced case of a fully cooperative stochastic game (one example is Optimal Adaptive Learning in (Wang & Sandholm, 2002)). Because of that, we will repeat that heuristics are still welcome to introduce some guide in the policy search (Bianchi et al., 2007) or to bias the action selection toward actions that are likely to result in good rewards.
If we take a look at the competitive learning models, the interest of agents is now in conflict and we can not assume they will do anything to help each other. Not surprisingly, there is a great influence here of game-theoretic concepts like Nash equilibrium. We are limited by now by the low development of the current techniques and many algorithms are only able to deal with static tasks (repeated, general sum games). In the presence of competing agents we hardly have guaranties in scenarios more complex than two agents with two options. If it were not bad enough, competition can result in cyclic behaviours where agents circle about one another due to non-transitive relationships between agent interactions (like rock-scissor-paper game). And, when scaled up to hundreds of agents in stochastic environments with partially observable states and many available actions, all current existing methods will probably fail. Because of that, it is fundamental to research more in this area where we have many open questions still.

4. Some scenarios and current research

Here we will show some developments that introduce improvements in common network tasks including some interesting approaches, even when some of them are not strictly related to learning and few of them are single agent versions. It could be good to notice that managing a complete network involves several topics so we could find useful related work in different areas of research. Our focus is in communication systems but, for instance in the multi-hardware configuration area and electricity networks we can find analogous problems too.

4.1 Specific challenges of multiagent solutions for network scenarios

The usual scenario for multiagents in machine learning is a group of robots doing some task like playing soccer; grabbing cans; or cleaning some rooms. In section 2 we have mentioned some characteristics and how they can affect the possible algorithms, but there are some extra issues that make network a different and interesting environment to apply multiagents techniques. For instance, communication between agents is an important topic in multiagent solution but in a network there is an extra problem: there will be a trade off between coordination messages vs bandwidth available for the system itself because administrative messages will reduce the available resource. In addition, as computer networks are highly heterogeneous environments with different device capabilities, it could be better suited to develop heterogeneous approaches or if we choose to use homogenous algorithm then to take special measures and abstractions. Privacy also would be of great value in scenarios such of those with presence of different telecommunication carriers while they are collaborating. In addition, some agencies are proposing changes in current network regulation and legislations that would change some of our current assumptions. And last but not least, new business models should be introduced by enterprises to survive. Because of all that, we can only try to imagine the difficulties will be posed to these autonomous network management scenarios. We will next mention some of the existing solutions to illustrate how far we are still to fulfil the mentioned challenges.

4.2 Classic network scenarios

The use of learning techniques has been introduced in the network management field several years ago with some initial scenarios of interest related to fault management and allocation tasks in server. The fault management case has two stages: fault detection and fault mitigation. It is important to detect the kind of fault in order to choose the best way of mitigate it or, if we can not mitigate it then to announce it clearly to the system administrator. One of the difficulties to identify the threat is the big amount of alarms that could be triggered by the same problem, so one important issue in this task is the correlation of alarms. It has been solved usually by classification techniques from the datamining field; neural network or belief networks are some options where some engine might find out the most probable cause for a given sequence of alarms. It can be found in (Steinder & Sethi, 2004) a solution to fault localization using belief networks.

We can also find the use of Reinforcement learning applied to automatic network repair in (Littman et al., 2004) where they study its feasibility and efficiency.

Some low level but critical functions which are worth mentioning are how much data send, when to do it, how to route it and how to control the access of new integrants to the network. In the routing approach of (Boyan & Littman, 1994) Q-routing had better results to non-adaptive techniques based on shortest path; it regulates the trade-off between the number of nodes a packet has to traverse and the possibility of congestion. The problem with routing algorithms is that in general do not scale well, usually make some assumptions to improve their performance or if not take millions of trials to converge on a network of just few nodes. There is a trade-off between proactive and reactive routing protocols, one improves the delay but increase the control overhead or it can decrease the overhead but at the price of not improving as much as before. Although there are many works about learning in routing there is still a constraint in how much time does it takes to converge against the dynamic of the system (the learned path can became inefficient quite fast) so there are several_deterministic techniques with "enough" performance that seems to make unnecessary the overhead of any learned approach in most situations. One of those heuristics is called AntNet (Caro & Dorigo, 1998) and uses swarm intelligence (based on animal behaviour, ants in this case). Their routing scheme adapts to changes in availability and quality of network resources. Each node keeps a routing table with probabilities of reaching known destinations through each outgoing link. Periodically each node sends an agent to a random destination. When the agent returns updates the routing table at each node on the path based on the measured time to reach the destination (latency). A different approach, based on market, could be found in iREX (Yahaya, 2006), an inter-domain scheme for data needing special quality of service on the Internet. Each domain independently advertises and acquires network resources in deploying an inter domain path while decides a price for the links that it owns. The cheapest path with quality assurance is preferred. Another interesting approach, even when there is no learning involved, is the Greedy Perimeter Stateless Routing (GPSR). It is a geographical routing scheme that improves times sending the data to a specific geographical location instead of a destination IP address (Karp & Kung, 2000). This is more efficient because geographical information is more relevant to choose between paths than just IP address.

Some higher level classic problems include task allocation and resource management. They have some similarities because in the end task allocation also is a request for resources. In (Bennani & Menascé, 2005) they propose a solution using analytic queuing network models combined with combinatorial search techniques. They defined a cost function to be optimized as a weighted average of the deviations of response time, throughput and probability of rejection and use predictive multiclass queuing network models, then the global controller algorithm executes a combinatorial search technique over the space of possible configuration vectors where the utility function will guide the choices. The data centre tries to maximize a global utility which is a function of the local utility functions of the various application environments. In the case of resource management tasks chosen measures usual include average job processing times, minimum waiting time for resources, resource usage and fairness in satisfying clients. With an economic approach, the same problem is presented in (Rodrigues & Kowalczyk, 2007) where they propose a price that is adjusted iteratively to find equilibrium between a set of demands and a limited supply of resource (this method was first applied by (Everett, 1963)) they suggest a mechanism to learn the negotiation. In (Abdallah & Lesser, 2006) they create an algorithm that mixes game theory with a gradient ascent learning algorithm for two players with two actions. Its reading is enough to show the hardness on the theoretical proof of this kind of scenarios where any analysis gets too complex too fast.

4.3 Quality of Service

Not all applications running on a network need the same amount and kind of resources to function so, in order to introduce the possibility to express the demand of different services it has been introduced the concept of Quality of Service (QoS) that states the restrictions that are required for a service to work properly. It allows the decoupling of different services and the definition of service oriented architectures where it is possible to choose between providers allowing market based approaches. Going one step further in the concept of quality measures there has been new proposals working with perceived quality instead of the previous model based on low level parameters and it has been coined the concept of Quality of Experience (QoE). For instance in (Ruiz et al., 2004) they introduce a genetic algorithm related to a model of the user-perceived QoS allowing the application to select the new combination of settings which maximizes the user's satisfaction for that particular network conditions. They states that the best for the user could differ if we only focus on the low level parameters, for example there is not a linear relation between the bandwidth and the user-perceived QoS, so the problem becomes now how to model user's perception.

4.4 Service Level Agreement

The negotiation of QoS can be well established between different networks by means of a service level agreement (SLA[6]) so it is worthwhile to study the improvement of this negotiation. For instance (Cosmin et al., 2010) introduces intelligent strategies for contract negotiation based on Bayesian framework.

4.5 Wireless

It is clear we should not limit to wired networks, the research community is also working in wireless sensor and vehicle networks. Those different kind of networks have also introduced some special needs, for instance scheduling data transmission is important because a path may not always be available, it is vital also to know how to construct hop-by-hop connectivity and besides, the network elements have to control the power they consume because they are mostly battery powered so it is a scarce resource. So, in addition to other goals, "survival" is an extra one that needs to be taken explicitly into account. In (Vengerov et al., 2005) they try to solve the power control problem by working in hybrid approaches using fuzzy rules to bring some flexibility in the expressivity of the reinforcement learning.

Historically we have been using the seven layers model from IBM to describe a network from the lower layer of sending/receiving bits to highest application layer, so it is natural to find works like (Legge & Baxendale, 2002) where they design a solution assigning one agent in charge for each layer. Otherwise, a new "wave" proposes that, for wireless scenarios, breaking that modular definition is actually better and some researches have embraced the Cross-Layer point of view (Lee & al., 2007). The idea behind this is that sharing parameters from different layers can increase the efficiency of the algorithms at the price of loosing the modularity of the IBM's model.

The resource management is not a simple task either, in (Shah & Kumar, 2008) they show the trade off between personal vs social interest and propose mechanisms to balance it aligning individual's utility with the global utility. Another related topic can be found in (Wolfson et al., 2004) where they study the dissemination of information about resources and propose an algorithm that attacks this issue. They developed an opportunistic dissemination paradigm, in which a moving object transmits the resources it carries to encountered vehicles and obtains new resources in exchange using economic models to incentive collaboration in the propagation by virtual currency involved.

---

[6]Part of a service contract where the level of service is formally defined.

5. Interesting future research and open questions

At this point we hope it is clear for the reader that finding the best policy with thousand of network elements in a dynamic environment in less than one hour is not possible (with our present techniques) so we need to continue the development with new ideas to solve the current open issues in multi agents learning techniques: how to coordinate, how to scale, and how to manage partial or incomplete information. Some possible options to explore are to define some information that allows us to process the data quicker in order to find a solution in a limited time; or to reduce the complexity and the heterogeneity of the agents. Other approach could be to leave the goal of finding the global optimal policy and use some near optimal concept with a trade-off measure between time to learn and error introduced in the best solution achieved. All that implies that to find a solution to a big problem we need to identify and add meta-information to drive the transformation from the original problem to a less complex task, maybe we could find a way to introduce time restriction in our algorithms. Another path to explore is the fact that existing MARL algorithms often require some additional preconditions to theoretically guarantee convergence. Relaxing these conditions and further improving of various algorithms in this context is an active field of study.

Another issue is that any realistic network will have multiple, potentially competing goals. Then it becomes a multi-objective problem and we need to find members of the Pareto-optimal set, where no single parameter can be improved without a corresponding decrease in the optimality of another parameter (Lau & Wang, 2005). Multi-objective optimization is a well-studied topic with good results for distributed multi-objective genetic algorithms (Cardon et al., 2000). There is still much work to do improving the selection between points of the Pareto frontier and developing more learning algorithms that do not depend only on genetic ones. And, to further complicate things, not only we have multiple goals but they can also be in different levels (or granularity). As it is mention in (Kephart & Das, 2007) an important issue is how to translate from resource level parameters to high level goals such as performance metrics (response times and throughputs) or availability metrics (recovery time or down time). It is key to manage the relationship between those metrics and the control parameters. Their work is interesting and relies on utility functions (although that approach also has some drawbacks at the time to express preferences with more than two parameters). That implies we need to improve the expressiveness of the goal function if we want to continue using utility based approaches. One possible approach is to reduce complexity, maybe less parameters is better than many if we need to reduce the time of learning, so how to choose which are the most important parameters to do the reduction may be another question to research.

Finally, because the great amount of tasks presented in a network environment we need a definition of a common benchmark to test different existing (and future) approaches. If every designer continues defining a new special scenario where his algorithm performs perfect then we will never be able to compare them. We acknowledge that this is not an easy task because the benchmark needs a definition of a "typical" network whereas a network could vary a lot about "typical" characteristics as load and user's request, being a very heterogeneous scenario. If it is not feasible to define only one benchmark we could create few illustrative scenarios with a clear list of assumptions and justification of their existence.

In this chapter we have stated several tasks that are important to manage a network in autonomous fashion; we have collected disparate approaches to multi-agent learning and have linked them to different networks tasks showing current advances and open questions. We believe significant progress can be achieved by more exchange between the fields of machine learning and network management. From one side, new features of future Internet (and networks in general) will introduce more demands and will boost the development of better algorithms so we expect the appearance of new techniques and from the other side the new techniques developed by machine learning field will allow new network functionalities that will improve its utility and our user's experience in the near future.

6. References

Abdallah, S. & Lesser, V. (2006). Learning the task allocation game, *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, AAMAS'06, pp. 850-857, 2006Hakodate, Japan

Alba, E. (2005). Parallel metaheuristics: a new class of algorithms. John Wiley & Sons, Inc., ISBN: 978-0-471-67806-9, NJ, USA

Bennani, M. & Menascé, D. (2005). Resource allocation for autonomic data centers using analytic performance models. *IEEE International conference on autonomic computing*, Seattle, Washington, USA

Bernstein, D.; Zilbertein, S. & Immerman, N. (2000). The complexity of decentralized control of MDPs, *Proceedings of UAI-2000: The sixteenth conference on uncertainty in artificial intelligence*, 2000, California, USA

Bianchi, R.; Ribeiro, C. & Costa, A. (2007). Heuristic selection of actions in multiagent reinforcement learning. *IJCAI'07*, 2007, India

Boutaba, R. & Xiao, J. (2002). Network Management: State of the art. *Communication systems: The state of the art (IFIP World computer congress)*, Kluwer, B.V., pp. 127-146, ISBN:1-4020-7168-X , August 2002, Deventer, The Netherlands

Boutilier, C. & Dearden, R. (1994). Using abstractions for decision-theoretic planning with time constraints", *Proceedings AAAI-94*. 1994, Washington, USA

Boyan, J. & Littman, M. (1994). Packet routing in dynamically changing networks: a reinforcement learning approach. In *Advances in Neural Information Processing Systems,* Vol. 7 (1994) pp. 671-678

Busoniu, L. ; Babuska, R. & De Schutter, B. (2008). A comprehensive survey of multiagent reinforcement learning, *IEEE Transaction on systems, man and cybernetics – Part C: Applications and reviews*, Vol. 38, No. 2 (2008) pp. 156-172

Cardon, A.; Galinho, T. & Vacher, J. (2000). Genetic algorithms using multi-objectives in a multiagent system. *Robotics and Autonomous systems*, Vol. 33, No. 3 (2000) pp. 179-190

Caro, G. & Dorigo, M. (1998). AntNet: distributed stigmergic control for communications networks. *Journal of Artificial Intelligence Research*, 9, pp. 317-365, 1998

Cosmin, G.; Dan Şerban, L. & Marius Litan, C. (2010). A Framework for Building Intelligent SLA Negotiation Strategies under Time Constraints. *Economics of Grids, Clouds, Systems, and Services 2010*, Vol. 6296 (2010) pp. 48-61, DOI: 10.1007/978-3-642-15681-6_4

Everett, H. (1963). Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, Vol. 11, No. 3 (1963) pp. 399-417

Guestrin, C.; Lagoudakis, M. & Parr, R. (2002). Coordinated reinforcement learning, in *Proceedings of the 19th International Conference on Machine Learning (ICML-02),* pp. 227-234, 2002, Sydney, Australia

IBM (2001). Autonomic manifesto, www.research.ibm.com/autonomic/manifesto

Karp, B. & Kung, H. (2000), GPSR: Greedy perimeter stateless routing for wireless networks, *MobiCom2000*, August, 2000, Boston, USA

Kephart, J. & Das, R. (2007). Achieving self-management via utility functions, *IEEE Computer Society*, 2007, Vol. 1, Issue 1 (2007) pp. 40-48, ISSN: 1089-7801

Kok, J. & Vlassis, N. (2006), Collaborative multiagent reinforcement learning by payoff propagation, *Journal of machine learning research 7*, pp. 1789-1828, 2006

Lau, H. & Wang, H. (2005). A multi-agent approach for solving optimization problems involving expensive resources, *ACM Symposium on applied computing*, 2005, New Mexico, USA

Lavinal, E.; Desprats T. & Raynaud Y. (2006). A generic multi-agent conceptual framework towards self-management. *IEEE/IFIP Network Operations and Management Symposium*, 2006

Lee, M. & Marconett, D. & Ye, X. & Ben, S. (2007). Cognitive Network Management with Reinforcement Learning for Wireless Mesh Networks, *IP Operations and Management, Lecture Notes in Computer Science*, Vol. 4786 (2007) pp. 168-179, DOI: 10.1007/978-3-540-75853-2_15, 2007

Legge, D. & Baxendale, P. (2002). An agent-based network management system, *Proceedings of the AISB'02 symposium on Adaptive Agents and Multi-Agent Systems*, 2002, London, England

Littman, M. ; Ravi, N. & Fenson, E. et al. (2004). Reinforcement learning for autonomic network repair. *IEEE 1st International Conference on autonomic Computing*, 2004

Makar, R.; Mahadevan, S. & Ghavamzadeh, M. (2001). Hierarchical multi-agent reinforcement learning. *Atents'01*, 2001, Montréal, Quebec, Canada

Papadimitriou, C. & Tsitsiklis, J. (1987). The complexity of markov decision processes. *Mathematics of Operations Research*, vol. 12, nº. 3 (1987) pp. 441-450, USA

Pynadath, D. & Tambe, M. (2002). The communicative multiagent team decision problem: Analyzing teamwork theories and models, *Journal of Artificial Intelligence Research*, Vol. 16 (2002) pp. 389-423

Rodrigues, E. & Kowalczyk, R. (2007). Learning in Market-based Resource Allocation. *6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)*, 2007, pp.475-482, ISBN: 0-7695-2841-4

Ruiz, P.; Botía, J. & Gómez-Skarmeta, A. (2004). Providing QoS through machine-learning-driven adaptive multimedia applications, *IEEE Transactions on systems, man, and cybernetics – Part B: Cybernetics*, Vol. 34, No. 3 (2004)

Shah, K. & Kumar, M. (2008). Resource management in wireless sensor networks using collective intelligence, *Intelligent Sensors, Sensor Networks and Information Processing, 2008. ISSNIP 2008.* pp. 423-428

Steinder, M. & Sethi, A. (2004) Probabilistic fault localization in communication systems using belief networks, *IEEE/ACM Transactions on Networking*, Vol. 12, No. 5 (2004) pp. 809-822

Strassner, J.; Agoulmine, N. & Lehtihet, E. (2006). FOCALE: A Novel Autonomic Networking Architecture. In *Latin American Autonomic Computing Symposium (LAACS)*, 2006, Campo Grande, MS, Brazil.

Sutton, R. & Barto, A. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press, 1998

Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. *In Proceedings of the Tenth International Conference on Machine Learning*, pp. 330-337, 1993

Tianfield H. (2003). Multi-Agent based autonomic architecture for network management, IEEE, 2003

Vengerov, D.; Bambos, N. & Berenji, H. (2005). A fuzzy reinforcement learning approach to power control in wireless transmitters, *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, pp. 768-778, ISSN: 1083-4419, August, 2005

Wang, X. & Sandholm, T. (2002). Reinforcement learning to play an optimal Nash equilibrium in team Markov games, *in Advances in neural information processing systems*, 2002

Weiss, G. (1999). Ed., *Multiagent Systems: A modern approach to distributed artificial intelligence*. Cambridge, MA: MIT Press, 1999

Wolfson, O.; Xu, B. & Sistla, A. (2004). An economic model for resource exchange n mobile peer to peer networks, *Proceedings of the 16th International conference on scientific and statistical database management*, January, 2004

Yahaya, A. (2006). iREX: Efficient inter-domain QoS policy architecture, *in Proceedings of IEEE Globecom*, 2006

11

Autonomous Decentralized Voltage Profile Control Method in Future Distribution Network using Distributed Generators

Takao Tsuji[1], Tsutomu Oyama[1], Takuhei Hashiguchi[2], Tadahiro Goda[2], Kenji Horiuchi[3], Seiji Tange[3], Takao Shinji[4] and Shinsuke Tsujita[4]

[1]*Faculty of Engineering, Yokohama National University*

[2]*Graduate School of Information Science and Electrical Engineering, Kyushu University*

[3]*Transmission & Distribution Systems Center, Mitsubishi Electric Corp.*

[4]*Smart Energy Technology Center, Tokyo Gas Co.,Ltd.*

Japan

1. Introduction

To realize the sustainable energy society, it is of prime importance to introduce the renewable energy, such as Photovoltaics (PV) and Wind Turbine generators, or Co-generation systems which can utilize the exhaust heat energy. These generators are called "Distributed Generator (DG)" and they are introduced to the distribution network mainly as shown in fig.1. However, in the distribution network with a large number of DGs, the voltage profile maintenance becomes important issue due to the reverse power flow caused by DGs.

Conventionally, the voltage profile is controlled within the allowable range by the use of load-tap-changing substation transformer (LTC) or Static Capacitor (SC) in order to compensate the voltage drop caused by the demand-directional power flow. There are a lot of studies in which the voltage profile is maintained by the effective utilization of those facilities when a large number of DGs are introduced. For example, the optimization technology based on the global information of distribution network is utilized to determine the control actions of voltage control equipments. However, it should be difficult to control the high-speed voltage change by LTC or SC because those equipments work by switching. Although the utilization of SVC (Static Var Compensator) is one of effective approach to realize the high-speed voltage control, it should not be desirable from a viewpoint of increase in cost.

To realize the high-speed and flexible voltage maintenance control with reducing capital investments, it should be effective to utilize the reactive power control of DGs. Supposing that a large number of DGs are introduced, it should be difficult to manage the all information of whole system and to control all DGs. Hence, the much attention is paid to the autonomous decentralized control method, for example, by P. N. Vovos or P. M. S. Carvalho. However, the cooperative work among multiple DGs is not considered in their papers because only the information of the connection node is utilized. Although Mesut E. Baran et al studied the cooperative work among DGs based on the multi-agent system, the

method is not autonomous decentralized one because the specific control signal is generated by the optimization based on the global information.
On the other hand, the authors have developed autonomous decentralized voltage control method so far using reactive power control of DGs. Specifically, voltage profile maintenance is realized by the reactive power control of inverter based on the multi-agent system. In those papers, it is supposed the agent program is installed to each DG. The agents determine their proper control actions based on the local information exchange among neighboring agents. Where, the feedback control based on integral logic is applied. The proposed method is composed by three control methods whose control purposes are different. It is possible not only to maintain the voltage profile but also to decrease the excessive reactive power. Additionally, in our other paper, the proposed method is enhanced to realize the effective utilization of free capacity.
In this chapter, we will describe the outline of the voltage profile control method proposed in our previous work. First, the concept of the reactive power control of DG and voltage profile control method are described in section 2. Next, the basic and enhanced method of autonomous decentralized control are shown in section 3 and 4, respectively. Finally, a conclusion is provided in section 5.

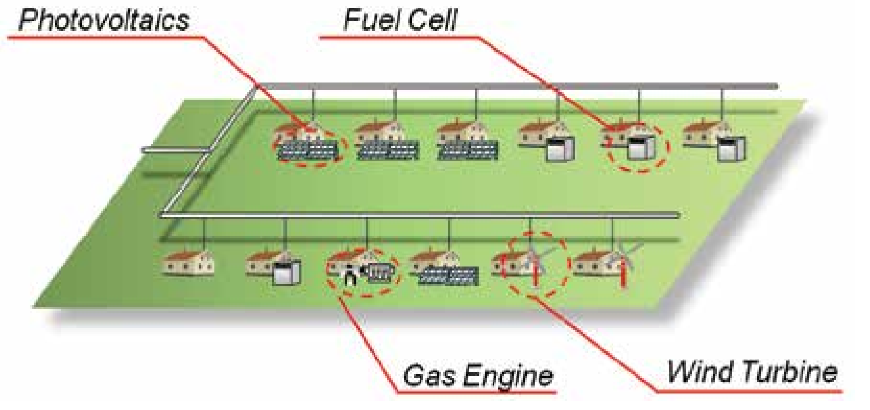


Fig. 1. Distribution network with distributed generators.

2. Concept of voltage profile control method

2.1 Reactive power control of inverter

There are various types of DGs. AC generator such as Gas Engine is connected to a distribution network directly and it can control its reactive power. On the other hand, the inverter is required as for the DC generator such as PV or Fuel Cell and so on. The self-excited inverter can control its reactive power in the case it has a "free capacity". In this paper, the voltage profile control method is developed supposing a large number of inverter-connected DGs are introduced to the distribution network.
If a DG is connected to the distribution system with a self-excited inverter, it can change its power factor. At this time, the inverter capacity, active power output and reactive power output must satisfy the following equation.

$$\sqrt{P_{DGi}^2 + Q_{DGi}^2} \le S_{INVi} \tag{1}$$

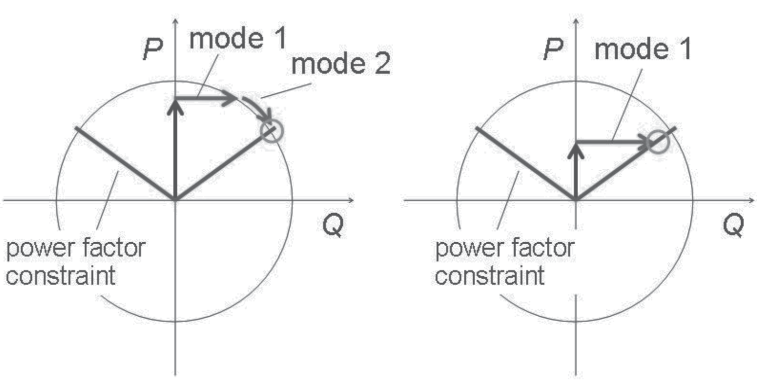
P_{DGi}, Q_{DGi} : active and reactive power output of DG i
S_{INVi} : inverter capacity of DG i
It is supposed the active power output is determined by DG owner because the DG owner control their DGs so freely in the framework of electricity liberalization. In the case the left part is smaller than the right part, it is defined that the inverter has a "free capacity". When the inverter has a free capacity, it can control its reactive power without decreasing its active power. Oppositely, when the inverter does not have a free capacity, it is required to decrease the active power to increase the reactive power. The reactive power control is classified into following two modes according to the free capacity.

mode 1: reactive power control without decreasing its active power when the inverter has a free capacity.

mode 2: reactive power control with decreasing its active power when the inverter has no free capacity.

The concept of the both control modes are shown in fig.2. Because the power factor constraint is considered in this study, there are two cases as shown in fig.2(a) and (b) according to the amount of active power output. When the active power output is large enough, the reactive power control is classified into two phases as shown in fig.2(a). First, only the reactive power is controlled using free capacity (mode 1). After the operating point reaches the inverter capacity constraint, the reactive power is increased with decreasing its active power (mode 2). On the other hand, when the active power output is not large, only mode 1 has a control capacity because lower limit of power factor becomes dominant constraint as shown in fig.2(b).

The power factor constraint is set up in order to avoid that adverse affect is caused by the excessive reactive power control by DGs. Therefore, in the case we have an assumption that the reactive power control of DG is effectively utilized for voltage profile maintenance, the power factor constraint does not seem to be required. Although the power factor constraint is considered in this study according to the conventional system requirement in Japan, the revalidation will be required in the future work.



(a) active power : large (b) active power : small

Fig. 2. Control mode.

The control capacities of both control modes are defined as follows according to the active power output as shown in fig.3. The dominant constraint to determine the control capacity of mode 1 depends on the amount of active power. In the case the active power is larger than the point "a" in fig.3, the capacity of mode 1 and mode 2 are determined by the inverter capacity and the power factor constraint, respectively. In the case the active power is smaller than the point "a", the capacity of mode 1 is determined by the power factor constraint and the mode 2 control is not available.

As above, the control capacities of both modes and the active power changes associated with the reactive power control are described as Eq.(2)-(5). In addition, in the case that the control capacity of either mode becomes zero, the capacity of the mode is defined as a sufficient small value, ε, because a problem is caused in the work of Q-Coop method which is described in section 3.2

- *if* $P_{original,i} > S_{INVi}\cos\varphi_{\max}$ (section A)

$$\begin{cases} Q_{\max 1,i} = \sqrt{S_{INVi}^2 - P_{original,i}^2} \\ Q_{\max 2,i} = S_{INVi}\sin\varphi_{\max} - Q_{\max 1,i} \end{cases} \tag{2}$$

$$P_{DGi} = \begin{cases} P_{original,i} & (-Q_{\max 1,i} < Q_{DGi} < Q_{\max 1,i}) \\ \sqrt{S_{INVi}^2 - Q_{DGi}^2} & (Q_{DGi} < -Q_{\max 1,i} \ or \ Q_{\max 1,i} < Q_{DGi}) \end{cases} \tag{3}$$

- *else if* $P_{original,i} < S_{INVi}\cos\varphi_{\max}$ (section B)

$$\begin{cases} Q_{\max 1,i} = P_{original,i} \times \tan\varphi_{\max} \\ Q_{\max 2,i} = 0 \end{cases} \tag{4}$$

$$P_{DGi} = P_{original,i} \tag{5}$$

$P_{original,i}$: active power of DG i determined by DG owner originally
$\varphi_{\max}$: upper limit of power factor angle [rad]
$Q_{\max1,i}$: mode 1 capacity of DG i
$Q_{\max2,i}$: mode 2 capacity of DG i

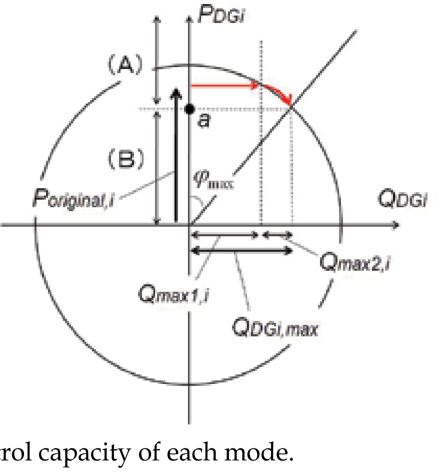


Fig. 3. Reactive power control capacity of each mode.

2.2 Voltage profile control method using inverter

In the proposed method, power factor control of all DGs are defined as control objects. However, since the DGs are demand side facilities, customers have no responsibilities to maintain system voltage. So, it is assumed that proper incentives are given to DG owners to realize the proposed method.

Although the inverter control is effective, the reactive power control of one or two inverter is not enough to realize the proper voltage maintenance. Hence, it is desirable that multiple DGs work as a group cooperatively. In order to realize such a cooperative work, a centralized or autonomous decentralized approach should be available. The concept of those approaches are described below.

(a) centralized control method

The ideal cooperative control should be achieved by determining the control signal based on the optimization using all information of whole system. Because the reduction of active power output is required to maintain the voltage profile, the maximization of total active power generated by DGs is treated as an object function. Specific formulation is as follows.

<object function>

$$\text{Maximize} \quad \sum_{i=1}^{m} P_{DGi} \tag{6}$$

<constraints>

$$P_{DGi} - P_{Li} - V_i \sum_{k=i}^{n} V_k \left\{ G_{ik} \cos(\theta_i - \theta_k) + B_{ik} \sin(\theta_i - \theta_k) \right\} = 0 \tag{7}$$

$$Q_{DGi} - Q_{Li} - V_i \sum_{k=i}^{n} V_k \left\{ G_{ik} \sin(\theta_i - \theta_k) - B_{ik} \cos(\theta_i - \theta_k) \right\} = 0 \tag{8}$$

$$\cos\varphi_{\max} \le \frac{P_{DGi}}{\sqrt{P_{DGi}^2 + Q_{DGi}^2}} \tag{9}$$

$$V_{\min} \le V_i \le V_{\max} \tag{10}$$

$$\sqrt{P_{DGi}^2 + Q_{DGi}^2} \le S_{INVi} \tag{11}$$

m : total number of DGs
n : total number of nodes
P_{Li}, Q_{Li} : active and reactive power of load i
V_i, θ_i : voltage and phase angle of node i
G_{ik}, B_{ik} : conductance and susceptance of branch i-k
$V_{\min}$, $V_{\max}$: lower and upper limit of voltage

Theoretically, the optimal control efficiency is realized using the centralized control method. However, there are following issues.

- The information and communication infrastructures should be required to gather all information of whole system.

- It is difficult to compensate the high-speed change of load and DG output because a long time is necessary to calculate the optimization.

(b) autonomous decentralized control method

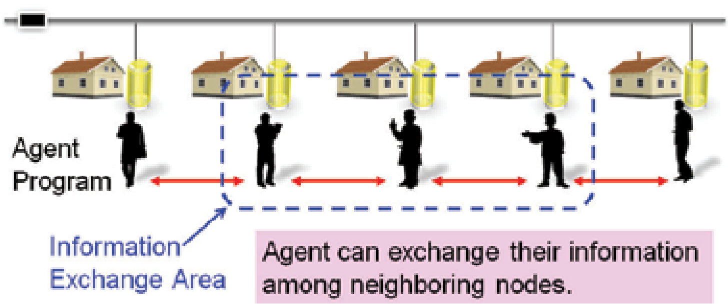


Fig. 4. Concept of multi-agent system in distribution network.

In this paper, we try to develop a voltage profile control method from a viewpoint of autonomous decentralized approach because it should be difficult to realize the centralized control method as described in the previous subsection. Distribution system operator installs the agent program into each DG as shown in fig.4. The agent sends a control signal to its DG in place of the distribution system operator. The agent of each DG determines the control action of its DG autonomously based on the local information exchange.
The class and amount of available information are extremely important for autonomous decentralized control. In this paper, we define the area where the agent can exchange information as "Information Exchange Area (IEA)". IEA is represented by a node number on system model. In the proposed method, IEA is defined as 1 in order to develop a control method which can work even in the case only the local information is available. Which means that each DG is controlled using information of self-node and neighboring nodes. Time delay related to communication is ignored and it is supposed that real-time communication is available. The detailed explanation will be provided after the next section.

3. Autonomous decentralized voltage profile control method -Basic method-

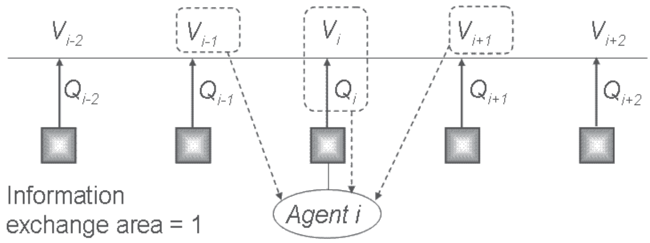
3.1 Concept of basic control method

The basic method of autonomous decentralized voltage control is described in this section. To simplify the problem, only the control mode 2 is treated supposing all DG has no free capacity. Basic control method is composed by following three methods: "Voltage reference method (V-Ref method)", "Reactive Power Saving method (Q-Save method)"and "Reactive Power Cooperation method (Q-Coop method)". In the V-Ref and Q-Save method, each agent requires the information about voltage profile within IEA and the amount of reactive power of its DG as shown in fig.5(a). On the other hand, each agent needs the information about the amount of reactive power of DGs within IEA additionally as shown in fig.5(b). The detail of each method is described as follows.

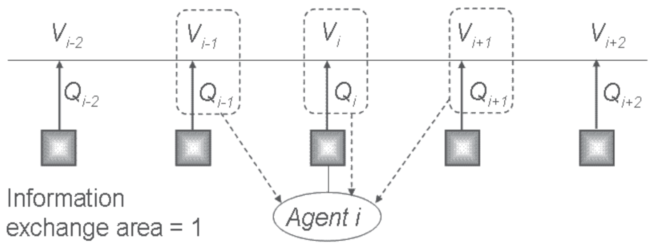
3.2 The combination of three control methods

(a) Voltage Reference method (V-Ref method)

Each agent gathers the voltage information of self-node and neighboring (Fig.5 (a)). When the voltage at any node within IEA deviates from the proper range, each agent coordinates its



(a) Information exchange of V-Ref and Q-Save method



(b) Information exchange of Q-Coop method

Fig. 5. The definition of information exchange area.

reactive power output in order to reduce the total amount of voltage deviation from reference value according to Eq.(11). Although it is important to control voltage profile close to reference value, we assume that the necessity of voltage control is lower within the proper range. Which means that the control object is achieved when the voltage profile is maintained within the proper range at least. Therefore, this V-Ref method does not work when the voltage profile within IEA is maintained properly. In addition, (t) is added to the time-variant variables and it is supposed that the real-time communication is realized without considering the time-delay.

$$\text{if} \quad V_k(t) < V_{lo} \quad \text{or} \quad V_{hi} < V_k(t)$$

$$T_\alpha \dot{Q}_i(t) = K_\alpha (V_{ref,i} - V_k(t)) \tag{11}$$

where,

V_{lo}, V_{hi} : lower and upper limit of voltage constraint (proper range)

T_α : time constant of V-Ref method [sec] , K_α : control gain of V-Ref method

$V_{ref,i}$: reference value of voltage control at node *i*

(it is defined as 1.0[p.u.] at all nodes in this paper)

$V_k(t)$: voltage at node *k* where the voltage deviation value becomes maximum within IEA

(b) Reactive Power Saving method (Q-Save method)

When the voltage profile within IEA does not deviate from the proper range, it is desirable that the reduction of power factor is recovered by saving the excessive reactive power output. Such a control is realized using Eq.(12), where $Q_{ref,i}$ is the reference value of reactive power output and it is set to 0 (p.f.=1.0) usually. Because the priority of this control method is lower than Q-Coop method, the dead band Δ is set as shown in fig.6. When the reduction of reactive power causes the voltage deviation again, Q-Save method should be stopped at the time. Such a control is realized with combination of three methods as described later. The concept of information exchange of this method is shown in fig.5(a).

$$\text{if} \quad V_{lo} + \Delta < V_k(t) < V_{hi} - \Delta$$

$$T_\beta \dot{Q}_i(t) = K_\beta (Q_{ref,i} - Q_i(t)) \tag{12}$$

where,
T_β : time constant of Q-Save method [sec]，K_β : control gain of Q-Save method
$Q_{ref,i}$: reference value of Q-Save method at DG i (in this paper, 0.00[p.u.])

(c) Reactive Power Cooperation method (Q-Coop method)

In the autonomous decentralized method, the information about the voltage deviation is not detected from the agents at distant nodes over IEA. Therefore, we propose Q-Coop method which does not use voltage information directly. In this method, the power factors of DGs are controlled to be equalized within IEA. The concept of information exchange is shown in fig.5(b) and this method is formulated as Eq.(13)-(15).

$$\text{if} \quad V_{lo} < V_k(t) < V_{hi}$$

$$T_\gamma \dot{Q}_i(t) = K_\gamma (Q^*_{ref,i}(t) - Q_i(t)) \tag{13}$$

$$Q^*_{ref,i}(t) = S_i \sin\theta^*_{ref,i}(t) \tag{14}$$

$$\sin\theta^*_{ref,i}(t) = \frac{\sum \sin\theta_j(t)}{M_i + 1} \tag{15}$$

where,
T_γ : time constant of Q-Coop method [sec]，K_γ : control gain of Q-Coop method
$\theta_j(t)$: power factor angle of DG j
$Q^*_{ref,i}(t)$: reference value of Q-Coop method of DG i
M_i : total node number included in IEA of DG i
S_i : inverter capacity of DG i
$\theta^*_{ref,i}(t)$: an average value of θ_i within the IEA.
Following merits are obtained by local equalization of the power factors.

- The chain of Control Action

When a certain DG increases its reactive power output, other DGs which locate on neighborhood also increase their outputs in order to equalize their power factors. This control action leads to a chain reaction. As a result, the distant DGs also increase their reactive power outputs. From the standpoint of whole system, which seems that the distant DGs help the voltage control of distant area by changing their reactive power outputs. This control action is interpreted as a cooperative control.

- Equalization of Power Factor

When a control load gathers to a certain DG, the power factor of the DG decreases greatly. Lower the power factor decreases, the more active power of the DG decreases to generate the same amount of reactive power because reactive power (Q) and active power (P) must satisfy Eq.(1). At this time, generating efficiency greatly decreases. Therefore, the economic efficiency will be improved with an equalization of the reactive power outputs.

(d) Operating condition of each control method

Each DG switches its control method using voltage information within IEA, as shown in fig.6. When there exists a voltage deviation node within IEA, V-Ref method works to improve the voltage profile directly. V-Ref method stops and both Q-Save and Q-Coop methods work when the voltages of all nodes within IEA are improved to the proper range. At this time, the dead zone is set not to cause a chattering between V-Ref and Q-Save method. We can see from the same figure that the control area of Q-Save method and Q-Coop method is almost the same. Then, we encounter the difficulties that the proper control method must be selected depending on the situation. To decide the proper control method is not easy from a viewpoint of autonomous decentralized control. In the proposed method, both control methods are executed simultaneously and a flexible control is realized according to the following logic.

When we apply the both Q-Save and Q-Coop methods, the proper setting of control parameters is needed. Q-Save method works to decrease the reactive power output to improve the efficiency, and Q-Coop method works to increase the reactive power output to improve the voltage profile. The stationary voltage profile is determined by control gain of both methods. The mechanism is represented as fig.7 when both methods are applied. Assume that all voltages are within a target range as shown in fig.7(a), the difference of reactive power outputs between DG i and j disappears due to the work of Q-Coop method first. Once reactive power outputs are equalized, only the control effect of Q-Save method remains and reactive power outputs decrease gradually. On the other hand, when there exists a voltage deviation node, V-Ref method works preferentially at the node. As shown in fig.7(b), the reactive power output of the DG is stack to a maximum value, and reactive power output of next node is determined to a value where the product of the error between the reactive power and its reference value and Q-Save gain is equal to that of the error and Q-Coop gain. Therefore, if K_γ is large enough, the effect of Q-Coop method is also large. If voltage profile is within the proper range, Q-Save method works effectively without

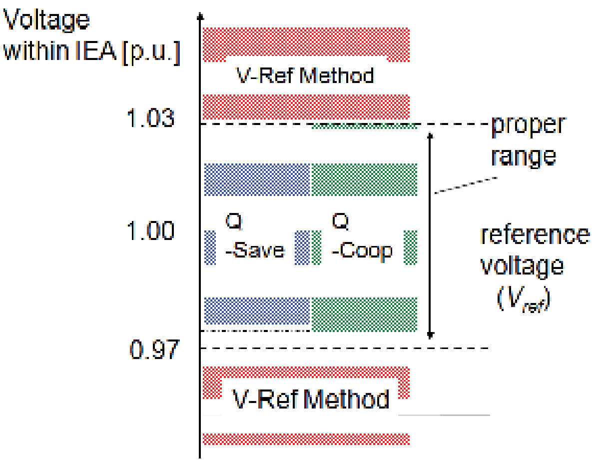
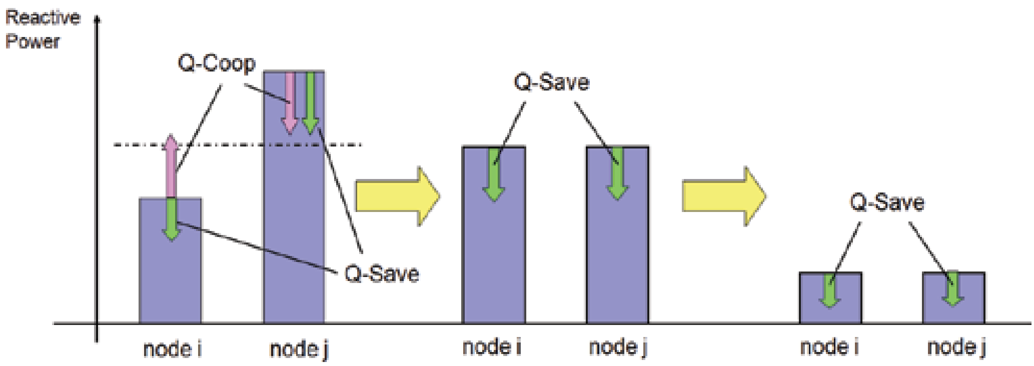
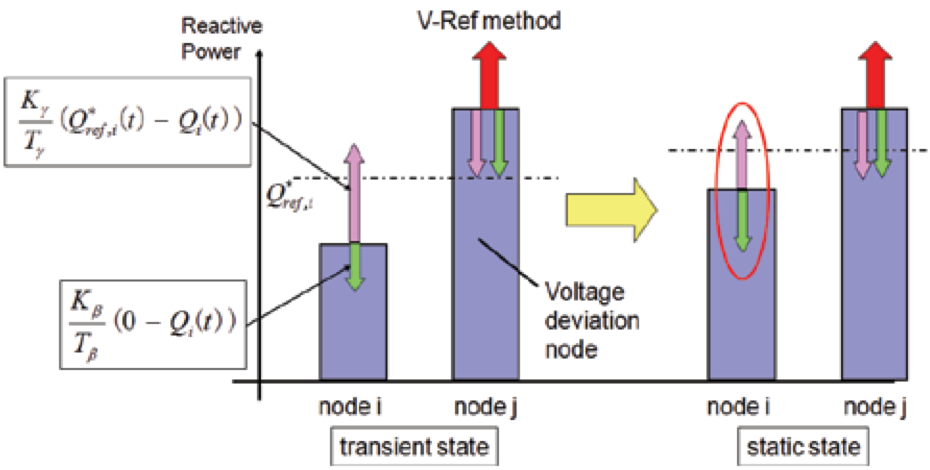


Fig. 6. Operating conditions of each method.

relation to the Q-Coop gain since Q-Save method starts to work after the equalization control has finished. Hence, the large value of K_γ does not influence to the performance of Q-Save method.
Thus, K_γ should be set to a large number. However, the gain constant must be set carefully since too large gain causes an unstable control when the time-delay is considered.



(a) In the case the voltage profile is within the target range



(b) In the case there exist a voltage deviation node

Fig. 7. Cooperative work of Q saving method and Q cooperation method.

3.3 Simulation results by 4-node model system

(a) Model system

The proposed method is tested in 4-node model system shown in fig.8. Both DGs and loads are connected to all nodes and the load changes discontinuously as shown in fig.9. Before time=0 [sec], all loads are light and the voltage profile is within the proper range without using reactive power control of DGs. The control parameters are determined by trial and error according to following guidelines as shown in table 1.

- K_α is set to 1.00 because V-Ref method is defined as a basic control.

- The priority of Q-Save method is low because its main purpose is the improvement of economical efficiency. Therefore, K_β is set to half of K_α (0.50) in order that it works gradually.
- K_γ is set to 8.00 because it must be sufficient large value compared with K_β.
- Supposing that the gradual control, time constants of all control methods are set to 4.00 [sec].

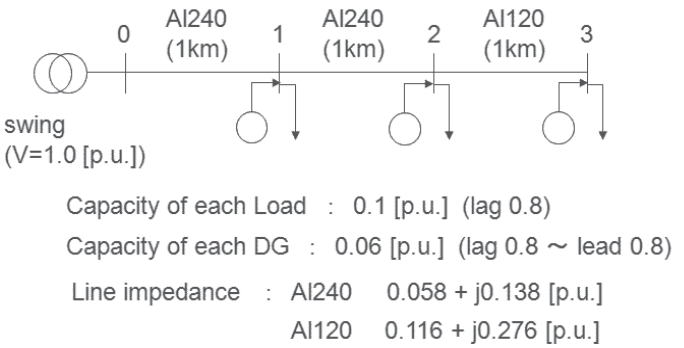


Fig. 8. 4-node model system.

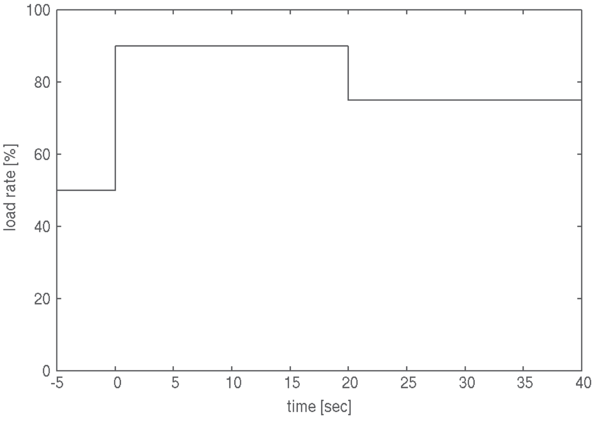


Fig. 9. Load change pattern.

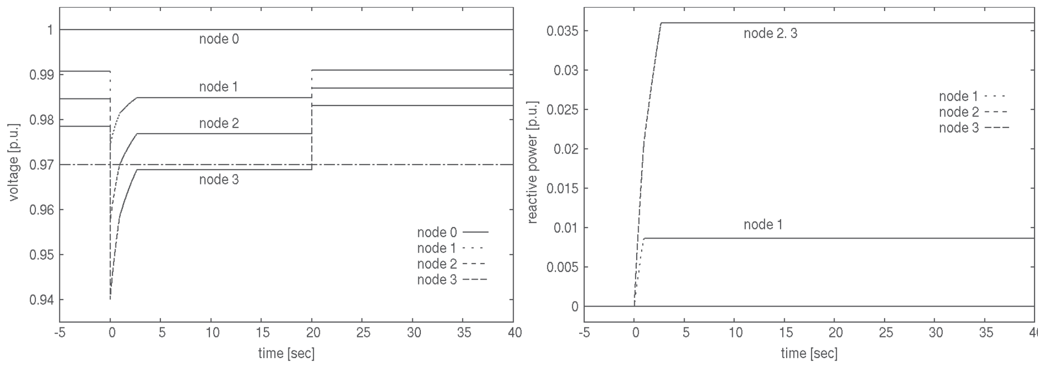
T_α	4.00 [sec]	T_β	4.00 [sec]	T_γ	4.00 [sec]
K_α	1.00	K_β	0.50	K_γ	8.00

Table 1. Control parameters.

(b) Simulation Results

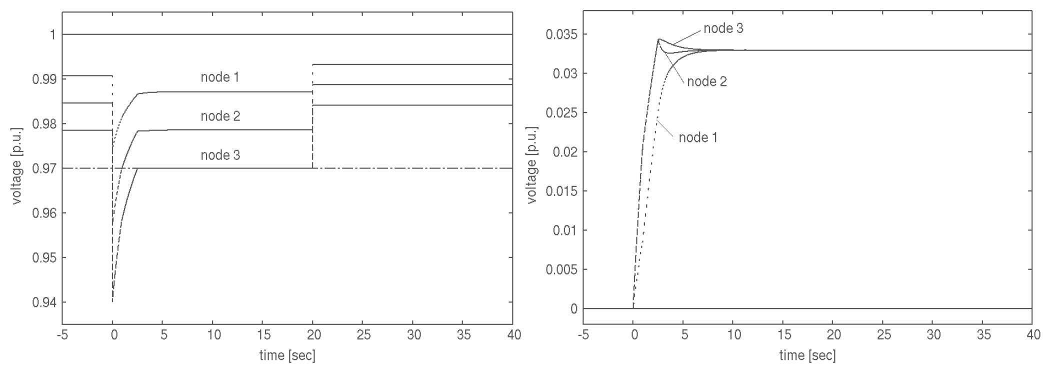
Figure 10 shows the simulation results in the case only V-Ref method is applied. At this time, each DG works to improve the voltage profile only using local information about voltage within IEA. We can see from these results that there are two problems as below.

- voltage at node 3 is not within the proper range.
- after time=20 [sec], the power factor of each DG remains to be decreased.



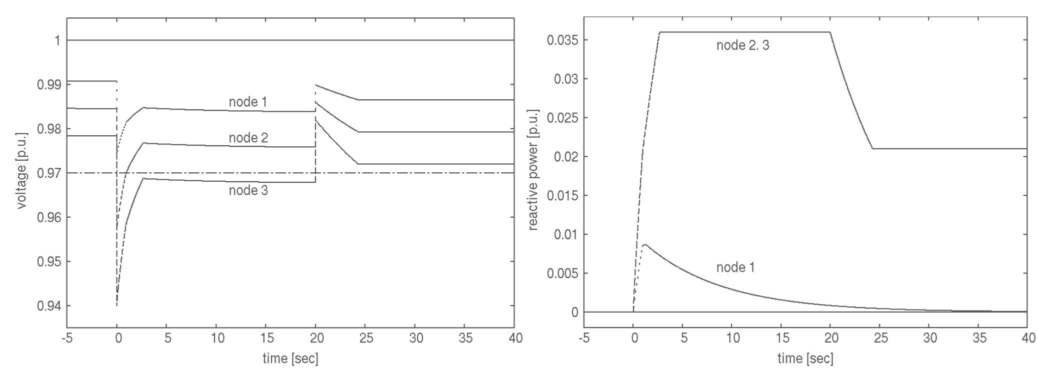
(a) Voltage change of each node (b) Reactive power change of each DG

Fig. 10. V-Ref method.



(a) Voltage change of each node (b) Reactive power change of each DG

Fig. 11. V-Ref method and Q-Coop method.



(a) Voltage change of each node (b) Reactive power change of each DG

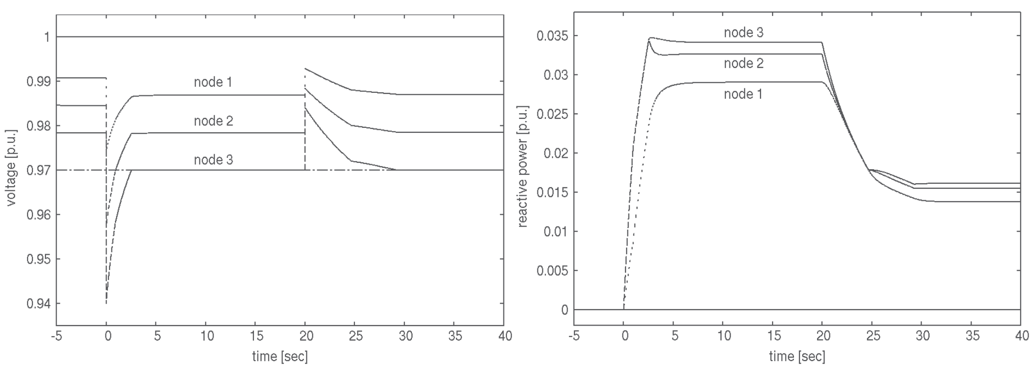
Fig. 12. V-Ref method and Q-Save method.

The first problem is the result from that the reactive power control of DG2 and DG3 which can observe the voltage at node 3 directly reaches the maximum value. At this time, it is possible to improve the voltage at node 3 if the reactive power control of DG1 is available.

However, it is difficult to utilize the reactive power control of DG1 because the voltage profile at node 1 and node 2 which DG1 can observe directly remains to be within the proper range. If it is assumed that the information about the reactive power control is exchanged among agents, the agent of DG1 can estimate the voltage deviation occurs at any node because the reactive power of DG2 increases. Figure 11 shows the simulation results with V-Ref and Q-Coop methods. We can see the reactive power of all DGs are equalized at the stationary point due to the work of Q-Coop method and the voltage profile is within the proper range by the cooperative work.

The second problem is the result from that V-Ref method does not work when the voltage profile of all nodes are within the proper range because the dead zone of V-Ref method corresponds with the proper range. Therefore, as a countermeasure of this problem, the simulation result with V-Ref and Q-Save method is shown in fig.12. Since the DGs whose local voltage are within the proper range change their control methods to Q-Save method, the reactive power of each DG decreases after time=20 [sec] before the voltage profile reaches the lower limit, including the dead band, again.

Finally, figure 13 shows the simulation results with three control methods. Both two problems are solved at the same time by the cooperative work of Q-Save and Q-Coop methods described in fig.7. Before time=20 [sec], Q-Coop method works dominantly because the voltage at node 3 reaches the lower limit of proper range. Oppositely, after time=20 [sec], reactive power output of all DGs are equalized immediately by the work of Q-Coop method, and after that, the reactive power output decreases due to the work of Q-Save method.



(a) Voltage change of each node (b) Reactive power change of each DG

Fig. 13. The application of all methods.

Table 2 shows the comparison of the control effects. Table 2(a) and (b) show the comparisons related to "all time period" and "steady solution" based on Eq.(16) and (17), respectively.

$$\int_0^T \sum_i^m Q_{DGi}^2(t)dt \tag{16}$$

$$\sum_{i=1}^m Q_{DGi}^2 \tag{17}$$

T : simulation time [sec]

The control effect is compared with "the control effect without Q-Save method". In table 2(b), it is also compared with "optimal solution". Although the formulation of this optimization is according to the section 2.2, the object function is replaced by Eq.(17). The utilization of Eq.(17) seems to be almost same as Eq.(6) because only the mode 2 control is treated in this section and the voltage – reactive power sensitivity is larger than voltage – active power sensitivity. From table 2(a), we can see the total amount of reactive power is reduced by about 39%. On the other hand, from table 2(b), the difference between the proposed method and optimal solution is approximately from 7 to 19%.
Additionally, the proposed method does not need the information about "voltage –reactive power sensitivity" which is often utilized for voltage profile control in conventional control manner. Hence, it is possible to apply the proposed method to the 4-node model system without considering the difference of distribution lines between node 2 and 3.

	without Q-Save method	with Q-Save method
Eq.(16)	0.1250 (100%)	0.07645 (61.2%)

(a) Comparison of all time period.

	without Q-Save method	with Q-Save method	optimal solution
Eq.(17) t=15[sec]	0.00326 (114%)	0.00308 (107%)	0.00287 (100%)
Eq.(17) t=35[sec]	0.00326 (562%)	0.000689 (119%)	0.000580 (100%)

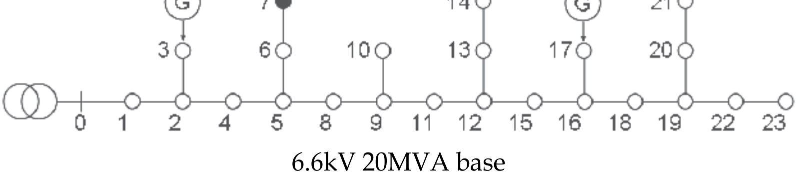
(b) Comparison of steady solution.

Table 2. Comparative table of economical efficiency (4-node model system).

3.4 Simulation results by 24-node model system

(a) Model system

The proposed method is tested using 24-node model system shown in fig.14 to test its effectiveness in the large scale model system. DG whose capacity is small is connected to each node. Although the large capacity DG is connected to node 17, it is uncontrollable and its active power output changes as shown in fig.15. Voltage increases largely at the downstream side of the feeder when the active power output of DG17 increases. However, at that time, voltage drop at the upstream side also becomes problem because the large capacity load is also connected to node 7. Therefore, it is required to maintain voltage profile considering both upper and lower constraints. On the other hand, voltage drop becomes problem in whole system when the active power of DG17 is small.



6.6kV 20MVA base

Fig. 14. 24-node model system.

The detailed settings of model system are listed in table 3. Control parameters are same as table 1 and K_γ is modified to 48 to enhance the work of Q-Coop method.

Demand of each node node 7 Others	 0.075[p.u.] (p.f.lag 0.80) 0.02[p.u.] (p.f. lag 0.80)
DG capacity of each node Additional Generator : node 3 node 17	0.02[p.u.] 0.08[p.u.](p.f.lead 0.80) variable (p.f. lead 0.90)
Power factor control range of controllable DG	lag 0.8～lead 0.8
Proper range of voltage	0.97 ～ 1.03 [p.u]
Line impedance	Al240 (400m/1section) 0.0232+j0.0552 [p.u.]

Table 3. System constant of 24-node model system.

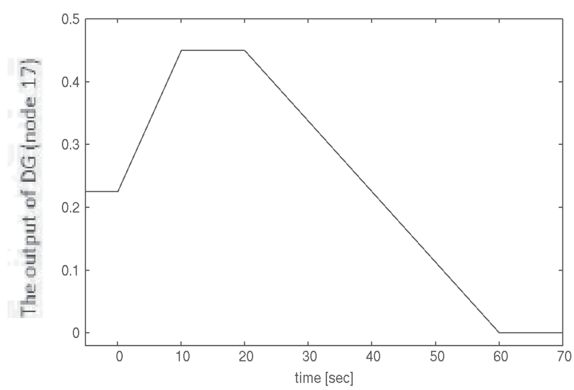
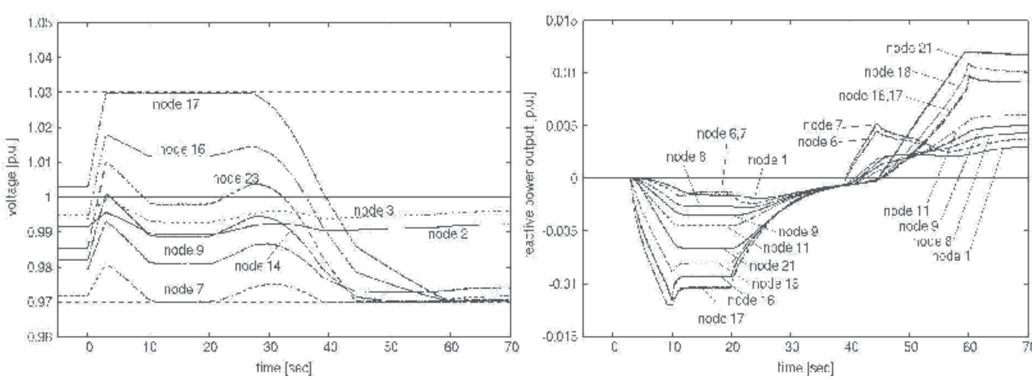


Fig. 15. The active power change of DG (node 17).

(b) Simulation results



(a) Voltage changes of representative points (b) Reactive power changes of DGs

Fig. 16. Simulation results by proposed method.

The change of voltage profile and reactive power in the case the proposed method is applied are shown in fig.16. By the active power change of DG17 from time=0 [sec], the voltage profile of whole system reaches the upper limit during about 30 seconds. We can see from the figure that the proposed method works well and voltage profile is maintained within the proper range due to the reactive power control of multiple DGs. From time=30 [sec] to 40 [sec], voltage profile is within the proper range originally and each DG decreases its reactive power to improve the power factor. After time=40 [sec], voltage profile is also maintained by the reactive power supply although the voltage profile of whole system becomes lower.
Control effects are compared in Table 4 as shown in the case of 4-node model system. From table 4(a), we can see the evaluation index is improved by approximately 60 percent using Q-Save method. This is due to the difference mainly from 30 [sec] to 40 [sec] when the voltage deviation does not occur. From table 4(b), we can see the proposed method with Q-Save method provides the comparable results with the optimal solution.

	without Q-Save method	With Q-Save method
Eq.(16)	0.08352 (100%)	0.03302 (39.5%)

(a) Comparison of all time period.

	without Q-Save method	with Q-Save method	optimal solution
Eq.(17) t=60[sec]	0.003888 (295%)	0.001331 (101%)	0.001319 (100%)

(b) Comparison of steady solution.

Table 4. Comparative table of economical efficiency (24-node model system).

4. Autonomous decentralized voltage profile control method - Advanced Method -

4.1 Concept of preferential utilization of free capacity

In section 3, it is supposed that inverter has no free capacity. However, it is expected DG has a free capacity except at the peak time because the inverter capacity is determined by the maximum active power output. Reactive power control using the free capacity, which is called “mode 1”, does not need the active power decrease. Hence, in this section, mode 1 based on the free capacity of inverter is defined to be “high priority control”. The proposed autonomous decentralized control method is enhanced in this section so that it can utilize the high priority control effectively. The control theory in this section should be applied to both the inverter control and SVC control. When the both equipments are included in the distribution network model, the proposed method works to realize their cooperative work. However, to simplify the explanation, only the inverter control is treated in this section.

4.2 Multiplexing of autonomous decentralized control

First, the work of basic control described in the previous section is summarized again. The specific control procedure of basic control is as follows:

Step 1. When the voltage at any node is out of proper range, the DGs which can observe the voltage deviation adjust the reactive power in order to recover the voltage deviation (V-Ref method)

Step 2. The neighboring DGs control their reactive power by the work of Q-Coop method.

Step 3. The control action due to the Q-Coop method travels to the distant nodes like chain reaction.

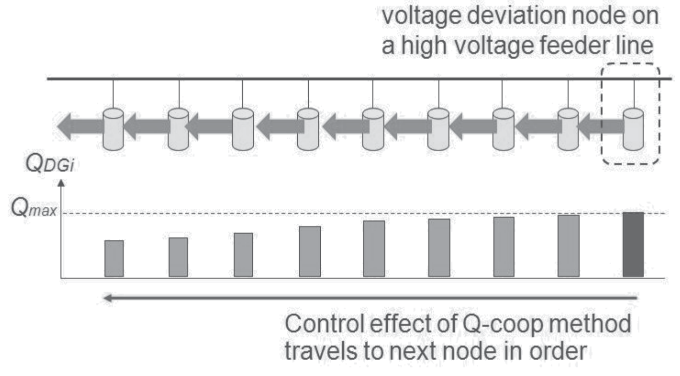


Fig. 17. Concept of Q-Coop method.

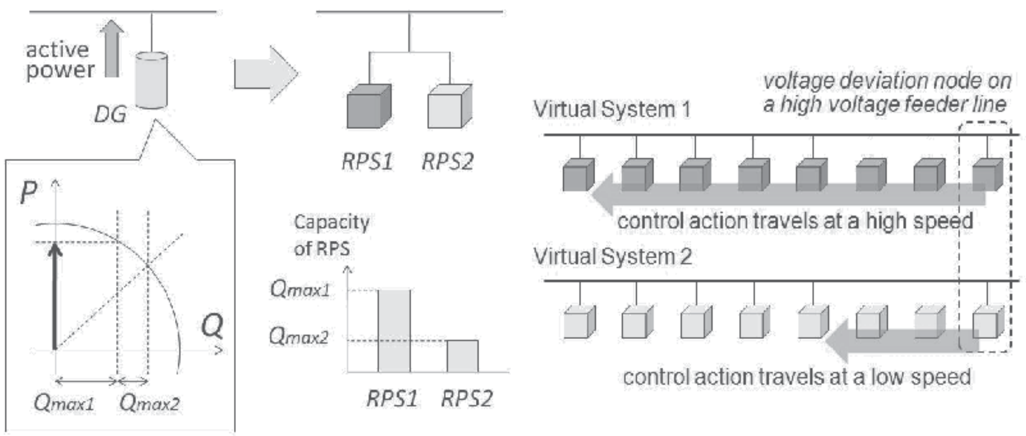
According to above procedure, nearer the DG locates, larger the reactive power is controlled as shown in fig.17. Therefore, the amount of reactive power output of each DG depends on the distance between the DG and the voltage deviation node. At this time, the control priority is not considered. Thus, in this section, the distribution network is virtually divided into two systems. One is "Virtual System 1" composed by the control of mode 1, and the other is "Virtual System 2"composed by the control of mode 2. Control speed of Virtual System 1 is set to be higher than that of Virtual System 2. Which enables to utilize the mode 1 control preferentially. Specific procedure is as follows.

Step 1. The agent of each DG has two Reactive Power Sources (RPS1 and RPS2) virtually. The capacity of RPS1 and RPS2 are defined as same as the capacity of mode 1 and mode 2, respectively. In the case that the capacity of either RPS1 or RPS2 becomes zero, the capacity is defined as ε. The capacity ε needs to be sufficient small value, and in this study, ε is one-twelfth of inverter capacity. As a result, all DGs have capacities of RPS1 and RPS2. (fig.18 (a))

Step 2. The neighboring agents exchange their reactive power output of RPS1 and RPS2. The required information for RPS1 control are "voltage value at self-node and neighboring nodes" and "reactive power output of self-node and RPS1 at neighboring nodes" while the information about RPS2 is not required. Oppositely, the controller of RPS2 does not need the information of RPS1. Hence, it is possible to divide the information exchange system into two systems. They are defined as "Virtual System 1" and "Virtual System 2". (fig.18(b))

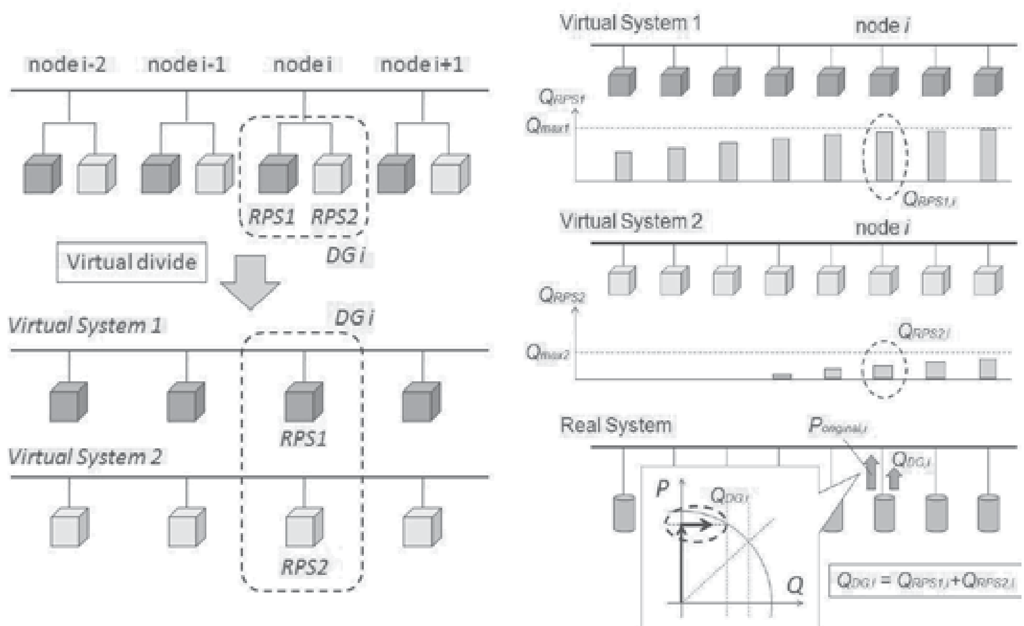
Step 3. In each Virtual System, the agent of each DG controls its RPS1 and RPS2 individually using control logic described in 4.3 which is based on the basic method. The control speed of each Virtual System is adjusted so that the Q-Coop method works faster in Virtual System 1 than that in Virtual System 2. (fig.18(c))

Step 4. Virtual reactive power of each RPS1 and RPS2 are summed up by the agent of each DG and it becomes the reference value of each DG. If the reference value is smaller than its capacity of mode 1, the active power decrease is not necessary. (fig.18(d))



(a) STEP1 : Reactive Power Source.

(c) STEP3 : Autonomous Decentralized Control in each Virtual System.



(b) STEP2 : Virtual System.

(d) STEP4 : Reactive power of each real DG.

Fig. 18. Proposed method by virtually divided system.

The concept of the proposed method is shown in fig.18. Both the high-speed Virtual System 1 and the low-speed Virtual System 2 work in parallel, and chain reaction of Q-Coop method works faster in Virtual System 1. As a result, mode 1 controls of all DGs are utilized preferentially because the reactive power control of RPS1 which is included in Virtual System 1 works faster.

4.3 Autonomous decentralized voltage profile control method in each Virtual System

The reactive power control of each Virtual System is also composed by V-Ref method, Q-Save method and Q-Coop method. The control logic of V-Ref method is same as that in section 3, and it is simply modified to Eq.(18) to realize the multiplexing. Q-Save method and Q-Coop method also need some modifications. The equalization of power factor is main aim of Q-Coop method in the proposed method in section 3.3. However, in the case the inverter has a free capacity, power factor is not proper index for expressing the control efficiency because the reactive power control based on mode 1 does not decrease the control efficiency. Hence, in this section, the ratios of the amount of reactive power to its maximum value are utilized in place of power factor. The ratio is also utilized in Q-Save method. The modified equations are shown in Eq.(19)-(21).

(a) Voltage Reference method (V-Ref method)

$$if \quad V_k(t) < V_{\min} \quad \text{or} \quad V_{\max} < V_k(t)$$

$$\begin{cases} T_{\alpha 1}\dot{Q}_{RPS1,i}(t) = K_{\alpha 1}(V_{ref,i} - V_k(t)) & (RPS\,1) \\ T_{\alpha 2}\dot{Q}_{RPS2,i}(t) = K_{\alpha 2}(V_{ref,i} - V_k(t)) & (RPS\,2) \end{cases} \tag{18}$$

where,

$Q_{RPS1,i}$, $Q_{RPS2,i}$: reactive power of RPS1 and RPS2 of DG i

$T_{\alpha 1}$, $T_{\alpha 2}$: time constant of V-Ref method in each Virtual System [sec]

$K_{\alpha 1}$, $K_{\alpha 2}$: control gain of V-Ref method in each Virtual System

$V_{ref,i}$: reference voltage at node i (usually defined as 1.00 [p.u.])

$V_k(t)$: voltage at node k where the voltage deviation value becomes maximum within IEA

(b) Q Saving method (Q-Save method)

$$if \quad V_{\min} + \Delta < V_k(t) < V_{\max} - \Delta$$

$$\begin{cases} T_{\beta 1}\dfrac{\dot{Q}_{RPS1,i}(t)}{Q_{\max 1,i}} = K_{\beta 1}(Q_{ref,i} - \dfrac{Q_{RPS1,i}(t)}{Q_{\max 1,i}}) & (RPS\,1) \\ T_{\beta 2}\dfrac{\dot{Q}_{RPS2,i}(t)}{Q_{\max 2,i}} = K_{\beta 2}(Q_{ref,i} - \dfrac{Q_{RPS2,i}(t)}{Q_{\max 2,i}}) & (RPS\,2) \end{cases} \tag{19}$$

where,

$T_{\beta 1}$, $T_{\beta 2}$: time constant of Q-Save method in each Virtual System [sec]

$K_{\beta 1}$, $K_{\beta 2}$: control gain of Q-Save method in each Virtual System

$Q_{ref,i}$: reference reactive power of Q-Save method (it is set to 0.00 [p.u.] in this section)

(c) Q Cooperation method (Q-Coop method)

$$if \quad V_{\min} < V_k(t) < V_{\max}$$

$$\begin{cases} T_{\gamma 1}\dfrac{\dot{Q}_{RPS1,i}(t)}{Q_{\max 1,i}} = K_{\gamma 1}(Q^{*}_{RPS1,ref,i}(t) - \dfrac{Q_{RPS1,i}(t)}{Q_{\max 1,i}}) & (RPS\,1) \\ T_{\gamma 2}\dfrac{\dot{Q}_{RPS2,i}(t)}{Q_{\max 2,i}} = K_{\gamma 2}(Q^{*}_{RPS2,ref,i}(t) - \dfrac{Q_{RPS2,i}(t)}{Q_{\max 2,i}}) & (RPS\,2) \end{cases} \tag{20}$$

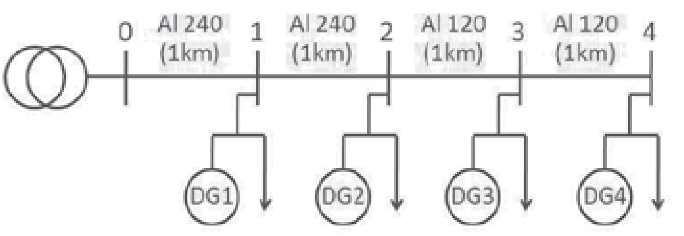
$$\begin{cases} Q^*_{RPS1,ref,i} = \dfrac{1}{M_i} \displaystyle\sum_{j \in G} \dfrac{Q_{RPS1,j}}{Q_{\max 1,j}} \\ Q^*_{RPS2,ref,i} = \dfrac{1}{M_i} \displaystyle\sum_{j \in G} \dfrac{Q_{RPS2,j}}{Q_{\max 2,j}} \end{cases} \tag{21}$$

where,
$T_{\gamma 1}$, $T_{\gamma 2}$: time constant of Q-Coop method in each Virtual System [sec]
$K_{\gamma 1}$, $K_{\gamma 2}$: control gain of Q-Coop method in each Virtual System
$Q^*_{RPS1,ref,i}(t)$, $Q^*_{RPS2,ref,i}(t)$: reference reactive power of Q-Coop method in each Virtual System
M_i : total node number included in IEA
G_i : node group neighboring node i
Above three control methods do not always work, and they are switched by the voltage information within IEA. When the voltage deviation node is included in IEA, reactive power is controlled according to Eq.(18). When the voltage deviation node is not included, the reactive power control is determined as a sum of the Eq.(19) and (20). Control parameters of each Virtual System are determined individually according to following guidelines.

- Time constant of RPS1 is set to a small value in order that control becomes faster in Virtual System 1.
- It is desirable that Q-Save method works faster in Virtual System 2. So the control gain of Q-Save method of RPS2 is set to a larger value.

4.4 Simulation results using 5-node model system

(a) Model system



6.6kV 20MVA base

Fig. 19. 5-node model system.

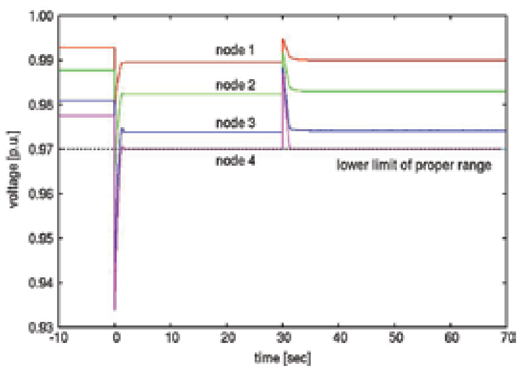
The proposed method is tested using 5-node model system shown in fig.19. Both load and DG are connected to all nodes and load capacity is relatively larger. Then, the voltage drop becomes main issue in this case. The detailed settings of the model system are listed in table 5. Although the active power of all DGs are set to 0.050 [p.u.], inverter capacity of DG1,2 and DG3,4 are different as shown in the table and the DG1 and DG2 have free capacities. Additionally, DG1,2 have only the free capacity as shown in the section (B) in fig.3. Because the DG3,4 have no free capacity, they have only the control capacity of mode 2 oppositely. The amount of each load is set to 50% at initial condition. After that, the loads show the stepwise change from 50% to 90% at time=0[sec], and from 90% to 75% at time=30 [sec], respectively.

Load capacity of each node	0.070[p.u.] (p.f.lag 0.80)
Inverter capacity (active power of DG) : node 1,2 node 3,4	 0.080[p.u.] (0.050 [p.u.]) 0.050[p.u.] (0.050 [p.u.])
Power factor control range of controllable DG	lag 0.8～lead 0.8
Objective range of voltage	0.97 ～ 1.03 [p.u]
Line impedance Al240 (1km/1section) Al120 (1km/1section)	 0.058+j0.138 [p.u.] 0.116+j0.276 [p.u.]

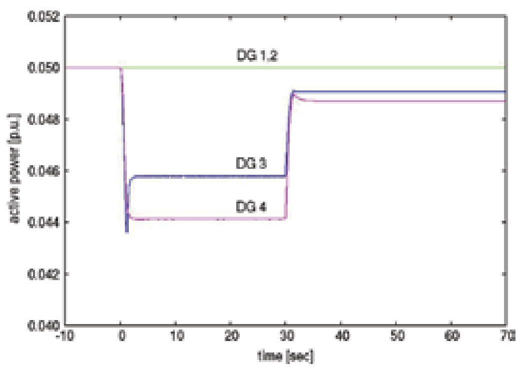
Table 5. System constants of 5-node model system.

(b) Application of basic method (case 1)

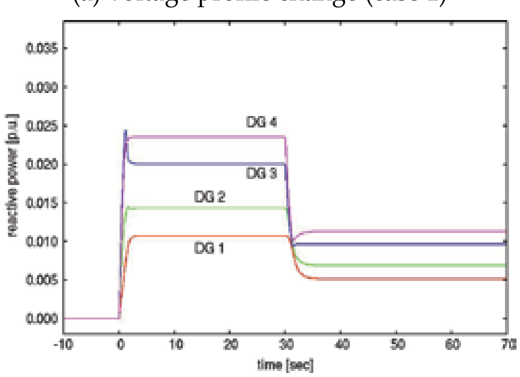
Control parameters are listed in table 6. First, fig.20 shows the simulation results by the basic method proposed in section 3. Figure 20(a)-(d) show the voltage profile change, active and reactive power change, and the ratio of reactive power to its control capacity. Voltage decreases largely by the load change at time=0 [sec]. DG3 and DG4 increase their reactive



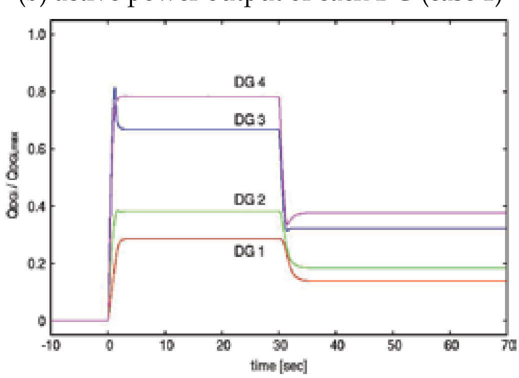
(a) voltage profile change (case 1)



(b) active power output of each DG (case 1)



(c) reactive power output of each DG (case 1)



(d) ratio of reactive power of DG to its capacity (case 1)

Fig. 20. Simulation results in case 1.

power in order to recover the voltage profile at node 3 and node 4. To help the reactive power control at downstream side, the reactive power of DG1 and DG2 are also increased by the work of Q-Coop method. In the case the basic method is applied, the nearer the voltage deviation node, the larger the reactive power is controlled. Therefore, the amount of reactive power control of DG1 and DG2 become smaller relatively. Hence, the reactive power of DG3 and DG4 become larger than that of DG1 and DG2. As a result, the active power of DG3 and DG4 decrease as shown in fig.20(b). Especially from fig.20(d), we can see that DG1 and DG2 are utilized only 30-40[%] of their free capacity while reactive power of DG3 and DG4 are utilized about 70-80[%] of their capacity. As seen above, in the case the control priority is not considered, it is possible the heavy control burden is imposed to the specific DG whose free capacity is small. At this time, it should be possible that large active power reduction occurs at the DG.

T_α	1.00 [sec]	T_β	1.00 [sec]	T_γ	1.00 [sec]
K_α	0.25	K_β	0.50	K_γ	3.00

Table 6. Control parameters. (case 1).

(c) Application of advanced control method (case 2)

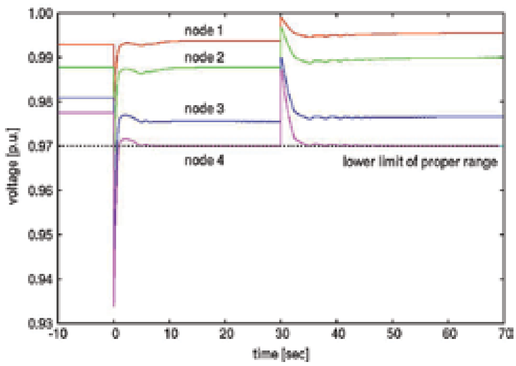
The proposed method is applied to the same 5-node model system. Control parameters are determined as table 7 according to the following guidelines in order to make a fair comparison with the results by previous subsection

- Let the control performance of V-Ref method at mode 2 become same as that of the case 1. Specifically, the control gain and time constant are adjusted to satisfy $K_\alpha / T_\alpha = K_{\alpha 2} / T_{\alpha 2}$.
- Let the control performance of Q-Coop method at mode 1 become same as that of the case 1. Specifically, the control gain and time constant are adjusted to satisfy $K_\gamma / T_\gamma = K_{\gamma 1} / T_{\gamma 1}$.

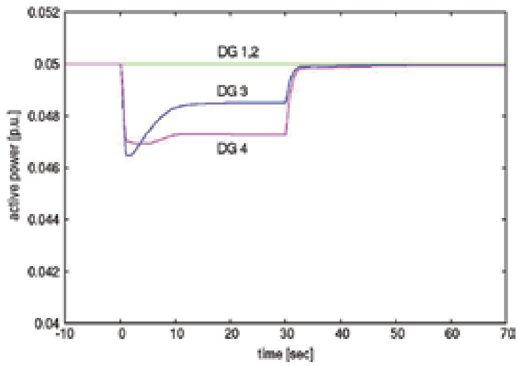
Mode 1	$T_{\alpha 1}$	1.00 [sec]	$T_{\beta 1}$	1.00 [sec]	$T_{\gamma 1}$	1.00 [sec]
	$K_{\alpha 1}$	1.00	$K_{\beta 1}$	0.05	$K_{\gamma 1}$	3.00
Mode 2	$T_{\alpha 2}$	4.00 [sec]	$T_{\beta 2}$	4.00 [sec]	$T_{\gamma 2}$	4.00 [sec]
	$K_{\alpha 2}$	1.00	$K_{\beta 2}$	2.00	$K_{\gamma 2}$	1.00

Table 7. Control parameters. (case 2).

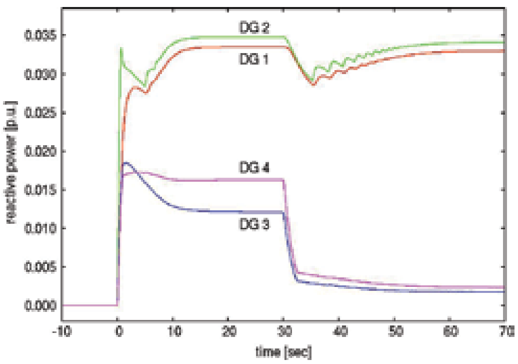
Figure 21 shows the simulation results by the proposed method. Figure 21(a)-(d) show the voltage profile change, active and reactive power change, and the ratio of RPS1 output to its control capacity. Voltage at node 2, 3, 4 become lower than the lower limit of proper range at time=0 [sec] due to the load change at the same time (fig.21(a)). Then, the voltage profile is recovered before time=1[sec] by the work of reactive power control of DGs immediately. Especially, the reactive power of DG1 and DG2 are utilized greatly due to the work of Virtual System 1. Specifically, after the voltage deviation occurs, the proposed autonomous decentralized control starts to work at both Virtual Systems and the reactive power control of Virtual System 1 works faster than that of Virtual System 2. We can see from fig.21(d) that the Q-Coop method works rapidly in Virtual System 1.



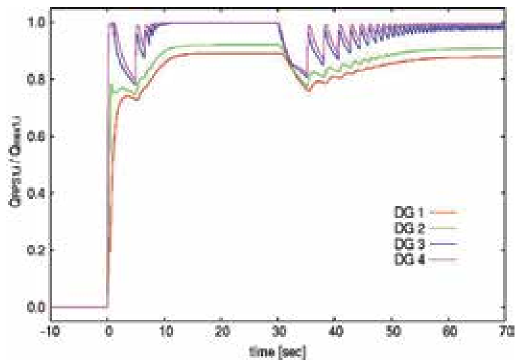
(a) voltage profile change (case 2)



(b) active power output of each DG (case 2)



(c) reactive power output of each DG (case 2)



(d) ratio of reactive power of RPS1 to its capacity (case 2)

Fig. 21. Simulation results in case 2.

It should be noted that the capacities of RPS1 at DG3 and DG4 are defined as ε virtually although the DG3 and DG4 have no free capacity. Utilizing the capacity of mode 1, DG3 and DG4 scarcely increases their ratios of reactive power to the maximum value when the voltage drop occurs. As seen above, even in the case a certain DG has no capacity of either RPS1 or RPS2, it can control its "ratio" using the small control capacity. In addition, because the reactive power by RPS1 of all DGs are not over the capacity of mode 1, only the mode 1 works actually. Therefore, the active power of DG1 and DG2 do not decrease as shown in fig.21(b). Although the reactive power of DG3 and DG4 decrease, the amount of active power reduction becomes smaller compared with case 1.

At the time=30 [sec], the required amount of reactive power for voltage profile maintenance decreases since the load becomes light again. By the work of Q-Save method, the reactive power of RPS2 decreases faster at this time because the control gain of Q-Save method is set to large value in Virtual System 2. Therefore, the reactive power of DG3 and DG4 decrease faster than that of DG1 and DG2. Which results in the rapid recover of active power reduction. The reactive power of DG1, 2 increase again after time=35 [sec]. This is because the reactive power of DG1 and DG2 are required again for voltage profile maintenance according to the reactive power decrease of DG3 and DG4. Then, the Q-Coop method works in both Virtual Systems and the reactive power of multiple DGs increase. Because the Q-Coop method in Virtual System 1 works faster, the reactive power of DG1 and DG2 increase largely. At this time, the required amount of reactive power is supplied by the Virtual System 1. In Virtual System 2, the reactive power do not increase and Q-Save method continues to work. As a result, at the steady solution, control mode 1 is effectively utilized and minimum amount of control mode 2 works properly.

4.5 Simulation results using 24-node model system

(a) Model system

The proposed method is tested using 24-node model system described in section 3.4. Although the overview of the system model is the same, the inverter capacity and active power of DGs are modified as shown in table 8, where the DGs at odd-numbered nodes have free capacity. The active power change of DG17 is modeled as fig.22. Table 9 shows the

Inverter capacity (active power of DG) : node 1,3,5, … , 23 node 2,4,6, … , 22	 0.030[p.u.] (0.020 [p.u.]) 0.040[p.u.] (0.040 [p.u.])

Table 8. Modified part of 24-node model system.

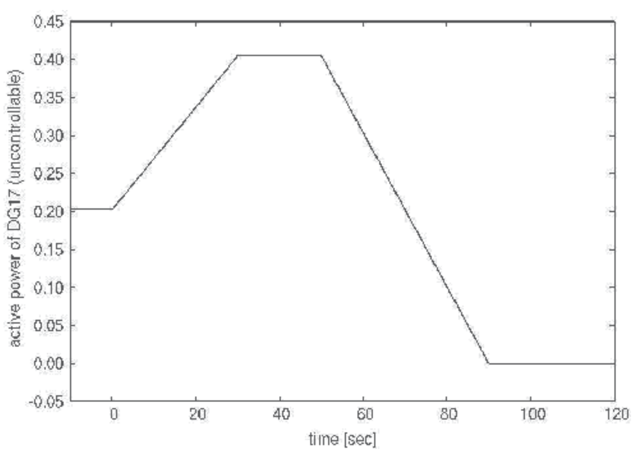
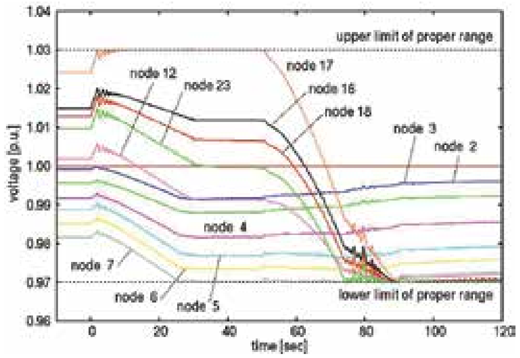


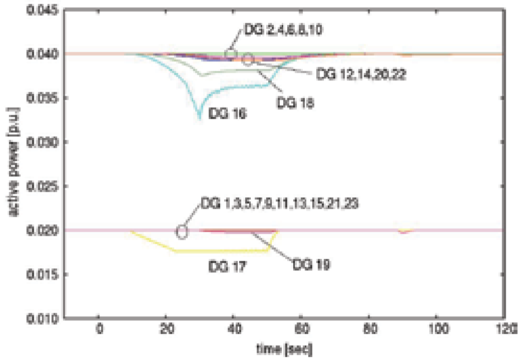
Fig. 22. Active power change of DG17.

Mode 1	$T_{\alpha 1}$	1.00 [sec]	$T_{\beta 1}$	1.00 [sec]	$T_{\gamma 1}$	1.00 [sec]
	$K_{\alpha 1}$	1.00	$K_{\beta 1}$	0.03	$K_{\gamma 1}$	1.00
Mode 2	$T_{\alpha 2}$	10.0 [sec]	$T_{\beta 2}$	10.0 [sec]	$T_{\gamma 2}$	10.0 [sec]
	$K_{\alpha 2}$	1.00	$K_{\beta 2}$	0.50	$K_{\gamma 2}$	1.00

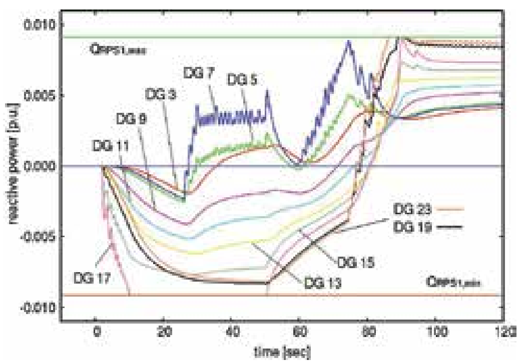
Table 9. Control parameters in 24-node model system.



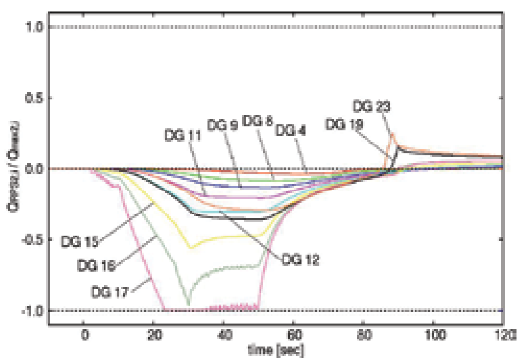
(a) voltage profile change (case 3)



(b) active power output of each DG (case 3)



(c) reactive power output of RPS1 (case 3)



(d) ratio of reactive power output of RPS2 to its capacity (case 3)

Fig. 23. Simulation results in case 3.

control parameters. Each parameter is determined by trial and error considering control stability.

(b) Simulation results (case 3)

Figure 23(c) shows the reactive power of RPS1 injected by DG3, 5 and 7 in order to control the voltage profile, active power of DGs, and the reactive power change of RPS1 and RPS2. As shown in fig.23(a), the voltage at node 17 reaches the upper limit at time=1[sec] by the active power change of DG17.

Before time=50 [sec], the voltage profile at downstream side of the feeder increases and it decreases around node 7 oppositely. Because the high and low voltage areas are included at the same time, the appropriate adjustment of voltage profile is required. Figure 23(c) shows RPS1 of DG3, 5 and 7 inject the reactive power in order to control the voltage profile within the proper range while RPS1 of other DGs absorb the reactive power to avoid the over voltage. As for the RPS2, the reactive power of most DGs are not so large except DG15,16,17 (fig.23(d)). DG16 and DG17 increase their reactive power largely due to V-Ref method because the voltage at node 17 reaches the upper limit of proper range. The reactive power of DG15 also becomes large due to Q-Coop method because DG15 neighbors DG16 whose reactive power is large. The active power of DG15 does not decrease because the sum of the reactive power of RPS1 and RPS2 is lower than the capacity of mode 1 while the active power of DG16,17,18 decrease. In addition, as for other than those above, we can see the active power decrease slightly at even-numbered DGs which have no free capacity due to the reactive power control of RPS2.

After time=70 [sec], it becomes possible for all DGs to supply the reactive power cooperatively since only the lower limit of voltage profile becomes problem. Hence, the reactive power of RPS1 of all DGs are controlled largely and the voltage profile is maintained properly without depending the reactive power control of RPS2.

As above, the proposed method works effectively even in the large scale system model. However, it is likely important to consider the control stability in the case that the proposed method is applied to the relatively large scale system. The decision technique for control parameters considering control stability should be developed in the future work.

5. Conclusion

To solve the energy and environmental issue, the introduction of DGs is one of important technologies. In this paper, we described a new voltage profile control method of a distribution network with a large number of DGs. Specifically, we have developed an autonomous decentralized voltage control method based on a multi-agent system using a reactive power control of inverter. The proposed method is composed by three control methods whose control purposes are different and only the local information exchange is required for flexible voltage control. In addition, as an advanced control method, we have described the voltage profile control method which can utilize a free capacity of inverter preferentially. Both basic and advanced methods are tested using 4-node, 5-node and 24-node model system and their effectiveness are shown. Future works are as follows.

- In our proposed method, the control object is limited to the continuous control such as reactive power control of inverter. However, there exist some discontinuous voltage control facilities such as LRT or SC in the distribution network. The cooperative work among continuous and discontinuous control should be developed.
- The control parameters are determined mainly by trial and error in this paper. However, in the future work, they should be adjusted automatically. Especially, it is expected to utilize the learning function of agents for automatic parameter adjustment considering control dynamics.

6. References

F.A. Viawan and D. Karlsson: “Combined Local and Remote Voltage and Reactive Power Control in the Presence of Induction Machine Distributed Generation”, IEEE Trans. Power Syst., Vol.22, No.4, pp.1002-1012 (2007)

F.A. Viawan and D. Karlsson: "Voltage and Reactive Power Control in Systems with Synchronous Machine-Based Distributed Generation", IEEE Trans. Power Delivery, Vol.23, No.2, pp.1079-1087 (2008)

T. Senjyu, Y. Miyazato, A. Yona, N. Urasaki and T. Funabashi: "Optimal Distribution Voltage Control and Coordination With Distributed Generation", IEEE Trans. Power Syst., Vol.23, No.2, pp.1236-1242 (2008)

Naoto Yorino, Takahiro Miki, Yuuki Yamato, Yoshifumi Zoka and Hiroshi Sasaki: "A Time Scale Separation Method for the Coordination of Voltage Controls for SVC and SVR", IEEJ Trans. PE, Vol. 124, No. 7, pp.913-919 (2004)

Daisuke Iioka, Kuniaki Sakakibara, Yasunobu Yokomizu, Toshiro Matsumura and Norihisa Izuhara: "Distribution Voltage Rise at Dense Photovoltaic Power Generation Area and its Suppression by SVC", IEEJ Trans. PE, Vol. 126, No. 2, pp.153-158 (2006)

P. N. Vovos, A. E. Kiprakis, A. R. Wallace and G. P. Harrison: "Centralized and Distributed Voltage Control: Impact on Distributed Generation Penetration", IEEE Trans. Power Syst., Vol.22, No.1, pp.476-483 (2007)

P. M. S. Carvalho, P. F. Correia and L. A. F. M. Ferreira: "Distributed Reactive Power Generation Control for Voltage Rise Mitigation in Distribution Networks", IEEE Trans. Power Syst., Vol.23, No.2, pp.766-772 (2008)

Shunsuke Tanaka and Hirokazu Suzuki: "A Study on Voltage Compensation Method using Autonomous Decentralized Control of Distributed Generators", IEEJ Trans. PE, Vol. 129, No. 7, pp.869-879 (2009)

Mesut E. Baran and Ismail M. El-Markabi: "A Multiagent-Based Dispatching Scheme for Distributed Generators for Voltage Support on Distribution Feeders", IEEE Trans. Power Syst., Vol.22, No.1, pp.52-59 (2007)

Takao Tsuji, Ryoichi Hara, Tsutomu Oyama, and Keiichiro Yasuda: "Autonomous Decentralized Voltage Profile Control of Super Distributed Energy System using Multi-agent Technology", IEEJ Trans. EIS, Vol.126, No.2, pp.167-176 (2006)

Takao Tsuji, Tadahiro Goda, Kazushige Ikeda and Seiji Tange: "Autonomous Decentralized Voltage Profile Control of Distribution Network considering Time-Delay", Proceedings of the International Conference on Intelligent System Application to Power System, WA3-2 (2007)

Takao Tsuji, Tadahiro Goda, Kazushige Ikeda and Seiji Tange: "A Study of Autonomous Decentralized Voltage Profile Control of Distribution Network considering Economic Efficiency", IEEJ Trans. PE, Vol. 128, No. 1, pp.174-185 (2008)

Takao Tsuji, Kohsuke Tomura, Tsutomu Oyama, Takuhei Hashiguchi Tadahiro Goda, Seiji Tange and Toshio Nomura: "A Study of Centralized Voltage Profile Control of Distribution Network considering Dynamics of Distributed Generator", IEEJ Trans. PE, Vol.129, No.4, pp.507-516 (2009)

Takao Tsuji, Noriyuki Satoh, Takuhei Hashiguchi, Tadahiro Goda, Seiji Tange and Toshio Nomura: "A Study of Economical Incentives for Voltage Profile Control Method in Future Distribution Network", IEEJ Trans. PE, Vol.129, No.6 (2009)

Takao Tsuji, Takuhei Hashiguchi, Tadahiro Goda, Takao Shinji and Shinsuke Tsujita: "A Study of Autonomous Decentralized Voltage Profile Control Method considering Control Priority in Future Distribution Network", IEEJ Trans. PE, Vol. 129, No. 12, pp.1533-1544 (2009)

12

Multi Agent Systems combined with Semantic Technologies for Automated Negotiation in Virtual Enterprises

Koppensteiner Gottfried[1], Merdan Munir[1], Lepuschitz Wilfried[1], Moser Thomas[2] and Reinprecht Constantin[3]
[1]*Vienna University of Technology, Automation and Control Institute*
[2]*Vienna University of Technology, Inst. of Software Development & Interactive Systems*
[3]*Vienna Institute of Technology, Department for Information Technology*
Austria

1. Introduction

In today's competitive global markets, no company can be successful without effectively managing its supply chain. Procurement is a crucial process and it accounts for more than half of the enterprises' sales volume. However, various problems have been identified in the past studies in this process: paper-based purchasing cycles and lack of automation, lack of intelligent advice tools for finding suitable suppliers, as well as difficulties to evaluate the performance of the suppliers (Lee et al. 2007). Appropriate supply chain organization will help companies to resolve conflicts, to reduce logistic lead times and total costs, to increase profits and to become more competitive on the market (Jain et al. 2009). Recent developments of electronic means for communication and collaboration between business partners led to the emergence of electronic negotiation as an alternative to manual contracting. The interest in e-negotiation is motivated by its potential to provide business partners with more efficient processes, enabling them to gain better contracts in less time (Rinderle and Benyoucef 2005). E-negotiation should allow automated or semi-automated processing of the information and transactions which take place between two companies and thus reduce costs and increase speed. However, by increasing the degree and the sophistication of the automation on both sides, commerce becomes much more dynamic, personalized, and context sensitive (He et al. 2003). A major challenge in distributed software development is the integration of knowledge scattered over processes, tools and people to facilitate an effective and efficient coordination of activities along the development lifecycle. The EU research FP7 focuses on a key strategic research challenge[1] in the area of semantic systems/technology development to deal with the "growing load and diversity of information and content" which particularly applies to software engineering projects where semantic support for data analysis and reasoning would improve data and knowledge exchange among distributed project teams and promises more effective collaboration.

---

[1] http://cordis.europa.eu/fp7/ict/content-knowledge/fp7_en.html

Due to the high level of complexity and heterogeneity in these environments, traditional electronic commerce technologies have difficulties to support automated negotiation. The agent approach offers a convenient way to deal with complexity, autonomy and heterogeneity by providing the most natural means of representing the distinct individuals and organizations, suitable abstractions, the ability to wrap legacy systems and flexibility for organizational structure changes (Jennings 2001). The benefit of agent technology in contrast to the commonly used mathematical programming is that it is more versatile and can easily capture the qualitative as well as the quantitative nature of a complex technical system (Alvarado et al. 2007). The agents, each having individual goals, abilities and authorities, make decisions on behalf of their principals negotiating with each other, buying as well as selling goods or services and trying to achieve the best possible results. In this context, autonomous software agents are cited as a potentially fruitful way of approaching e-commerce automation (Kowalczyk et al. 2002). Nevertheless, traditional negotiation approaches pose a number of constraints on the type of interactions that can take place among agents, allowing only agents identified in advance to participate in a negotiation and predeterminating the protocols to be used (Tamma et al. 2005). Besides, most of the used MAS are designed with a single negotiation protocol explicitly hard-coded in the agents resulting in an inflexible environment able to accept only agents designed for it (Preist et al. 2002). Moreover, typical semantic gaps exist in the exploitation of current agent systems considering the fact that mostly just syntactic matching is used for detecting equalities between exchanged data during the communication between agents.

To allow communication among heterogeneous systems, there must be an agreement on both the syntax and the semantics of the exchanged messages or of any other type of interaction. The syntax defines the structure of a language, i.e., a grammar typically in a form of rules that govern the structure of sentences. Semantics refers to the aspects of meaning as expressed in the language, i.e., the sense of language elements and their combination, including the relation of these elements to the real world. Ontologies are a way of representing concepts and relationships embedded in the negotiation environment such that they are semantically meaningful to a software agent, ensuring that agents refer to the same object (Beam and Segev 1997). Ontologies can be used to describe the semantics of the information sources and make the contents explicit, thereby enabling the integration of existing information repositories. The negotiation ontology, that specifies basic terms an agent has to share in order to enable understanding of the used concepts and conditions under which the interaction between agents takes place, is defined by Tamma et al. (Tamma et al. 2002). Besides, the negotiation ontology is used to describe interaction protocols related to the dynamic Cross-Organizational Workflow context in order to permit the different agents to dynamically select at run-time a given negotiation protocol during the opening of a negotiation step (Bouzguenda et al. 2008). However, the overall efficiency of the negotiation process is, besides the general "understanding", strongly influenced by different attributes of the deal (quality of product/service, availability, risk, delivery time, etc.) whose consideration/non-consideration can significantly influence the agent's results. Besides, up to now, most of the research done on automated negotiation considers models where the price is the unique strategic dimension (Renna and Argoneto 2010).

A decision support system (DSS) is required which is able to check multiple attributes of a deal and by merging them suggests suitable solutions (David et al. 2006). In this work we combine a multi-agent framework with ontology-driven solutions to support and automate the procurement process. The resulting Semantic-enabled Decision Support System (SemDSS), which is composed of agents with different goals and tasks organized in a suitable architecture, is able to simultaneously consider multiple attributes of a negotiation and assists users in the decision-making process. In our approach, the agent is an autonomous semantic entity having specific tasks and knowledge about its domain of application in the procurement process, about strategies, which can be used to achieve a specific goal during the procurement process, as well as about (other) relevant agents involved in the DSS. Considering that the ontological framework is the key for agents to "understand" the process and to link the context of data, process or services within or between entities, we introduce a semantic foundation to bridge relevant semantic gaps in order to radically improve the automated data processing in semantically heterogeneous environments, i.e., information from different users or open source communities with their own API standards. This semantic foundation is based on a semantic model, describing the information to be integrated as well as artifacts in a way, which is both human-readable as well as useful for machine-based automated parsing and logic reasoning. The model is supported by automated semantic annotations of artifacts, based on their natural language descriptions.

Motivation study

The Automation and Control Institute of the Vienna University of Technology has conducted the study "Roadmap for the movement of SMEs towards semantic knowledge-based systems" within the framework of a project financed by the Vienna Economics Chamber. This study analyzed current state of the art and issues related to the application and capabilities of semantic technology in more than 100 Austrian companies, which are employed in various sectors (information and consulting, transport, tourism, industry, bank, etc.). The results of this study have shown that almost 40% of companies have difficulties in the data exchange with other business partners, since data is not represented in a uniform format. 51% confirm that the amount of manual data processing is too high. For almost 62%, the optimization of business processes and improving cooperation with business partners would be the main reason to introduce a new IT system (Merdan et al. IN PRESS).

2. State of the art

2.1 Agent-based decision support systems

Decision support systems have been used by negotiators to support individuals in negotiations. They are user-oriented, because they help users to understand and formalize their objectives and preferences. Moreover, they are problem-oriented because they help users to understand the problem structure, search for solutions, and conduct sensitivity analyses (Kersten and Lai 2007). Early DSS work advocates the development of such systems to allow users to quickly develop them application-specific (Bui and Lee 1999). Multi-agent technology is a superior solution that can enhance a system's connectivity, extensibility, reusability, and manageability. Intelligent agents are goal-oriented, collaborative, flexible, self-starting, intelligent and interactive (Turban et al. 2005). These characteristics of agents ensure the application of multi agents in DSS to be successful (Zhang 2009). Some reports on the application of agent technologies for DSS exist in literature. Ai et al. presented a distributed marketing decision support system (Ai et al. 2004). Houari and Far presented a novel architecture based on multi-agents technology to support information and knowledge extraction over distributed data sources in order to use them in the decision making process (Houari and Far 2004). Zhang and Xi present a decision support system framework for the partner choice and measurement (Zhang and Xi 2005). Neubert et al. proposed a software agent, capable of conducting an automated negotiation in order to assist the human decision-maker in an environment consisting of small independent units (Neubert et al. 2004). Wang et al. presented the architecture of Web-oriented Warfare Command DSS (Wang et al. 2005). Zhang and Yu reported on the bidding decision-making for power suppliers (Zhang and Yu 2006). Puigjaner et al. conducted a research in the context of supply chain in the chemical process industry (Puigjaner et al. 2008). However, in most of the cited papers, the considered solutions are pretty much custom oriented and the negotiation process takes just one parameter (the price) or only one item into account. Moreover, the presented approaches lack solutions that support interoperability when applied in a heterogeneous environment. Having identified the need to extend current DSS approaches to effectively and efficiently facilitate the integration of data from heterogeneous systems, the research on semantics provides a range of very useful concepts to build a strong semantic foundation for DSS.

2.2 Ontologies for integrating domain, project and tool data

Ontologies can represent domain vocabulary, i.e., the conceptualizations that the terms in the vocabulary are intended to capture (Chandrasekaran et al. 1999, Davies 2006). Ontologies have been used in IT for a range of objectives: for clarifying the knowledge structure, for reducing conceptual and terminological ambiguity and for enabling knowledge sharing extensions to typical IT projects (Calero et al. 2007, Hepp et al. 2007, Pastor et al. 2008). Typical objectives for ontology usage include a) the declarative specification of a system; b) the support for manual or semi-automated consistency checks; c) the improvement of the documentation and a reduction in the efforts needed for maintenance; d) the reuse of aspects between different domains or tasks; and e) the acquisition and storage of knowledge, e.g., with reasoning (Happel and Seedorf 2008, Baclawski et al. 2001).

2.3 Semantic integration of data from heterogeneous sources

Providing integrated access to multiple heterogeneous sources in information systems raises challenging cooperation and interoperability issues (Bergamaschi et al. 1999): firstly, how to determine if the sources contain semantically related information, i.e., information related to the same or similar real-world concept(s), and secondly, how to handle semantic heterogeneity to support integration and uniform query interfaces. The successful use of the ontologies concept within modern data integration and business intelligence techniques and technologies has been reported particularly within heterogeneous domains where the use of common meta-data, services and processes seems most valuable (Fonseca et al. 2000). An ontology enables information from one resource to be mapped accurately at an extremely granular level to information from another source. The ontology provides the common vocabulary for the data integration – showing the preferred names for a concept, and the synonyms and properties associated with this concept. This enables the forward-looking integration by collecting data using names that are already well understood rather than ones that might not be shared widely throughout the organization. This makes the assimilation of new data easier and quicker, and facilitates communication between groups (Eilbeck et al.

2005). Furthermore, the ontology can grow over time as new data become available, new links are continually being made and new knowledge assimilated in the ontology (Kiryakov et al. 2004). Moser et al. (Moser 2009; Moser et al. 2010) introduced the Engineering Knowledge Base (EKB) framework as a semantic web technology approach for addressing challenges coming from data heterogeneity that can be applied for a range domains, e.g., in the production automation domain and also software engineering. Further, Biffl et al. (Biffl et al. 2010) used the approach for solving similar problems in the context of open source software projects, in particular, frequent-release software projects.

2.4 Tool support for semantic annotation

Semantic annotation is a specific metadata generation and usage schema targeted to enable new information access methods and extend existing ones. Annotation schemas are often based on the understanding that the named entities mentioned in documents or artifacts constitute important parts of their semantics. Furthermore, using different sorts of redundancy, external or background knowledge, those entities can be coupled with formal descriptions and thus provide more semantics and connectivity to other resources (Michelson and Knoblock 2007). Ontological structures may give additional value to semantic annotations. They allow additional possibilities concerning the resulting semantic annotations, such as reasoning. Furthermore, an ontology directs the attention of the annotator to a predefined choice of semantic structures and therefore gives some guidance about what and how items residing in the documents may be annotated (Staab et al. 2001).

There are a number of tricky issues with providing semantic annotation: First of all, the semantic annotation task does not adhere to a strict template structure, such as Dublin Core to name one of the more sophisticated ones in use. Rather it needs to follow the structure given by schema definitions that may vary with, e.g., domain and purpose. Semantic annotations need to be congruent with ontology definitions in order to allow the advantages we have indicated above. Secondly, semantically interlinked metadata is labor-intensive to produce and, hence, expensive. Therefore duplicate annotation must be avoided. Because semantic annotation is a continuous process in a distributed setting, there are several sources for duplication. Thirdly, purely manual annotation is very expensive. Therefore, only very valuable information will be annotated and it is necessary to help the human annotator with his task. What is needed is support for automatic — or at least semi-automatic — semantic annotation of information. Finally, there is a lack of experience in creating semantically interlinked metadata. It is not clear how human annotators perform in total and, hence, it is unclear what can be assumed as a baseline for the machine agent (Cimiano et al. 2004).

In the context of distributed projects automated semantic annotation tries to identify the semantic meaning of natural language terms or descriptions of artifacts by parsing the terms and analyzing the interrelationships between noun groups and verbal groups, based on a set of semantic categories defined for a certain project. By now, semantic annotation is primarily a manual task that needs additional time and efforts, resulting in reluctance of the project stakeholders to fulfill the task if the added value of their action is not clearly visible. The semantic annotations can later on be used for the derivation of new or à-priori unknown facts using semantic reasoning mechanisms provided by ontology reasoners (e.g., RACER[2], Pellet[3] or FaCT[4]). Using the capabilities provided by these reasoners, it is possible

[2] http://www.racer-systems.com/

to detect new or unknown relationships between software artifacts, errors or faults in the consistency of the artifacts stored/created by the different tools, as well as to effectively perform change impact analyses by exploiting the information on relations between the modeled individuals/artifacts (Michelson and Knoblock 2007).
Because meaningful sentences are composed of meaningful words, any system that hopes to process natural languages as people do must have information about words and their meanings. This information is traditionally provided through dictionaries, and nowadays machine-readable dictionaries are widely available. But dictionary entries evolved for the convenience of human readers, not for machines. WordNet[5] provides a more effective combination of traditional lexicographic information and modern computing. WordNet is an online lexical database designed for the usage under program control. English nouns, verbs, adjectives, and adverbs are organized in sets of synonyms, each representing a lexicalized concept. Semantic relations link these synonym sets (Miller 1995).

3. System architecture

3.1 SemDSS architecture

The agents' architecture for the SemDSS System is derived from a general automated negotiation framework, where two or more entities try to reach agreement on one or more matters of common interest. The system is composed of three main parts (see Figure 1): (i) the Auction Framework, (ii) the Agent based Negotiation System with the Decision Support Algorithm (DSA) and (iii) the Decision Support System for Supplier Ranking.

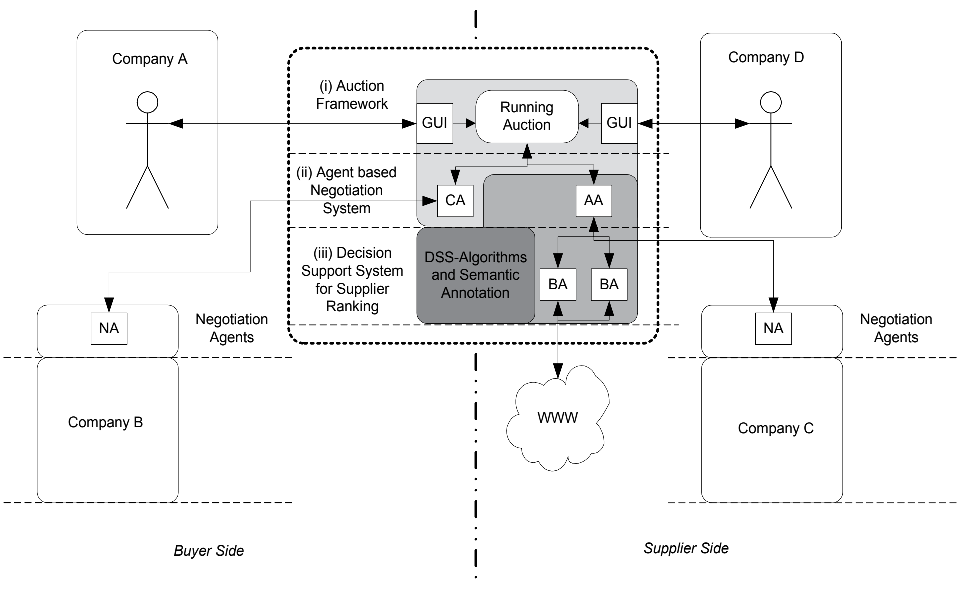


Fig. 1. SemDSS Overview

[3] http://pellet.owldl.com/
[4] http://jade.tilab.com/
[5] http://wordnet.princeton.edu/

The Auction Framework defines the environment required to enable an automated negotiation. It specifies the responsibilities and behaviors of involved entities and handles all necessary activities during the negotiation process allowing one entity (a purchaser) to start the negotiation as well as other ones (suppliers) to participate in it and place bids. The Agent based Negotiation System and related architecture is already developed and successfully tested at ACIN and consists of a Contact Agent (CA), responsible for the first contact of a user (buyer or seller) with the Auction Framework and an Auctioneer Agent (AA), which is upgraded in the presented system. Additionally, at each company there could be a Negotiation Agent (NA) which represents the user/company. The only tasks of this NA are to compete in auctions if the company wants to buy something or to inform the Auction Framework if the company wants to start an auction in order to sell a product or a service. The auctions themselves are running on the central Auction Framework, which is placed on its own server. The company's NAs are running on the company's server and can interact with other agents of that company. The CA on the negotiation system is just responsible for the first requests from the company's NA and for creation of the AA for each auction. The negotiations are running directly between the NAs on the company's sides and an AA on the Auction Framework. The AA, which collects information about every auction and stores them in its knowledge base using semantic annotation methods to enhance the information quality, is also used as the interface between the Agent based Negotiation System and the Decision Support System for Supplier Ranking. The AA's further responsibilities are associated with the system's supplier ranking activities and will be more elucidated. This system is using standardized protocols and messages based on ontologies and is able to run different kinds of auction types. It integrates a special Decision Support Algorithm, which is able to select the most profitable auction type for a specific good on the buyer side as well as to propose the most suitable strategy for placing bids on the seller side (Koppensteier et al. 2009a). It has been shown that companies which are using the Auction Framework and the Agent based Negotiation System, as described in (Koppensteiner et al. 2009), could enhance their agility and improve interoperability when forming virtual enterprises.

3.2 Decision Support System for Supplier Ranking

The Decision Support System for Supplier Ranking, as a third part of the SemDSS, is derived from the normal auction process steps (see Figure 2). Normally, the first step in an auction process is to (i) start the auction. A buyer introduces information into the auction systems concerning the wanted goods, about the expected performance (such as delivery, lead-time, quality, price, or some combination of these variables) as well as about the start and end of an auction. The AA, which is automatically created, has a special behavior to store all the given information in its knowledge base using a semantic annotation method. After that, the AA has to (ii) wait for the bids of possible suppliers and to monitor the auction. It creates a Bid Agent (BA) for each incoming bid with semantically prepared information about the auction properties and the supplier. The bids are (iii) evaluated by the BA according to correctness and reliability of the vendor and forwarded further to the AA. In the case of a misplaced bid, the BA informs the AA and determinates itself. If more than one supplier places a suitable bid, the AA has to rank the bids.

At some point, companies must reevaluate or even eliminate suppliers that do not fit well with the buyer. This can be a huge task, depending on the number of suppliers and the obtained information. In order to do this, internal and external information will be collected and evaluated. Nevertheless, many large companies are divided into units, each with a separate purchasing sector. Sharing information across units can occur through informal meetings, strategy development sessions, newsletters, or the development of a comprehensive database containing information about potential supply sources. Internal sources, even those from diverse business units and their further experience with the suppliers, can provide a great deal of information about potential supply sources.

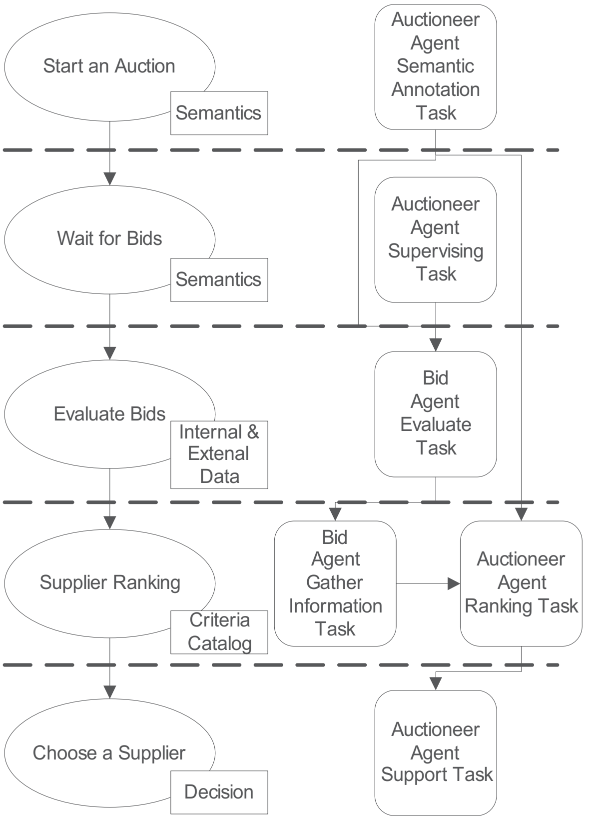


Fig. 2. Overview of the DSS-based auction process

The gathering of internal information and their representation in the knowledge base is one important task of the BA. It uses an ontology-based approach where different kinds of matchmaking strategies are combined together to provide an adaptive, flexible and efficient knowledge discovery environment (Horrocks and Li 2004, Mäkelä 2005, Bianchini et al 2008). Moreover, buyers and sellers are increasingly using the internet to help locate potential sources that might qualify for further evaluation. However, a major problem with searching in the Web today is that data available in the Web has little semantic organization beyond simple structural arrangement of text, declared keywords, titles, and abstracts (Luke and Rager 1996). The BA is using diverse semantic-based search algorithms to get information about the supplier in the World Wide Web (Sycara et al. 2003, Rao and Su 2004, Milanovic and Malek 2004). The content of Web pages is expressed by referring to ontologies, which provide a conceptual model to interpret the content (Bruijn et al. 2009). After collecting information about potential supply sources, the BA must begin to filter and consolidate the information. The information has to be arranged according to a self-defined criteria list,

defined at the start of the auction. This especially, due to the amount of data that are being gathered by the BA about different suppliers. At the end, the AA gets all the collected and arranged information from the different BAs and performs its ranking task. The decision about the ranking is documented and reported to the buyer, which is also done by the AA.
The presented Decision Support System for Supplier Ranking enhances the existing system with the BA abilities, which are used for evaluating the bids' correctness and for finding as much information as possible about the supplier. The BA agent is equipped with related behaviors to gather information, filter information, arrange them in its knowledge base and learn from this information. As information collector, this agent identifies eventual file locations about the supplier or an auction user as well as the desired amount of information by enquiring the AA. It gets the information about the buyers' preferences such as time-constraints, suppliers' reliability, and quality of products. If necessary, it uses also the sources like the Credit bureau[6] or similar sources to be able to offer reliable information.

3.3 Semantic annotation integration

The Semantically-enabled DSS provides a framework and integrated methodology consisting of a) the semantic model (describes domain, suction, suppliers' knowledge); b) tools for adapting, maintaining, and analyzing knowledge bases; c) semantic services (connection of the semantic model to the auction system repository, semantic tools) and user interfaces to view and manipulate the semantic model; and d) processes and methods for adapting SemDSS to the auction platform, setting up auctions with SemDSS. The first goal of SemDSS is to provide semantic foundations for auctions to support semantic data analysis. The semantic foundation layer is depicted in the lower part of Figure 3.

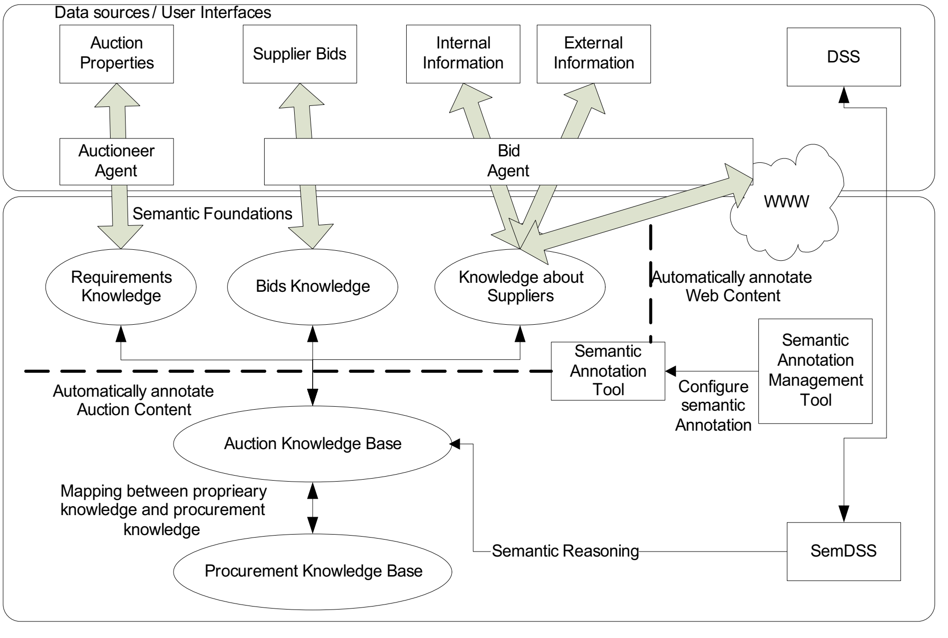


Fig. 3. Semantic Foundations for SemDSS

[6] http://www.ksv.at/KSV/1870/

3.4 Negotiation ontology

It is important for each company to define its own point of interest in a negotiation and to be able to estimate other participants. Due to the complex and dynamic nature of the VE it is hard to capture all related concepts in a persistent ontology. It is much simpler to isolate the ontology part that is only related to data exchange and communication processes associated with it. In this context an ontology was defined that offers the representation and semantics of data about negotiation and presents the link to all other concepts in the company (see Figure 4). This negotiation ontology will include a description of basic company internal concepts (order, user, product/service, interfaces, etc.).

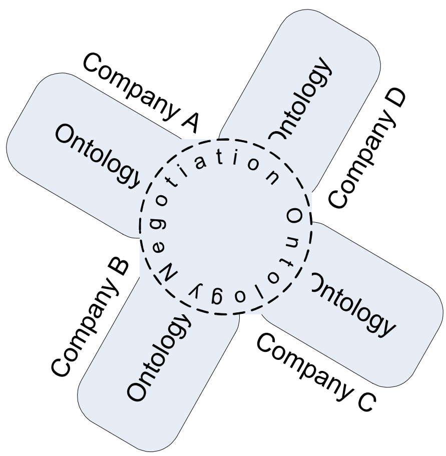


Fig. 4. Negotiation Ontology

An agent that has to make related conclusions should have knowledge about all related concepts within the domain of application. It has to know the users' economic profile (value of product/service to be sold/bought, its total cost including shipment and production, quality, etc.) and its complexity and risk profile (size of supply market, uniqueness of product, availability, etc.). Agents within the SemDSS are also externally influenced in terms of their relationships with other participants, which depend on each other considering goals to be achieved or tasks to be performed. Goals could be to buy at the lowest possible price or at the highest possible quality. Tasks could be a participation in a negotiation or the supervision of an auction. Additional factors possibly influencing negotiations with other agents could be the number of possible sellers/buyers, communication and information sharing, reputation, their reliability and the quality or quantity of the product/service. To be able to reach its goals and accomplish the tasks as good as possible a decision about an auction or strategy has to be done according to all these attributes. Consequently, it is necessary to model and represent all these concepts such as the different possible auction types, negotiation processes and rules as well as dependencies between agents and partners in the supply chain.

The negotiation ontology (see Figure 5) ensures that the agents assign the same meaning to the symbols used for messages during the negotiation process. Moreover, an ontology supports not only the communication between agents but also gives the possibility for agents to reason about the domain of application. Furthermore, it enables that agents participate in negotiations without prior knowledge of the negotiation mechanism and the ability to exchange knowledge about it reduces the amount of knowledge hardcoded in the agents (Tamma et al 2005). The presented ontology offers a representation of the general framework in detail to an agent.

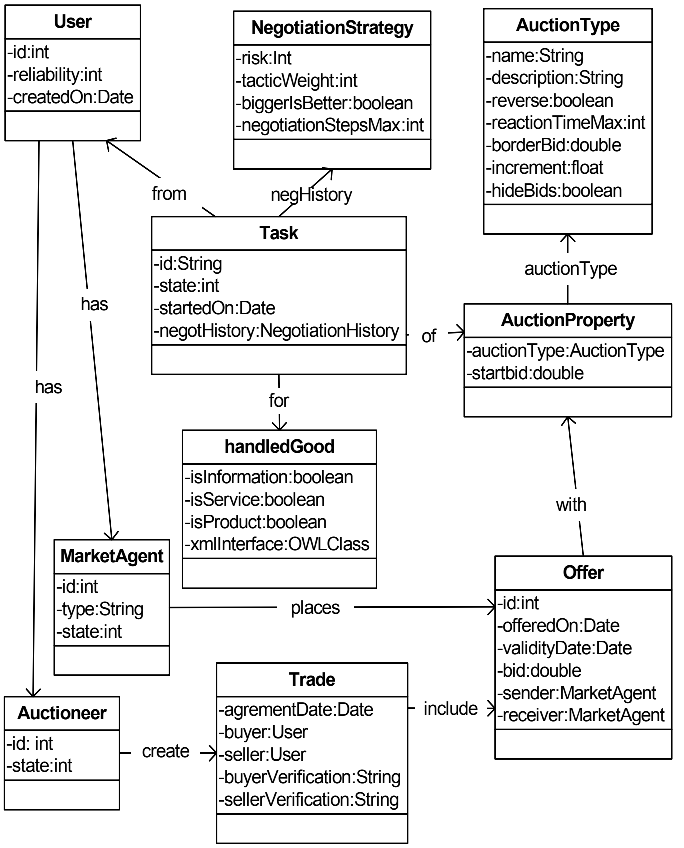


Fig. 5. Overview of the Negotiation Ontology

4. System implementation

Our MAS is based on the Java Agent Development Environment[7] (JADE) framework. The framework is used to establish the distributed agent network placed on different platforms. We also use this framework in order to support the exchange of ACL-messages between the distributed agent platforms. We used the Jess expert system shell[8] (JESS), in order to implement each agent's rule-based behavior. Every piece of information (i.e., every instance) of the ontology is mapped into a JESS fact. The JESS engine requires these facts to match them with the if-parts of rules and to execute appropriate rules. Each agent of our MAS is equipped with the negotiation ontology to share a common understanding within negotiations with agents of other platforms. The negotiation ontology can be seen as an add-

[7] http://www.fipa.org/
[8] http://herzberg.ca.sandia.gov/

on to the companies' ontologies. We used Protégé[9] to develop an agent's knowledge base. Every NA is also equipped with a graphical user interface through which the user can compete in negotiations as well as to start new ones.

Figure 6 demonstrates, how the graphical user interface (JAVA), the knowledge ontology (Protégé), the agent's rule-based behavior (JESS) and the multi-agent framework (JADE) work together in order to send a message from one agent platform to another.

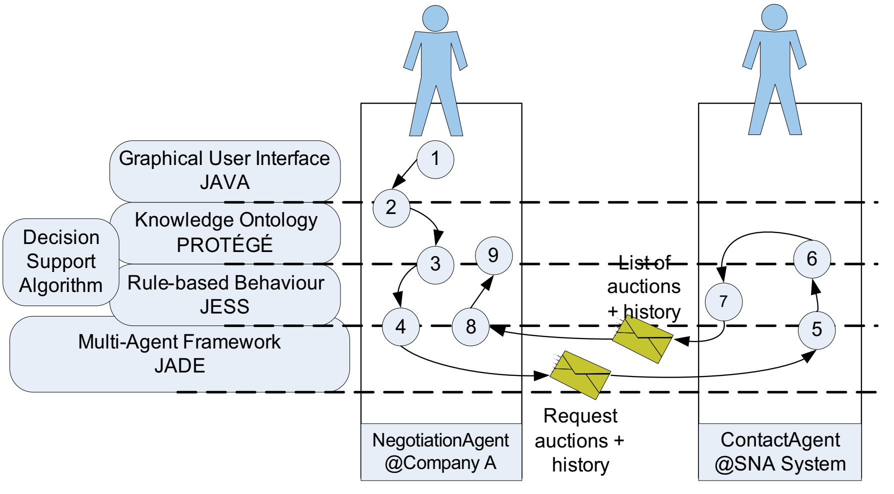


Fig. 6. A Negotiation Agent requests a list of auctions and history information

The simplified process to request negotiation and history information from the Contact Agent of the semantic negotiation system is described as following:

9. The graphical user interface (GUI) is created for the user. With this GUI the user can compete in negotiations as well as start a negotiation. In this example, the user wants to compete in an auction/negotiation because he wants to buy a specific merchandise/service. With the, the user can specify which merchandise/service he wants to buy and the user can also adjust his settings according to his maximum price as well as his price- and his quality-focus.
10. The information from the GUI is mapped to the ontology. The ontology represents the actual world of an agent. Each instance of the ontology is mapped as fact into the JESS fact base, which is part of the agent behavior. So it is possible that the agent can react on facts according to its rules.
11. Based on the facts from step 2, a rule to send a message to the CA of the Auction Framework requesting the list of available auctions and the history information for this merchandise will be executed.
12. The message from the NA to the CA is sent through the JADE runtime environment to the related CA of the semantic negotiation platform.
13. The CA receives the message and maps it to its knowledge base.
14. After mapping the message to the knowledge base, a rule will be executed which searches all available negotiations as well as the negotiation history for the merchandise, the NA wants to buy.

[9] http://protege.stanford.com/

15. The CA sends the message with the list of negotiations and the history information back to the NA.
16. The NA receives this message and maps it to its knowledge base.
17. Based on the facts of step 8, a rule will be executed which maps the information of the content of the message of step 8 to the NA's knowledge base. Another rule will be executed, which calls the Decision Support Algorithm. The DSA evaluates the possibly best auction according to the price- and the quality-focus of the user and the negotiation history for that merchandise/service.

Figure 7 is the continuation of Figure 6. It shows the simplified basic activities of the Negotiation Agent and the Auctioneer Agent in an auction or negotiation between companies.

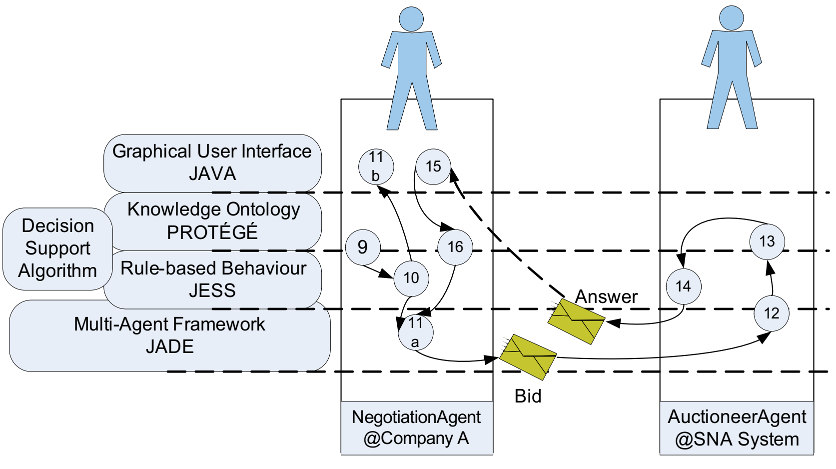


Fig. 7. The Negotiation Agent negotiates with an Auctioneer Agent

We will continue explaining with step 9 which is identical in both Figures 6 and 7.

9. The DSA evaluates the possibly best auction according to the price- and the quality-focus of the user and the negotiation history for that merchandise/service.

After step 9, the NA would send a message to the AA in order to register for this auction. The AA would answer with a call for proposal (CFP) which is the starting point for the actual negotiation with bids and answers. Because the bid and answer messages between the NA and the AA are the most important ones, they will be described in the following:

10. The NA prepares a bid for the auction he joined.
11. Step 11 is split into two tasks:
    a. The message containing the bid is sent through the JADE runtime environment to the related AA.
    b. The bid is also shown in the GUI.
12. The AA receives this message with the bid and maps it to its knowledge base and starts the Bid Agent for the gathering of information about the seller.
13. The AA compares the bid with the current price of the auction and bids from other agents, as well as the information from the NA.
14. Based on the auction type, the AA sends information about the current state of the auction to all NA which join the auction.
15. The answer from the AA is shown in the GUI of the NA.

16. According to the answer of the AA, the NA either places another bid or waits for bids of other agents and the end of the auction (if it currently has the highest bid).

This negotiation cycle goes on as long as the maximum price of a NA is not reached and as long as the runtime of the auction has not expired. If the NA places another bid, the cycle continues with step 11a.

4.1 General use case concept

In order to present our concept, we selected different companies with related products and service to build a VE. The major aspect here is that associated and related ontologies cover different concepts and workflows within these companies. Moreover, while such companies can be placed anywhere, the possibility exists that they use different words for the description of the same concepts or vice versa. The selected VE-concept is shown in Figure 8.

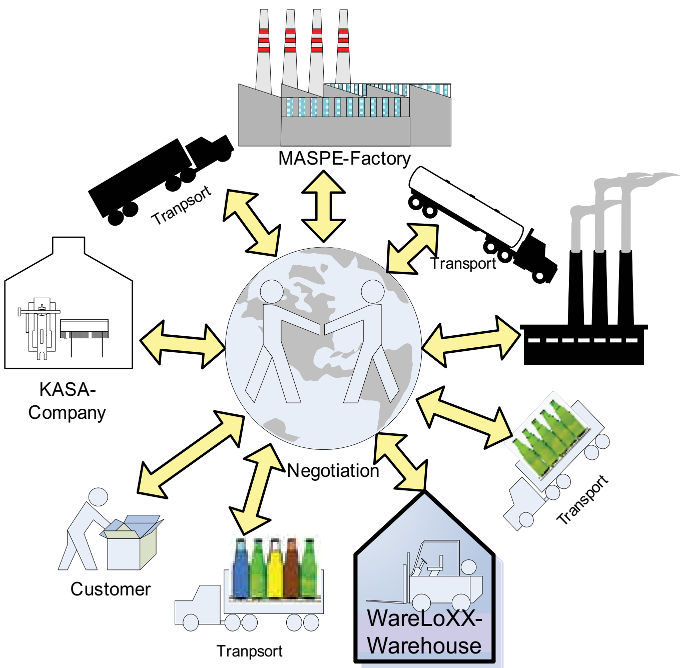


Fig. 8. Virtual Enterprise Concept

The following use-cases were separately analyzed and developed by four project groups:

- KASA-Ontology: A representation of a company for agent based assembly automation;
- MASPE-Ontology: An agent based batch processing factory for liquids;
- WareLoXX-Ontology: A warehouse system for the commitment of orders; and
- KABA-Ontology: A bottling plant for the filling of bottles combined with LiStoSys-Ontology.

The specified companies need to cooperate and negotiate with each other to be able to place products on the market. The usage of the negotiation ontology that links all the different ontologies of the different companies enables the determination of an equivalent or the semantically closest concepts.

4.2 Different companies – different ontologies

In Figure 9 two different ontologies for the handling of the term "product" are shown. On the left side of the picture an ontology for batch processes is shown. The class Product serves as a unique naming class for a product (i.e., for instance a specific amount of a pharmaceutical product) by using an ID and refers to the class Recipe, which contains all required material resources (such as raw material) and operations to manufacture this product step by step. The concepts of certain classes, such as the concept of a batch or a recipe, are derived from the relevant standard IEC 61512 Batch Control10.One the right side a product is presented as a hierarchy of subassemblies and parts together with all their properties and relationship between them. Parts are defined as components, described by a set of attributes, properties, constraints and relations to other parts. Each company in the VE has the same ontology-bases and also their own independent ontology concepts, necessary for covering their internal workflow. The negotiation ontology is used to ensure the overall understanding during communication and to enable the mapping of external information and knowledge into an internal company representation.

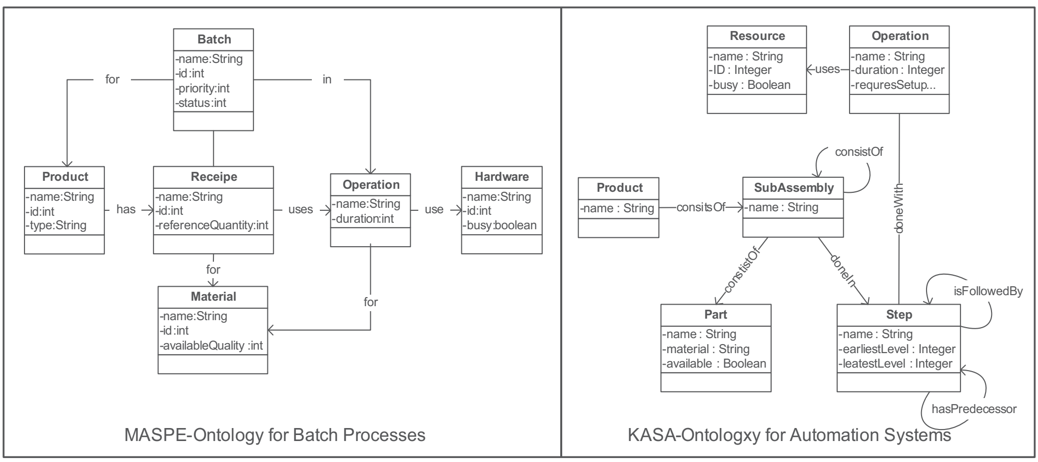


Fig. 9. Different Ontologies

We presented only a particular example the show the diversity between ontologies of different companies. Detailed examples can be found in (Koppensteiner et al. 2009).

5. Evaluation

5.1 Evaluation of intelligent agent approach in negotiation systems

This section will demonstrate the benefits of intelligent agents within the SemDSS framework as opposed to ordinary negotiation agents. The DSS negotiation agents have the ability to consider not only the price as the attribute for competing in negotiations, but also some other attributes such as quality, availability, etc. For the following two evaluations of our approach, negotiations with a start price between 30 and 70 and a runtime between 15 and 30 seconds were randomly generated. Besides the price and the runtime, all negotiations share the same attributes as quality, increment step or terms of payment. The

[10] IEC TC65, IEC 61512-1: Batch Control – Part 1: Models and terminology, International Electrotechnical Comission (IEC), 1997

negotiation agents have the task to buy products from 10 different negotiations. Their maximum price varies randomly between 70 and 100 units and changes for each negotiation they participate in. Figure 13 shows a comparison between one intelligent agent (DSS_Agent, dashed line) and three ordinary negotiation agents (Agent_1, Agent_2, Agent3). The intelligent agent chooses in which negotiation to participate in, as well as which maximum price it should offer in order to buy the products at the lowest possible price. Furthermore, the agent uses a history analysis to get the mean average product prices of the past which helps it to choose a negotiation and a maximum price.

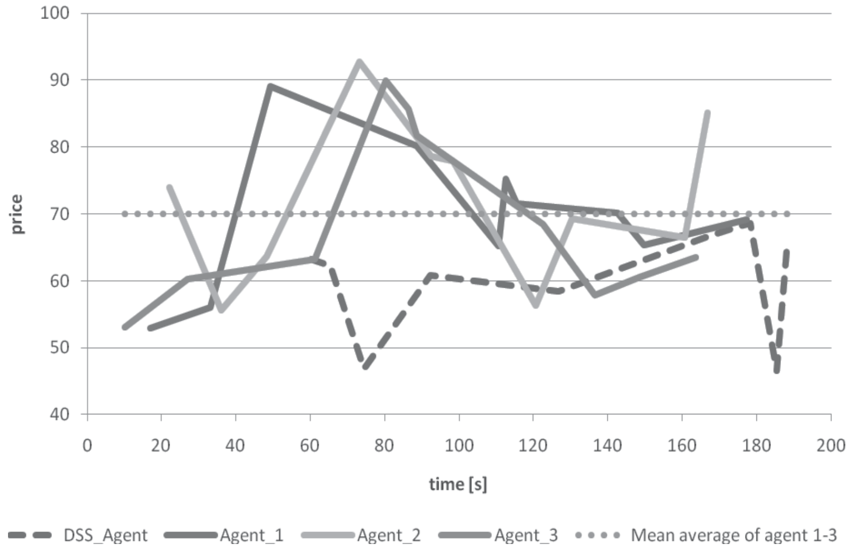


Fig. 10. Comparison between intelligent and ordinary negotiation agents

Figure 10 demonstrates that the intelligent DSS agent buys the wanted products at a cheaper price than all other agents. It might seem strange that the intelligent agent wins its first negotiation after 60 seconds compared with the other agents which win after about 10-20 seconds. This fact can be explained with the use of the DSA, the random generation of auctions as well as the random decision of a maximum price for a negotiation. The DSS agent decides to take a low maximum price for the first auctions and increases its maximum price for every lost auction. The DSA, which is used to evaluate in which negotiation the intelligent agent should compete in, currently does not consider if the products are needed urgently or not. An effective way to reduce the time until the intelligent agents wins its first negotiation would be to add a time factor (are the products needed urgently or not) to the DSA or simply to select a higher maximum price for the first auctions. The figure also shows that the mean average price of negotiations of the three ordinary agents (dotted line) is higher than each single negotiation of the intelligent agent.
In order to prove that the intelligent agent with the SemDSS framework gives advantages in negotiations, the evaluation shown in Figure 10 has been repeated 10 times and the mean average price of each negotiation agent for each repetition can be seen in Figure 11. The figure clearly demonstrates the advantage of an intelligent agent as opposed to ordinary negotiation agents. The DSS agent wins negotiations on an average with a 19.87% lower price than the other negotiation agents do.

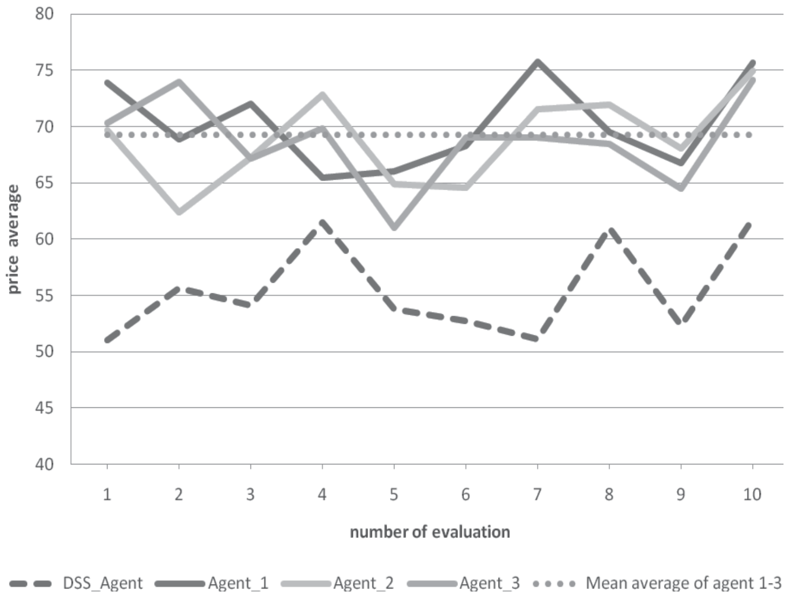


Fig. 11. Comparison between intelligent and ordinary negotiation agents

As already mentioned, the DSS negotiation agents are able to consider more than just one attribute of a negotiation. Figures 12 and 13 deal with a comparison between a price-focused, a quality-focused and a price- and quality-focused DSS agent. The intelligent agent considers the price as well as the quality of products, weights and ranks them and chooses the best suitable negotiation according to the ranking. For the evaluation, the system created random negotiations with a quality between 1 and 10 and a random price between 20 and 60. Negotiations with products of higher quality tend to have a higher price than negotiations with products of lower quality. Each of the three agents will participate in negotiations as long as it wins 10 of them. To reach more significance, this process has been repeated 10 times and the mean average price and quality values of the negotiations an agent won can be seen in Figures 12 and 13.

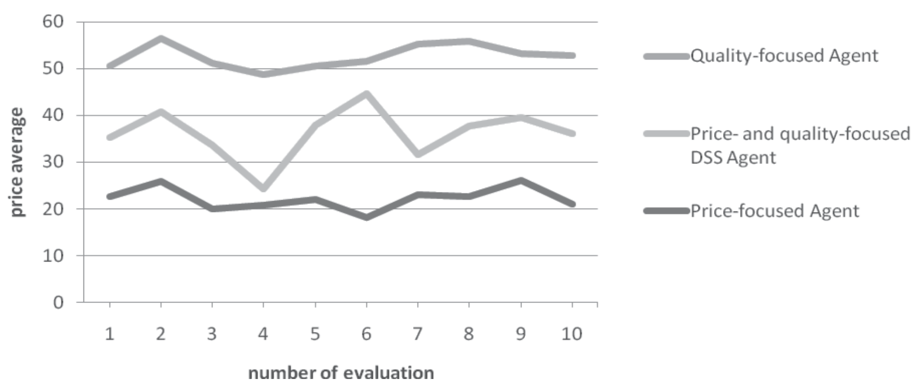


Fig. 12. Comparison between quality-, price-, and quality- and price-focused agents

Figure 13 shows the mean average prices of the three different agents. The mean average price of the multi-attribute focused agent is 36.14 as compared with 22.2 (price-focused agent) and 52.61 (quality-focused agent). These numbers alone are not remarkable, but they become remarkable, if we also consider the average quality of the products in Figure 13. The price- and quality-focused DSS agent's average product quality is 6.4 in relation to 2.2 (price-focused agent) and 8.1 (quality-focused agent). The data shows that the intelligent multi-attribute focused agent buys products 31.3% cheaper but only with a 21.68% lower quality than the quality-focused agent. The price- and quality-focused agent buys products on an average of 62.81% more expensive but with a 191.14% better quality.

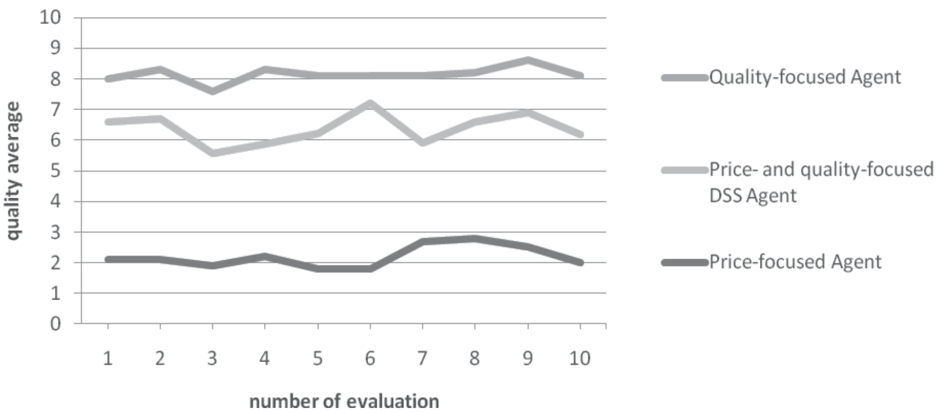


Fig. 13. Comparison between quality-, price-, and quality- and price-focused agents

The evaluations clearly demonstrated the advantage of an intelligent multi-attribute focused DSS agent in comparison to ordinary negotiation agents.

6. Conclusion

Most procurers want suppliers that can offer all of products and services defect-free, delivered just in time (or as close to this ideal as reasonably possible) and reliable. A supplier rating is needed to determine which suppliers are capable of coming satisfactorily close to this and thus to be retained as current suppliers. The procedure of a multifactor comparison, the combination of valuating the individual subjective criteria and the pricing and quality factors must often be done manually, which is error-prone and time intensive. Depending on the chosen level of complexity, usable tools range from spreadsheets to cost intensive and highly complex extensions of ERP-Systems, e.g. SAP. A solution supporting the user on the one hand in all automatable steps and on the other hand being flexible enough to implement the different methods of supplier rating is strongly needed, which at the same time should be achievable also for small and medium sized companies. Intelligent agent technology is recognized as a promising approach for enhancing a system's connectivity, extensibility, reusability and for dealing with its complexity, autonomy and heterogeneity. However, typically semantic gaps exist in the exploitation of current agent systems considering the fact that most times just syntactic matching is used for detecting equalities between exchanged data during the communication process. Ontologies can be used for representing concepts and relationships embedded in the negotiation environment so that they are semantically meaningful to a software agent, ensuring that agents refer to exactly the same object. In this chapter we combine multi-agent technology with ontology-driven solutions to build the Semantic-enabled Decision Support System, respectively SemDSS, which is able to support and partly automate procurement processes. We introduce semantic foundations to bridge relevant semantic gaps in order to radically improve the automated data processing in semantically heterogeneous environments. Our approach offers solutions for solving the interoperability problem in the procurement process.

Moreover, such a system increases the overall process efficiency and helps minimizing subjective judgment. In the last section, we presented the advantages when using SemDSS agents in automated negotiation. Using a history analysis and by slowly increasing its maximum offer after every lost auction, the DSS agent was able to win negotiations on an average lower price than the other negotiation agents. Besides, the SemDSS negotiation agents have shown benefits due to their ability to consider more than just one attribute of a negotiation.

Our future work will be concerned with the integration of the presented system in a real company framework as well as with the demonstration of the achieved results. Furthermore, we will focus on ontology merging and mapping, since it is complementary with our approach.

7. References

Aberdeen Group, The E-Procurement Benchmark report, August 2006.

Ai W.G., Sun J., and Li H., A distributed marketing decision support system based on multi-intelligent-agent, Machine Learning and Cybernetics, *Proceedings of 2004 International Conference*, Aug. 2004, Vol 1, pp. 233-238.

Alvarado M., Sheremetov L., Bañares-Alcántara R., and Cantú-Ortiz F., Current challenges and trends in intelligent computing and knowledge management in industry, *Knowl. Inf. Syst.*, vol. 12, 2007, S. 117-127.

Baclawski K., Kokar M., Kogut P., Hart L., Smith J., Holmes W., Letkowski J., Aronson M., (2001) Extending UML to Support Ontology Engineering for the Semantic Web, *Proceedings of the 4th International Conference on UML*, Springer, pp. 342-360

Beam C. and Segev A., Automated Negotiations: A Survey of the State of the Art, *WIRTSCHAFTSINFORMATIK*, vol. 39, 1997, pp. 263--268.

Bergamaschi S., Castano S., Vincini M., (1999) Semantic integration of semistruc-tured and structured data sources. *SIGMOD Rec.* 28, 1 (Mar. 1999), 54-59

Bianchini D., Antonellis V., and Melchiori M., Flexible Semantic-Based Service Matchmaking and Discovery, *World Wide Web*, vol. 11, 2008, S. 227-251.

Biffl, S., Sunindyo, W., and Moser, T., Semantic Integration of Heterogeneous Data Sources for Monitoring Frequent-Release Software Projects, in *4th International Conference on Complex, Intelligent and Software Intensive Systems* (CISIS 2010), Krakow, Poland, 2010, pp. 360-367.

Bouzguenda L., Bouaziz R., and Andonoff E., Dynamic Plugging Of Business Processes In Cross-Organizational Workflow, *International Journal of Computer Science and Applications*, vol. 5, 2008, pp. 141-164.

Bui T. and Lee J., An agent-based framework for building decision support systems, *Decis. Support Syst.*, vol. 25, 1999, S. 225-237.

Bundesverband für Materialwirtschaft, Einkauf und Logistik e. V., Marktstudie: Trends im Einkaufs-Outsourcing und eProcurement, 15.07.2003, Frankfurt

Bruijn J., Kerrigan M., Zaremba M. and Fensel D.:Semantic Web Services, Staab S. and Studer R., (Eds), *Handbook on Ontologies*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

Calero C., Ruiz F., Piattin F., (2007) *Ontologies for Software Engineering and Technology*, Springer

Chandrasekaran B., Josephson J. R. and V. R Benjamins, (1999) What Are Ontologies, and Why Do We Need Them? *IEEE Intelligent Systems*. 1999

Cimiano P., Handschuh S., Staab S., (2004) Towards the Self-Annotating Web. In: *Proceedings of the 13th International World Wide Web Conference*, WWW 2004, New York, USA

David E., Azoulay-Schwartz R., and Kraus S., Bidding in sealed-bid and English multi-attribute auctions, *Decision Support Systems*, vol. 42, Nov. 2006, pp. 527-556..

Davies J.. Semantic Web Technology, (2006) Trends and Research in Ontology-based Systems: Trends and Research. Wiley & Sons

Eilbeck K., et al. (2005) The sequence ontology: a tool for the unification of genome annotations. *Genome Biol.* 6, R44

Fonseca F., Egenhofer M., Davis C., (2000) Ontology-Driven Information; *The AAAI 2000 Workshop on Spatial and Temporal Granularity*

Happel H.J., and Seedorf S., (2006) Applications of Ontologies in Software Engi-neering, *Proceedings of the Workshop on Semantic Web Enabled Software Engineering* (SWESE)

He, M., Jennings, N. R. and Leung, H. (2003) On agent-mediated electronic commerce. *IEEE Trans on Knowledge and Data Engineering*, 15 (4). pp. 985-1003.

Hepp M., De Leenheer P., De Moor A., Sure Y., (2007) Ontology Management. *Semantic Web, Semantic Web Services, and Business Applications*, Springer

Hikkanen A., Kalakota R., Saengcharoenrat P., Stallaert J., Whinston A.B., Distributed decision support systems for real time supply chain management using agent technologies, *Readings in Electronic Commerce* (1997) 275–291.

Horrocks, I. and Li L.: A Software Framework for Matchmaking Based on Semantic Web Technology. *International Journal of Electronic Commerce*. Vol. 8, Iss. 4, 39--60 (2004)

Houari N., and Far B.H., An intelligent project lifecycle data mart-based decision support system, *Electrical and Computer Engineering, Canadian Conference*, 2-5 May 2004, Vol 2, pp. 727-730.

Jain V, Wadhwa S, Deshmukh SG. 2009. Revisiting information systems to support a dynamic supply chain: issues and perspectives. *Production Planning & Control: The Management of Operations* 20(1):17-29.

Jennings N.R., An agent-based approach for building complex software systems, *Commun. ACM*, vol. 44, 2001, pp. 35-41.

Kersten G.E. and Lai H., Negotiation Support and E-negotiation Systems: An Overview, *Group Decision and Negotiation*, vol. 16, 2007, S. 553-586.

Koppensteiner G., Merdan M., Hegny I., and Weidenhausen G., A change-direction-algorithm for distributed multi-agent transport systems, *Mechatronics and Automation, 2008. ICMA 2008. IEEE International Conference on*, 2008, pp. 1030-1034.

Koppensteiner G., Merdan M., Lepuschitz W., Vittori L. and List E., Ontology-oriented Framework for Virtual Enterprises. *International Conference on Knowledge-Engineering and Ontology-Management*, October 6 - 8, Madeira, 2009, Portugal

Koppensteiner G., Merdan M., Lepuschitz W., Reinprecht C., Riemer R., Strobl S.: 2009a, A Decision Support Algorithm for Ontology-Based Negotiation Agents within Virtual Enterprises;: *Future Information Technology and Management Engineering FITME 2009*, Sanya, China;

Kiryakov A., Popov B., Terziev I., Manov D., Ognyanoff D., (2004) Semantic annotation, indexing, and retrieval, *Web Semantics: Science, Services and Agents on the World Wide Web*Volume 2, Issue 1, December 2004, Pages 49-79

Kowalczyk, R., Ulieru, M., Unland, R.: Integrating Mobile and Intelligent Agents in Advanced E-commerce: A Survey. In: *Agent Technologies, Infrastructures, Tools, and Applications for E-Services, Proceedings NODe'2002 Agent-Related Workshops*, Erfurt, Germany. LNAI 2592, Springer-Verlag, pp.295-313, 2002.

Kuikka S., A batch process management framework: Domain-specific, design pattern and software component based approach, PhD thesis. Helsinki University of Technology, 1999. , 15

Lee C., Lau H., Ho G., and Ho W., Design and development of agent-based procurement system to enhance business intelligence, *Expert Systems with Applications*, vol. 36, Jan. 2009, S. 877-884.

Luke S. and Rager D., Ontology-Based Knowledge Discovery on the World-Wide Web*, The 13th National Conference on Artificial Intelligence* (AAAI96), 1996, S. 96--102.

Mäkelä, E.: Survey of semantic search research. In: *Proc. of the Seminar on Knowledge Management on the Semantic Web* (2005)

Merdan M., Zoitl A., Koppensteiner G., Demmelmayr F., Semantische Technologien – Stand der Technik, e&i Elektrotechnik und Informationstechnik, IN PRESS

Merdan M., Knowledge-based Multi-Agent Architecture Applied in the Assembly Domain, PhD Thesis, Vienna University of Technology, 2009.,

Merdan M., Koppensteiner G., Hegny I., and Favre-Bulle B., Application of an ontology in a transport domain, *IEEE International Conference on Industrial Technology*, 2008. ICIT 2008, 2008, pp. 1-6., 20

Michelson M. and Knoblock C. A., (2007) An Automatic Approach to Semantic Annotation of Unstructured, Ungrammatical Sources: A First Look, In *Proceedings of the 1st IJCAI Workshop on Analytics for Noisy Unstructured Text Data* (AND-07), Hyderabad, India

Milanovic N. and Malek M., Current solutions for Web service composition, *Internet Computing, IEEE,* vol. 8, 2004, S. 51-59.

Miller G. A., (1995) WordNet: a lexical database for English, *Communications of the ACM*, v.38 n.11, p.39-41

Moser T., Semantic Integration of Engineering Environments Using an Engineering Knowledge Base, PhD Thesis, Vienna University of Technology, 2009.

Moser, T.; Biffl, S.; Sunindyo, W., and Winkler, D. Integrating Production Automation Expert Knowledge Across Engineering Stakeholder Domains *Proc. 4th International Conference on Complex, Intelligent and Software Intensive System*s (CISIS 2010), IEEE Computer Society, 2010, 352-359.

Neubert R., Gorlitz O., Teich T., 2004.Automated negotiations of supply contracts for flexible production networks. *International Journal of Production Economics* 89(2), pp. 175–187.

Pastor Ó., España S. and González A., (2008) An Ontological-Based Ap-proach to Analyze Software Production Methods, Information Systems and e-Business Technologies, *2nd International United Information Systems Conference UNISCON* 2008 Klagenfurt, Austria, April 22–25, Proceedings, Vol. 5, pp. 258-270

Preist C., Jennings N.R., Bartolini C., and Bartolini C., A Generic Software Framework for Automated Negotiation, IN: *PROCEEDINGS OF SELMAS'2004*, LNCS 3390, vol. 2004, 2002, pp. 213--235.

Puigjaner L.,Guillen-Gosalbez G., Mele F.D.,2008. Multiagent framework for modeling and design of supply chain networks. In: Papageorgiou, Lazaros G., Georgiadis, MichaelC. (Eds.), *Supply Chain Optimization*, Part1. Wiley-VCH, pp. 45–83.

Rao J. and Su X., A Survey of Automated Web Service Composition Methods, In *Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition*, SWSWPC 2004, 2004, S. 43--54.

Renna P. and Argoneto P., Production planning and automated negotiation for SMEs: An agent based e-procurement application, *International Journal of Production Economics*, vol. 127, Sep. 2010, S. 73-84.

Rinderle, S. and Benyoucef, M. (2005) Towards the Automation of E-Negotiation Processes Based on Web Services - A Modeling Approach. In: *Proc. Int'l Conf. on Web Information Systems Engineering* (WISE'05), New York.

Rouwenhorst B., Reuter B., Stockrahm V., van Houtum G.J., Mantel R.J., and Zijm W.H.M., Warehouse design and control: Framework and literature review, *European Journal of Operational Research*, vol. 122, May. 2000, pp. 515-533.,

Staab S., Maedche A., Handschuh S., (2001) An Annotation Framework for the Semantic Web, In: S. Ishizaki (ed.), *Proc. of The First International Workshop on Multi-Media Annotation*. January. 30 - 31, Tokyo, Japan

Sünder C., Zoitl A., and Dutzler C., Functional structure-based modelling of automation systems, in *Int. J. Manufacturing Research*, vol. 1, iss. 4, pp. 405-420, 2006. 16

Sycara K., Paolucci M., Ankolekar A., and Srinivasan N., Automated discovery, interaction and composition of Semantic Web services, *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 1, Dez. 2003, S. 27-46.

Tamma V., Phelps S., Dickinson I., and Wooldridge M., Ontologies for supporting negotiation in e-commerce, *Engineering Applications of Artificial Intelligence*, vol. 18, Mar. 2005, pp. 223-236.

Tamma V., Wooldridge M., Blacoe I., and Dickinson I., An Ontology Based Approach to Automated Negotiation, *Agent-Mediated Electronic Commerce IV. Designing Mechanisms and Systems*, 2002, pp. 317-334.

Turban, E., Aronson, J. E., and Liang, T. (2005) Decision Support Systems and Intelligent Systems (7th edition), Prentice Hall Publication, pp. 223 and pp. 109, 707.

Van den Berg J.P., A literature survey on planning and control of warehousing systems, *IIE Transactions*, vol. 31, 1999, pp. 751-762.,

Wang X.F., Chen Y.Q., and Liu Y.L., Web-oriented warfare command decision support system based on agent and data warehouse, *Cyberworlds, 2005, International Conference*, 23-25 Nov. 2005, pp. 6-11.

Yuan Y., Liang T.P., Zhang J.J., Using Agent Technology to Support Supply Chain Management: Potentials and Challenges,Working Paper, University Hamilton, 2002.

Zhang H. Application of Multi-agent Technology to Information Systems: An Agent-based Design Architecture for Decision Support Systems, *Australasian Journal of Information Systems*, Vol 15, No 2 (2009)

Zhang Q., and Yu J.H., Bidding Decision-Making Support System for Power Supplier Based on Multi-Agent System, *Power System Technology, PowerCon 2006. International Conference*, Oct. 2006, pp. 1-6.

Zhang Z.Q., and Xi B., Research on Framework of Decision Support System for Partner Choice and Measurement in Supply Chain, *Machine Learning and Cybernetics, Proceedings of 2005 International Conference*, 18-21 Aug. 2005, Vol 1, pp. 378-382.

13

Intelligent Collaboration Environment in Multi-Agent System Enabling Software Dynamic Integration and Adaptive Evolving

Qingshan Li, Lili Guo, Xiangqian Zhai and Baoye Xue
Xidian University
China

1. Introduction

Along with the quick development of Internet, network techniques and software techniques, computing development has the remarkable characteristics of distribution, maneuverability and heterogeneous nature. How to expand a system quickly and how to integrate a legacy system into a new one have been more and more concerned. System integration is combined with independent units into a system with harmonious function and intimate connection, rather than simple accumulation of equipments. For lack of integration standards, some problems are brought to the systems inter-connectivity and interoperability. So the information of one system could not be shared by others, the services provided also could not be used and people have to develop a new one. Component-based technique is one of the main means for integrated problem, but it can not meet the need of autonomy, intelligence, dynamic property and domain-oriented extensibility.

The rapidly changing environment of internet demands that system integration development should dynamically adapt and actively react to the changing requirements and contexts. However, in the traditional component-based integration technologies, the integrated components are statically bound, and the collaboration mode among them is fixed so that it can not be adjusted and modified especially when system integration is performing. Therefore, these technologies are unable to adapt to the frequently changing requirements and environments.

Though the traditional component-based integration approach has brought some solutions towards system integration of information systems, the integrated components is statically bound and the collaboration mode of them is single and fixed, it can not be adjusted and modified especially when the system is running. As above described, in the Internet computing environment, the requirements are variable and the computing contexts are dynamic and changeable，which demands that the collaboration mode of the integrated systems can be dynamically and flexibly modified to rapidly respond to those changes. Therefore, the traditional component-based integration approach can not meet this demand to handle those changes.

With the emergence of agent technology, a new approach based on agent was proposed in system integration flied which can resolve the problem above well which the traditional component-based integration approach can not handle. Through wrapping legacy systems and newly-developed systems into agents which has the characteristics of intelligence, autonomy, and adaptability, the system integration becomes the integration of agents which can flexibly handle the changes. In this paper, based on agent technology, the agent dynamic discovery and collaboration mechanism is attained. Additionally, through the application of Contract Net Protocol into the cooperation among agents, the dynamic bidding of agent is implemented and the adaptive integration which can adapt to the variable environment is gained. Finally, with script-described integration requirement strategy, the flexible requirement transition which does not interrupt the running system is achieved by script switching, which accomplishes dynamic evolution of system integration. Based on above, the dynamic and flexible integration approach is achieved which perfectly handles the environment and requirement changes.

Agent technology provides a new way to resolve the above problems.Agents are intelligent, flexible and autonomous so that they are capable of adapting intelligently to the environment changes. In this chapter, the agents are applied to system integration to achieve dynamic system integration and adaptive evolving, through packing the sub-systems into the agents and the communication and collaboration among the agents, the software dynamic integration and adaptive evolving is implemented. More specifically, the agents in intelligent collaboration environment are based on Agent Model proposed in first section which makes the agents active and intelligent, so that the overall system is able to dynamically and immediately perceive the presence and disappearance of agents. The agents' collaboration mechanism is based on Contract Net Protocol and the related algorithms which enables the agents to cooperate more flexibly and intelligently through bidding and tendering. Furthermore, the integration logic which depicts the integration requirement is expressed as the script , with the interpretation of the script, the agents collaborate with each other in terms of the script and thus the system integration is performed dynamicall, especially, when the integration requirement is changed, the new script is designed after that, with the the interpretation of the new script, the new system integration is accomplished and the new requirement is satisfied, the script-based integration control mechanism makes the flexible integration transition which copes with the frequently changing requirements perfectly.

Agent technologies and multi-agent systems have also been applied to some actual systems, i.e. Agile Manufacturing system, Air traffic management system (Rabelo et al., 1998) and e-commerce systems. After the original CNP was proposed in (Smith & Davis, 1980), many researchers had made much effort to refine and extend it with more flexibility, adaptability and better performance. Some most typical works as shown here: T. Sandholm proposed a formal model (Sandholm, 1993), whose pricing mechanism generalizes CNP to work for both cooperative and competitive agents. By focusing on the issue of self-interest agents in automated negotiation system, Sandholm extended CNP to Leveled Commitment Contract (Sandholm & Lesser, 1996), which allows agent to end an ongoing contract if an outside offer is more profitable than current contract. Besides, there were some endeavors on formalization of CNP, e.g. M. d' Inverno and M. Luck (Inverno & Luck, 1996), Werner (Werner ,1989), M. Fisher & M. Wooldridge (Fisher & Wooldridge, 1994). By adding rejection and confirmation communicative action, FIPA made the original CNP an important standard for a whole range of prominent agent platform implementations. The newest extension for this standard is FIPA iterated CNP (ICNP). It allows multi-round iterative bidding . In order to enhancing CNP, T. Knabe (Schillo, Fischer, Kray,2000) proposed CNCP which is another helpful extension of CNP.

In the actual application, agents are being placed in dynamic network environment, and their kn- owledge, capability and overload will be undergoing dramatic changes with the task execution. Howev- er, a single agent cannot get a global knowledge of the task complexity and the states of other agents. Additionally, the manager should coordinate the relationship among agents according to the agents' ability. When selecting an agent as the successful bidder, the manager should consider some agent's successful experience of the task execution and give chances to those agents who lack experience from the task execution. Based on the above considerations, we introduce the perception coefficient and the degree of credibility into the traditional contract net protocol and propose the Contract Net Model based on Agent Initiative Perception (CNMAIP). CNMAIP is applied to the Agent-based System Integration Tool (ASIT), and has been tested with the Border and Coast Defense Simulation System. The result demonstrates that, CNMAIP not only increasingly self- adapts dynamically to the change of the agent's ability and the environments, but also improves the negotiation efficiency on the basis of sufficient ensuring negotiation quality.

Based on the above model, strategies and mechanism, the dynamic integration and adaptive evolving architecture is designed and the intelligent collaboration environment in multi-agent system is accomplished which involves Agent Wrapper, Script Design Tool and Run-time Support Platform and so on. The environment provides support not only for the integration development phase but also for the run-time management phase of system integration. Finnaly, the intelligent collaboration environment is demonstrated well by the integration simulation experiments of Border and Coast Defense System and the experiment result indicates that the dynamic software integration and adaptive evolving is achieved.

This chaper is divided into three sections: Agent Model, Dynamic Integration and Adaptive Evolving Architecture,Agent Collabration Mechanism and Algorithms. Agent Model section introduces the definition of Agent(includes Function Agent and Service Agent) and inner mechanism, the integration units is wrapped into the agents which is active and intelligent, Agent Model is the foundation of Agent Wrapper. Dynamic Integration and Adaptive Evolving Architecture section depicts the whole architecture and integration development environment , which consists of several tools and platforms. The last section decribes the Agent Collaboration Mechanism and Algorithms which enables the intelligent collaboration among the agents.

2. Agent model

Traditional agent models are divided into three types: reactive agent model, rational agent model and a hybrid of the above. Brooks first developed the idea of reactive agent model (Brooks, 1991) and proved its practical utility with robots experiments. The task strategy of reactive agent model is hierarchy. The low level tasks are atom behaviours which are similar to the human stress response. The high level tasks are composed task and can be divided into low level tasks. The reactive agent model is quite efficient in task execution. Although the reactive model embodies the process of reaction to its surrounding, it is often passive and is low in adaptability and portability. The rational agent model, originated from artificial intelligence, focuses on formal logic to describe agent (Cohen & Levesque, 1990). The most influenced is Anand S. Rao and Michiael P. Georgeff's BDI model (Rao & Georgeff, 1991). BDI model has a sound theoretical foundation, and many researchers and scholars, such as Shi Chunyi (ZHANG & SHI, 2002) and Shi Zhongzhi (DONG et al.,2004) in China, have made expansion and application on it. Core idea of BDI is its internal goal oriented reasoning and planning process which is expressed in Belief, Desire and Intention. BDI model performs well in agent intelligence, but it is difficult to implement and has a certain distance for practical application.

In recent years, Michael Wooldridge put forward the VSK agent model (Wooldridge & Lomuscio, 2000). It describes agent in the concepts of Visible, Perceive and Know. The VSK model also belongs to the rational agent model. However, it does not specify its internal structure, while it places emphasis on agent semantics under the MAS environment.

In our view, an agent is a collection of primitive components that provide a focused and cohesive set of capabilities. To obtain the domain-oriented extendibility and smart features, we design two categories of agent, Service agent and Function agent, which are short as SA and FA correspondingly. FA is a concrete entity (functional component) which possesses some basic capabilities while SA is an abstract entity which organizes the FAs together in terms of certain collaboration logic to provide a high level service. The semantic model is showed as Agent=<*Aid, Type, Mod, Des, Caps, Acq, Mq, Mh, Co, Rules*>.Of which:

Aid=<*ip, port, name*> is defined as the identifier, including the host machine IP and communications port, as well as the agent's name;

Type- represents the type of agent, namely, Service and Function, which are equivalent to SA and FA correspondingly;

Mod- represents the functional components whose functions are its capacities;

Des- describes the basic information of the agent;

Caps- describe the set of agent's capacities;

Acq- describes the agent's acquaintances. Each of the acquaintances includes its aid, and capability information. This attribute aims at SA, the acquaintances are its organized FAs. FA does not own this attribute.

Mq- represents message queue, which caches the received messages;

Mh- represents message processor, which analyses and encapsulates message content;

Co- represents agent collaboration engine, which mainly interprets and implements the service process of Service Agent;

Pe- represents agent sensor, which senses the related ability and new published tenders on CMB;

Rules=<$rule_1$, $rule_2$,…, $rule_n$> expresses the rules database of agent, which stores the integration rules of system integration. The behaviour of Agent is determined by the cooperative relationship with other Agents described in the integration rules.

In the model above, *Caps* and *Acq* behalf the behavior of agent.The *Caps* contains two types of capability: atomic capability and composite capability, corresponding to the capabilities provided by FA and SA. An atomic capacity can accomplish a specific task without the collaboration with the others, while a composite capacity is the combination of atomic capabilities that are not all provided by the same agent. The *Acq* describes the agent's relationships with other agents in the society, and its behaviours about the capabilities and addresses of those agents.

There are two message queues: incoming-message-queue and outgoing-message-queue corresponding to two threads respectively. *Mh* processes incoming messages from *Co* or *Pe,* putting them into the outgoing-message-queue. For each message awaiting dispatch, *Mh* queries the message object for the intended recipient. For each message in the incoming-message-queue, *Mh* extracts its content. If the content is simple, e.g. the response to the registration of an agent from CRC (Capability Register Center) or AMS(Agent Management Service), it is handled by the *Mh* itself. Other messages, e.g. tasks to be executed, are put into the task queue or the buffer in *Pe*.

The Rules and the *Co* are cores of an agent. The role of the former is to construct action sequences that achieve desired input goals. The role of the latter is to manage its problem solving behaviours, particularly those involving multi-agent collaboration.

The *Pe* can perceive the changes in the external environment. It can realize what new capabilities have increased in CRC and perceive tenders issued by CMB (Common Message Blackboard). With the help of the *Pe* in the control platform, it enhances the pro-active of agent to obtain more acknowledge about agent in the system, so as to make a better decision.

The message handler processes incoming messages from Co-ordination Engine or Sensor, putting them into the outgoing-message-queue. For each message awaiting dispatch, the communicator queries the message object for the intended recipient, and looks up a local address book for the recipient's address. If the address is found, the message is put into the incoming-message-queue. On the contrary, the communicator stores the message object onto a holding buffer, and queries known agents for the required address. Once the message recipient's address is received, the communicator removes the relevant message from the holding buffer and proceeds to dispatch the message. In the event that no address is found or network communications fails, a suitable error message is generated, which the communicator adds to the reader's incoming-message-queue to be processed as a normal incoming message.

For each message in the incoming-message-queue, the communicator extracts its content. If the content is simple, e.g. the response to the registration of an agent from CRC or AMS, it is handled by the communicator itself. Other messages, e.g. tasks to be executed, are put into the task queue or the buffer in Sensor.

Actually the message handler is an agent's internal message sorting office, continually checking the incoming-message-queue for new messages, and forwarding them to the relevant components of the agent..

The role of an agent's Co-ordination Engine is to manage its problem solving behaviours, particularly those involving multi-agent collaboration. The role of the Planner is to construct action sequences that achieve desired input goals. It controls a single-agent planning mechanism. Currently, a planner based on a plan library which provides logical descriptions of planning tasks known to the agent is provided where plans are hierarchical skeletal structures the nodes of which may be either new subplans, or executable subtasks. The Co-ordination Engine and the Planner are cores of an agent. They contain the desire and intention of an agent. When processing a task, first the Co-ordination Engine invokes the Planner to decompose the task into subtasks, and then the Co-ordination Engine manages the execution of the subtasks. If one subtask cannot be finished by an agent itself, the agent asks a favour of its acquaintances. Once one acquaintance is found to aid in completing the subtask, the agent will send a message to it to request executing the subtask; otherwise a tendering is initiated by the agent. At the same time, the subtask is put into the internal buffer to prevent the subtask thread from running.

The Sensor can perceive the changes in the external environment. It can realize what new abilities has increased in CRC, if some of the ability is related to its capability, it will put the owners of the ability as its acquaintances into AD. In addition, it also can perceive tenders issued by PIB, get the tender and measure its capability, then decide to bid or not. The application of Sensor will increase the agent's initiative, and therefore the agent can get more knowledge of other agents in the system, so that the agent can make a better decision. With the help of Sensor and PIB in the control platform, the Contract Net Model based on Agent Initiative Perception introduced in next section is realized.

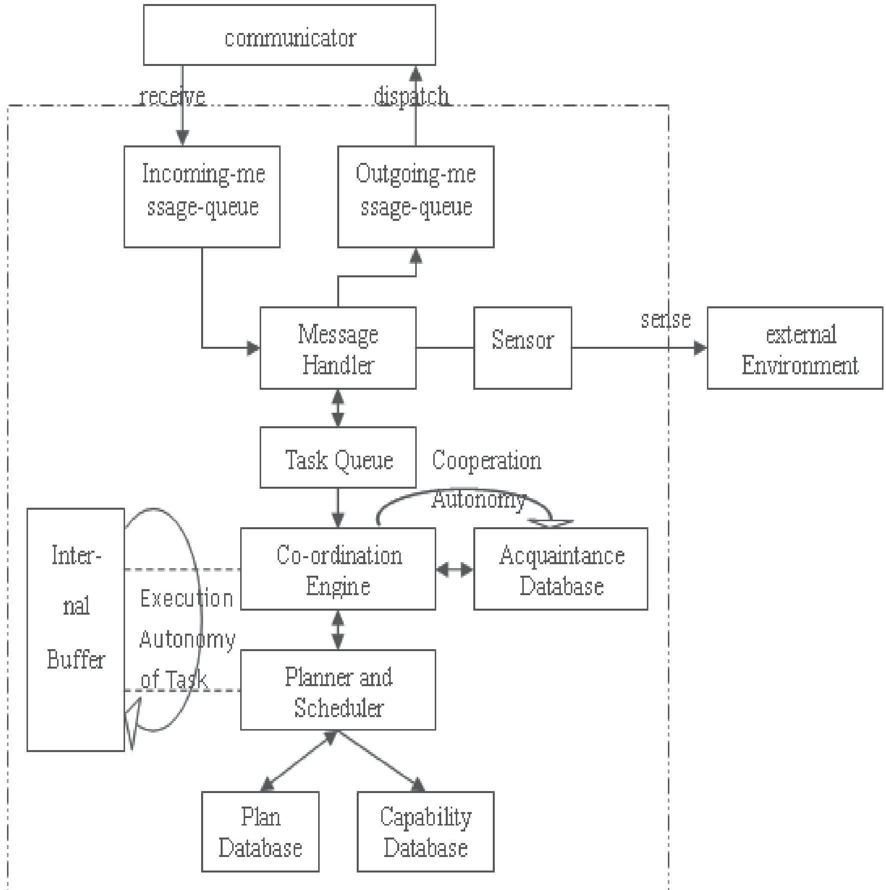


Fig. 1. Agent Model

Agent unit have the characteristic of intelligence and initiative, that's can work independently on its own knowledge, it also has the ability to actively find company. When the agent receives a task, it will call its own planner/scheduler to decompose capacity into several sub-capacities. When it find some sub-capacity do not exist in its own capacity database, the agent will give the sub-capacity to its co-ordination engine. Firstly the co-ordination engine find the acquaintances in the acquaintance database whether they possess this sub-capacity, if don't ,it will request collaboration to other agents through the contract net protocol which was widen by a conversational method, and ultimately complete missions with other agents by horizontal collaboration. The collaborative manner reduces the difficulty of realization of the requirement of integration and improves the intelligence of the system integration.

3. Dynamic integration and adaptive evolving architecture

3.1 Agent-based integration framework

Referring to FIPA platform specifications, this paper designs an agent-based integration framework in Fig. 1, which supports the dynamic integration in the distributed systems.

When the model is applied, only one computer acts as the control platform and the other computers as non-control platforms. Their coordination accomplishes the system integration. The CRC (Capability Register Center), AMS (Agent Management Service), CMB (Common Message Blackboard), and CA (Control Agent) just exist on the control platform, as special agents to provide services to the other Agents; there is an MTS (Message transfer Service) and an Agent Database on the control platform and each non-control platform.

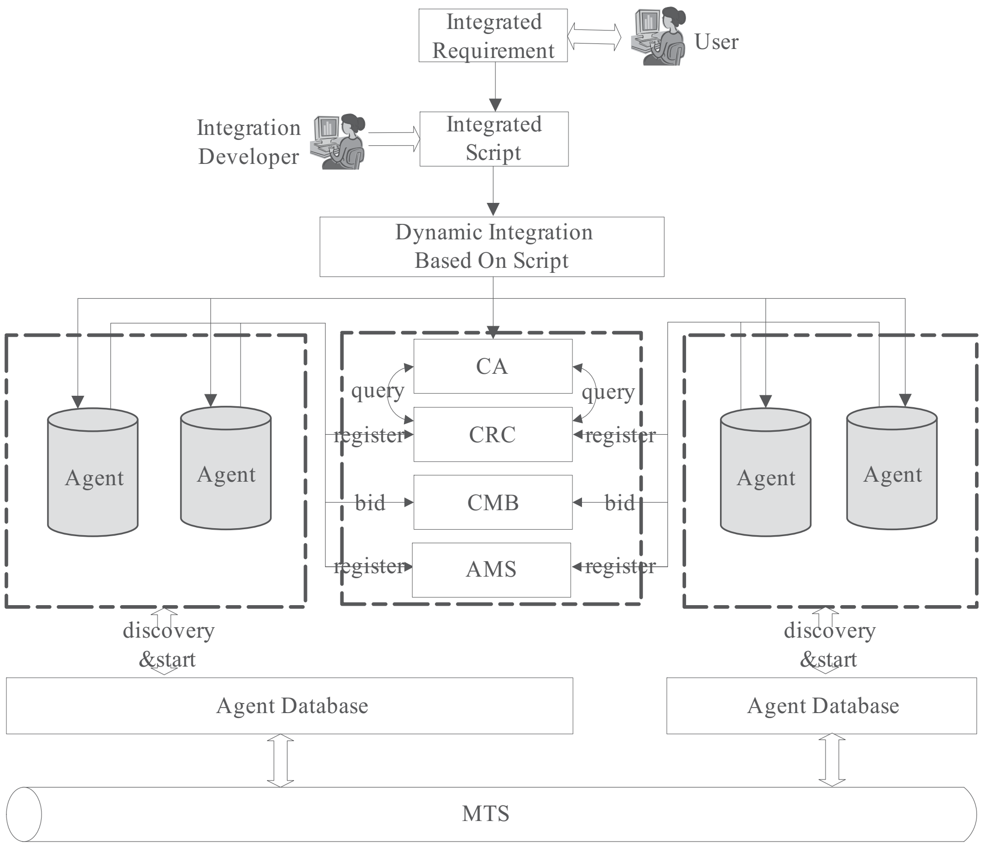


Fig. 2. Dynamic Integration and Adaptive Evolving Architecture

Among those agents, CRC is one of the core modules in the framework and is responsible for the registration of all the agents' capabilities in the platform and the query of these capabilities information; AMS performs real time monitoring on each state and address of the agents (idle, running, waiting and resume), queries the information according to the condition; CMB as the intermediate information service module, provides initial and response service for biding negotiation; CA is not only an intermediary serving but also controls the script interpretation by which integration rules is obtained to achieve System Dynamic Integration; as the relationship between integration units, MTS provides synchronous or asynchronous communication for the interactive and communication among platforms and modules; Agent Database is used to perceive the Agents on the platform, and once one Agent is perceived, it will be added into the local address book that exists in the Agent Database and starts running. Then the agent registers capabilities to CRC and addresses and states to AMS. Also, it can load and upgrade the definition files and capability components in run model to manage static maintenance for Agents.

In the function of this framework, in terms of the wrapper standard, integration developers obtain the definition files and capability components. They are loaded by Agent Database in run mode. Once one Agent is perceived, it will be added in the MAS. After users raise integration needs, integration developers call integration logic by logically abstracting those needs, and then it can be mapped to script. Therefore, CA interprets the script. When running into a simple capability of the script, it will send a message to CRC looking for agents with the capability. After receiving the searched result from CRC, it will obtain the integration rules and send messages to them to request executing the task. Finally, Agents will collaborate with other agents to realize system integration according to the contract net protocol.

It effectively uses the Pro-active, Autonomy and Collaborative of agent, combines the distribution control and concentration control, and achieves the aim of dynamic system integration and expansion in system integration. The problem of load balance is solved by playing a role of the Perceptive, Autonomy and Collaborative of Agent.

In the traditional component-based integration development, the relationship between traditional distributed components is static, which is difficult to adjust the interactive relationship among components dynamically on the fly. The agent-based system integration approach is applied to resolve the dynamic automation. In the approach, the dynamic discovery of agent is the critical part and premise to accomplish system integration. On finding agents that meet needs, it guarantees that agents can be combined and collaborated to accomplish tasks, which realizes system integration. The process illustrates that the dependence, between requirement and agents, is dynamic, but not static.

3.2 Dynamic discovery based on semantic ontology

The dynamic discovery of agent is the automatic location of agent that provides the special capability and accords with limited conditions. For example, assuming that it needs a member of interface display to show the situation, the common method is to redevelop it by requirement or to inquire the existed model base by keywords. But the method can not be used to inquire component and is difficult to meet the rapid construction of application system. Its reason is that facing the mass of components, they are common on the interface specification, but they greatly differ in the semantic meaning of different domains. So it is difficult to locate agent and guarantee the reuse and combination of agent correct and effective.

The core and critical of dynamic discovery of agent is to construct the effective matching algorithm. Mili, et al consider that it is necessary to do the accuracy of initial inquiry by domain knowledge and make more effective inquiry by reasoning based on case and other artificial intelligence technologies.

As the critical port of application system, besides the basic information of general components of function interface and parameter information, agent includes the rich information of model semantics and linguistic context that serve as heuristic information in the agent-matching algorithm. Because of the inefficiency of the pure keyword matching, this paper proposes an accurate matching based on semantic ontology to enhance the accuracy and efficiency of agent-matching algorithm, which is divided into two matching stages, domain location and ability orientation. At the stage of domain location, related domain ontology of an agent can be determined by the algorithm; and then at the stage of ability orientation, it realizes the agent location by the interface information of agent in the domain ontology.

$$DoC_a = \frac{\sum_{i}^{n} t_i \bullet v_i}{\sum_{i}^{n} v_i} \tag{1}$$

In the algorithm of domain location, the basic information of agent may be same as that of others and they are relevant with the concept of different domain-ontology, as a result it is necessary to definite the most relevant domain-ontology. We define the set of domain-ontology as AO={ao1,ao2,…,aon}, of which each domain-ontology aoi={a1,a2,…,an}, and ai is the set of agent involved in aoi. If there exists mapping relationship between the domain of agent in integration requirement and the concept of the domain of aoi $\in$ AO, aoi is the domain-ontology of the agent. By the domain location it can filter a mass of agent irrelevant with integration requirement.

In the algorithm of ability orientation, all the capabilities consist the capability set of Container={ Cf1, Cf2,…, Cfn}, where for each Cfi (1≤i≤|Container|), if there exists inclusion relation between the domain-ontology ao and integration requirement, it holds that all the capability of the agent are put into the set of candidate capability – Candidates. According to the semantic of agent, it makes the accurate matching between agent and integration requirement. In other words, it adjusts whether the capability of agent in Candidates, meets the demand of agent involved in integration requirement by semantic matching.

3.3 Dynamic integration and adaptive evolving

General Magic's Telescript system is developed based on the integration technology of script-interpretation, providing system integration with a new idea. However, the interaction among agent is limited to the local method invocations in Telescript, which adopts the static binding and unable to meet the changes of internet and integrated requirement. Therefore, making full use of the pro-active and collaborative of agent, improving the Contract Net Protocol (CNP) proposed by Reid G. Smith and Randall Davis, it implements a combination of bidding and script switching, to achieve the dynamic.

As is shown in the Fig. 3, when an SA wants to find a contractor through negotiation, it announces the task to be allocated by sending a Bid to CMB. Therefore, the SA is considered as an Initiator. Receiving an announcing from the Initiator, CMB gets the potential contractor (agent) according the announcing. Then CMB inquires CRC to locate agent dynamically. For example, the task in the announcing is type ti, according to the classification of task. If Agentj has an ability of type ti, Agentj is a potential contractor of the Initiator and will be a participant of the negotiation in the bidding. CMB will send calls for proposals to the participants. The main difference from CNP is that CMB sends calls for proposals to participant according the agent ability information it maintains, not to each agent in a broadcast way in CNP.

Meanwhile, CMB monitors the new register and update information. Once an agent is able to be a participant, no matter whether it is a new agent just deployed on to the platform or ability updated, CMB will send the call for propose to the agent if the announcing is still before the declared deadline.

$$DoA_a = \frac{1}{NCFP} \tag{2}$$

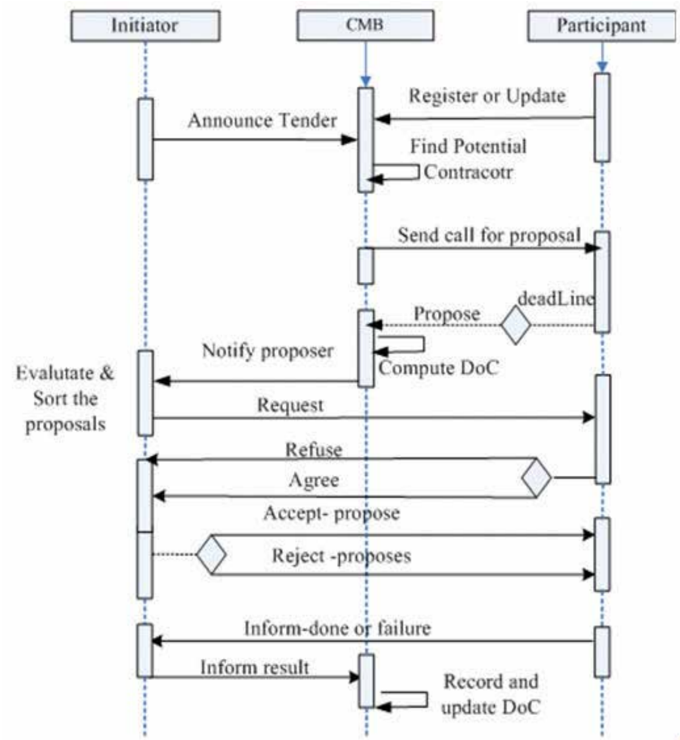


Fig. 3. Dynamic Bidding Process

The participant reads the call for propose sent from CMB and makes the decision whether to bid. The participant may send proposal to bid or refuse to propose by its own willingness, mainly including its ability, status and self-interest. The participant's propose includes the preconditions that the participant is setting out for the task, which are the cost and the number of call for proposals it has received etc. The information of call for proposals is helpful to evaluate the degree of availability of the participant and it is an important factor in awarding decision.

When CMB receives a propose from a participant, it computes the credibility of the participant (Degree of Credibility , Briefly DoC) to complete the task according negotiation history record in CMB. CMB will inform the Initiator with the information of the participant's proposal and its DoC.

Once the deadline passes, the Initiator evaluates and sorts received proposals. The Initiator will sort the received proposals by its cost, DoC and DoA. The higher of DoC and lower of DoA, the earlier the participant may receive the offer. The initiator will send a request to the participant in descending order of the computed value. The process of awarding can be described below.

If a participant receives the offer and agrees, the Initiator will send an accept-proposal message to the participant. The participant will be the contractor and begin to execute the specific task. When the task is done or failed, the contractor will inform the Initiator the result of task execution.

The Initiator will evaluate the result of task execution and inform CMB. CMB will record the negotiation and adjust the DoC of the participant.

The centralized control is the common way to realize the integrated control. It is easy to realize but it will result in the sharp decrease in efficiency of the integrated system. While the autonomy and intelligence of the agents make it interact with other agents initiatively by learning. We can make use of those characters of agents and adopt the control way which combines the distributed and centralized way, in which each agent controls its own behavior. This way not only avoids the sharp decrease in efficiency which is caused by centralized control, but also dynamically changes part or all of agent behaviors at run mode and eventually realizes dynamic system integration.

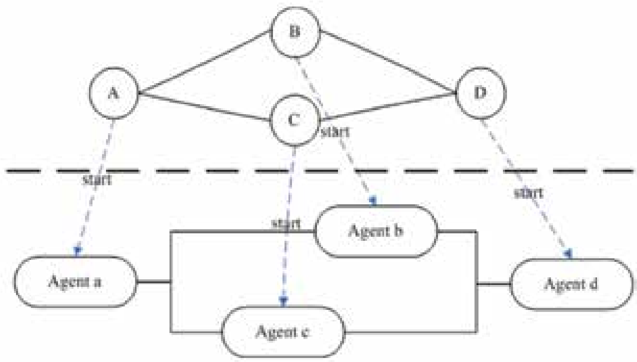


Fig. 4. Mapping Relation on Integrated Rules

Scripted-control integration can be illustrated easily as "once-interpretation and multi-execution". Once-interpretation refers to interpreting the integration scripts once, from which the description of interaction relationship among services is obtained and the integration rule is generated and distributed to the agents which participate in the application system integration. Multi-execution means that the participating agents consider the integration rule as a kind of knowledge and control the cooperation with other agents. As is described in Fig. 4, integration script describes the collaboration relationship among Task A, B, C, and D, aimed at accomplishing integration requirements. When the script is interpreted, the collaboration relationship between tasks is obtained: A→B, A→C, B→D, C→D. By inquiring CRC, the obtained Task A, B, C, D are accomplished respectively by the corresponding service which is provided by Agenta, Agentb, Agentc, and Agentd. The integration rule generated is distributed to the corresponding agent, and the collaboration relations are formed as is illustrated in Fig. 4. When Agents receive their corresponding rules, they will complete integration needs autonomously.

The script depicts the integration logic abstracted from integration requirements, so it has one to one correspondence with integration requirement. Therefore, Integration logic is bound to change when integration requirement needs, which makes integration developers to redefine the script. But in the traditional component-based integration development, it may stop the system via changing the integration logic, which decreases the overall system performance sharply. By switching the script dynamically, the system just replaces integration rules to achieve the flexible integration.

Script Switching, in essence, alters the integration logic smoothly in the process of system integration, under the charge of dispatching Integration Rules and the Agents' internal support. Integration Rules Dispatch is responsible for emptying and replacing the rules, and the Agents' internal support the Script Switching for storing and upgrading the rules.

There exist two situations during dispatching the rules. One is that Service Agenti is not in the pre-script; the other is that Service Agent Agentj is in the pre-script, but its integration logic is not the same. For the former, the rules need clearing; for the latter, the rules must be re-substituted. But both are to replace the integration rules in fact, and their difference lies in the implementation modes. In the Agent internal, storing integration rules provides the foundation for the system running, and updating rules ensures the consistency of integration logic.

When integration requirements change, first of all, the integrated developers abstract the requirements to obtain the integration logic, which is mapped to a script. In the fly, the integration logic is obtained by interpreting the script, and dispatched to the related agent. When receiving the logic, every Agent stores it in temporary zone. Once receiving the "Start" command, Agents substitute the rules of Rule Base with it in a split second. Combined with the flow defined in the Scomp of the related Agent, it forms entire task logic, which controls the system integration.

```
obtain the action and integration rules from message
If(action is "Clear_Rule")
{
assign the state of Agenti to idle and clear the rules base of Agenti.
}
if(action is "Dispatch_Rule")
{
search the capabilities by the agent's name;
if(the searching result is not empty){
clear the rules base of Agenti;
obtain the new rule, and write to the memory.}
}
send the result to CA via MTS
}
```

Algorithm 1. Script Switching Algorithm

4. Agent collaboration mechanism and algorithm

4.1 Partition and cooperating process of tasks

In the process of collaboration between agents, the first problem to be solved is task partition and allocation. The purpose of task allocation is to enable the system complete a task with as less cost as possible in its execution and communication, achieve both local and global objectives while ensure no conflicts occur between agents. At present, some typical works are heuristic allocation algorithm and queuing theory scheduling algorithm. Heuristic allocation algorithm merges the task partition and sub-task allocation as a single linear programming problem. This method is better in handling complicated tasks due to its use of focus coordination mechanisms and distributed collaborative mechanisms. However, when applied in closed-end MAS of small and medium-sized size, the cost of this algorithm is very high. If the completion of the task submitted by user just need one agent, the method of queuing theory is more suitable. The main idea of queuing theory is that each agent has a task queue and the agent performs the task in its task queue base-on the principle of FCFS (first come first served).The main algorithm for task distribution includes the smallest queue algorithm , the shortest waiting time algorithm as well as historical information algorithm. The specific algorithm for a task allocation should be chosen according the needs of the task. In the field of system integration, the complexity of the task is uncertain because of the uncertainty of the task. If just taking the heuristic allocation algorithm, it would be of high risk. And the queuing theory scheduling algorithm could not be just taken for it could complete the complex tasks. To solve the problems above, FSMAS makes uses of the federal structure of itself, and presents solution: MAS allocates task of high complexity to service agents and service agents partition the complicated tasks to simple tasks and assigns the simple tasks to an appropriate function agent, where the algorithm for task partition and task allocation is used. In addition, there will be a process of collaboration between agents after task allocation.

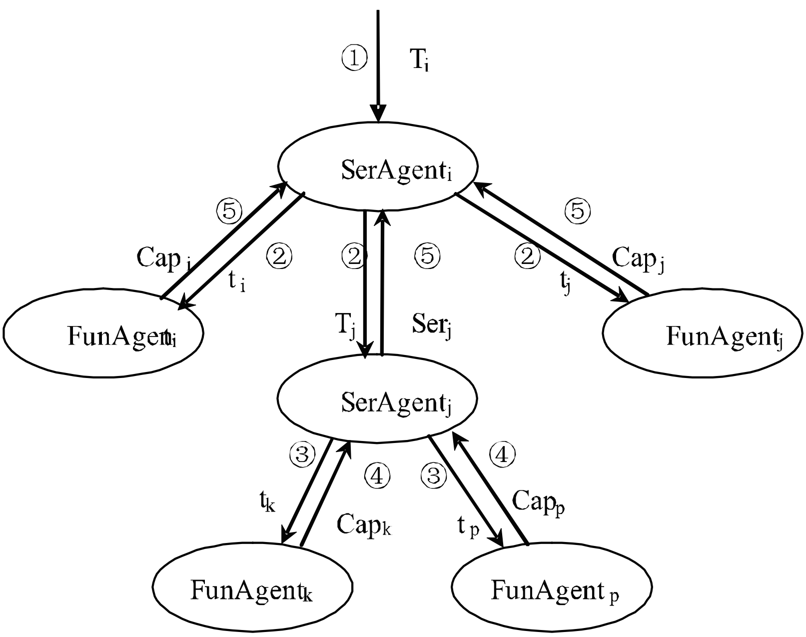


Fig. 5. Partition and Cooperating Process

At first, a whole task is divided by MAS into sub-tasks in a plan. Sub-tasks to be executed are allocated to some federal. The federal will make a partition according the information in the knowledge base of the service agent after getting the task. The sub-tasks will be distributed to and completed by Agent federations. After completing the sub-tasks, the federations collaborate with each other to finish the overall task. Tasks in FSMAS are divided into two major categories, the tasks which can be completed independently by a function Agent are atomic tasks, which are record as t. actually, a function Agent can only complete atomic tasks. Another kind of tasks completed by the collaborations among Agents, are collaborative tasks, record as T. It is defined as T= {Ti, Tk, Tk...ti,tj,tk,…}.

Fig.5. shows the flow chart as an instance of partition and collaboration of an overall task. The main steps are as follows.

① SerAgenti receive a certain task Ti;

② Ti is supposed to partitioned into ti, tj and Tj by SerAgenti, and distributed to FunAgenti, FunAgenti and SerAgentj;

③ Tj continues to be partitioned into tk and tp, which will be distributed to FunAgentk and FunAgentk;
④ FunAgentk and FunAgentk provide Capk and Capp to SerAgentj , SerAgentj organizes the capabilities to provide Serj;
⑤ SerAgenti organizes Capi , Capj and Serj to provide Seri. This organization has been able to complete the overall task Ti.

The specific algorithm for task partition is as follows.

```
Queue PartitionTask(Ti){
Queue taskqueue;
Ti →{task1, task2,….taskk}
taskqueue.pushback({task1, task2,….});
for(int i=1; i!=end; i++) {
If(∃taski | taski ∈ T){
Queue subtaskqueue=PartitionTask(taski);
        taskqueue.pushback(subtaskqueue);
}
}
return taskqueue;
```

Algorithm 2. Task Partition

When a service Agent gets a collaborative task, it partitions the task according to the definition file and would get a task queue. If there still exists some collaborative tasks and no member service Agents could complete those tasks, the original service Agent would continues to partition those tasks until all of the tasks in the task queue could be completed by the members in its federation. At last, the original service Agent would return the task queue.

MAS take the simple algorithm for task allocation. The tasks which are allocated to service agents are determined by integration control-logic defined by user. One simple implementation is to assign the task to the first service agent found by query. For function agents, there are two ways for task assignment, one of which is service agents assign task to specific function agents by its knowledge and the other is task is assigned to specific function agents through biding with contract net protocol. In the filed of system integration, a function agent usually just serve as a function supply unit. There usually would not be many agents for an integration function unit. Therefore, the queuing theory allocation algorithm can not significantly improve system efficiency while the calculation of information would cause unnecessary system costs. The adoption of simple task allocation algorithm in FSMAS avoids the calculation of the length of task queue and waiting time and can meet the task allocation requirement in system integration filed.

4.2 Agent collaborative algorithm

The typical work of collaboration in MAS is Contract Net Protocol. The main procedure includes task announcement, sending call for proposal, bidding, evaluation and awarding and contract execution. Distributed Sensing System of MIT and expert union system UNIONS of Chinese Academy of Science Mathematics were both typical system based-on contract net protocol. However, contract net protocol also has its problems such as frequent communication, waste of system resources and limited scope of application. Blackboard

Collaboration is a more widely used collaboration method. The main idea is the Blackboard provides sharing solve space when a number of agents collaborate to solve a problem. All the agents involved can "see" the information of the blackboard and record the intermediate result to it until the problem is resolved. In this way, agents are independent of each other and the communication load is reduced. What agents need to do is to response the blackboard and to use the same communication language.

As the problems of CNP mentioned in the introduction, a new collaborative algorithm "acquaintance first base on CNP" is proposed in this paper. Its main idea is: service Agents choose the Agents in their acquaintance base to complete tasks first, if the Agents in acquaintance base can not complete the task, the service Agent would initiate a bid to get the Agents needed.

As Algorithm 3 shows, when a service Agent receives a task, it partitions the task by the partition algorithm has been mentioned. First, the Agent gets the Agents' information which would be organized according to the definition files. Second, the service Agent would search the Agents' information in CRC, the information would show which Agents are the acquaintances of the Service Agent and which are not. Third, if all the Agents involved are acquaintances, the service Agent would organize them and return a service; if not, then initiates a bid. When the service Agent gets all the Agents, it would organize them and return a service.

```
SerAgentExeTask(Ti){
taskqueue=PartitionTask(Ti);
taskqueue×CRC→Agentqueue;
    if((∀ agent | agent ∈ Agentqueue)→(agent ∈ Ab))
      Ser={Ser1,…Serm, …Cap1, …Capn}
|Seri ∈ Serbi, Capi ∈ Capbi;
    else {
            Bidagent | agent ∈ Agentqueue && agent ∉ Ab;
            Bidagent×BidMod→bid;
            Initiate(bid);
            Wait();
            Bidagents=Accept(bid);
Ser={Ser1,…Serm,…Cap1,…Capn,Bidagent1,…Bidagentk}
|Seri ∈ Serbi, Capi ∈ Capbi, Bidagenti ∈ Bidagents;
}
    return Ser;
```

Algorithm 3. Service Agent Execute A Task

Algorithm 4 shows how CMB process a bid when receives it and how the Agents wins the bid. When a service Agent initiates a bid, it would send a message about the bid to CMB, CMB would analyze the message and get some information about the bid after it received the message, then it would query CRC according to the message and get an AgentSet, next, it would send Msg which records the bid's information to all the Agents in AgentSet and wait for reply. CMB would send the information WinMsg of first replying Agent to the initiator Agent. And to every one in AgentSet, when they got the message, they would check their stat first and send WinMsg if the stat is idle. Then it would be waiting for invoking.

```
Bid(bid)
{
bid×CRC→AgentSet;
for(Agenti=First(AgentSet); Agenti!=Last(AgentSet); i++)
{
    SendMsg(Agenti, Msg);
}
Wait();
Send(WinMsg) | WinMsg is sent to initiator;
}
WinBid(bid)
{
    if(state==idle)
SendMsg(WinMsg);
```

Algorithm 4. Bidding Algorithm

Using “acquaintance first based on CNP” collaborative algorithm has the advantages as follows: 1. choosing acquaintance to collaborate could greatly enhance the efficiency of the collaboration; 2. the bidding mechanism could ensure the completion of tasks and the success rate of coordination; 3. adding acquaintance automatically after bidding, which prepares for the next collaboration.

4.3 Collaboration exception handling

Several exceptions always appear in the collaboration process. If those exceptions could not be handled appropriately, not only the robustness of system would be reduced significantly, but also the system would be attacked easily and even cause the system breakdown. Collaboration exceptions means any of a deviation from a normal execution process (such as resources, conflicts, overtime, etc), or that means a deviation from the best design. Collaboration exception could be divided into task partition exception, task allocation exception and collaboration executing exception. If the exceptions belong to the first two categories, the handling method is allocating the tasks again. This paper focused on the approaches which handle the exceptions appear in the task collaboration. These exceptions are divided into internal and external exceptions. Internal exceptions could be handled by the Agents who generate them. And external exceptions are those ones could not be handled by the owner without external assistance.

For internal exceptions, the handling mechanism inside of the Agent could handle them appropriately. And to handle external exceptions, some collaboration between Agents needed. The strategy is not only to recover the normal collaboration between the Agents, but also have some measures to be taken to make the abnormal Agents produce smaller impact in late collaboration. In FSMAS, service Agents manages communication between federations, function Agents could not communicate with others. So within a federation, service Agent can be used as a trigger to detect the collaboration exceptions, and also could handle the exceptions by the handling mechanism of themselves. And to between federations, the exceptions would be handled by the Agents who collaborate with the abnormal Agent.

The process of handling exceptions is as Algorithm 5 (suppose that agent1 produces an exception).

```
if(agent1 has e )
{        // e  is an exception
if (agent1 can handle e in_model)
 {
Handle(e);
 if(agent1 is OK)
 continue;
      Else
           Hand e to funAgent1;   //funAgent1 manages
// agent1
}
else
      Hand e to funAgenti;  // funAgenti collaborates
// with funAgent1
```

Algorithm 5. Exception Handling Algorithm

5. Agent-based intelligent collaboration environment

5.1 Agent-based system integration platform and tools

Based on above, the agent-based system integration architecture is designed, which supports the dynamic integration and adaptive evolution in the distributed system. Based on the architecture, Agent-based System Integration Development Environment (ASIDE) is implemented which involves Agent Wrapper, Agent Warehouse, Script Design Tool and Run-time Support Platform.

Agent Wrapper is responsible for packing the integration units (legacy systems or newly developed systems) into Function Agents and Service Agents which have the unified interfaces and possess some specific capabilities (functions). Script Designer is the tool for designing the script in DCISL and it has Text Mode and Graphic Mode. Moreover, the semantics, syntax and integration logic checking are also involved in Script Designer to ensure the correctness, reliability and integrity of the integration logic. With the data integration development tool, the data conversion regulation files are generated in terms of the data conversion relations (as shown in Fig.12 and Fig.13), which are the plug-ins. After being deployed in the platform, the data conversion regulation files are extracted the related information by the engine to perform the data conversion among the agents.

With the run-time management tools, the intelligent and dynamic integration process is obtained in the run-time support platform. AMS, Integration Control Tool and CMB are the three critical tools among the run-time management tools to implement this process. AMS is in charge of managing the overall agents which exists in MCP and NCP. When some new agents are loaded into platform, AMS will perceive them immediately and allocate resource to them, and then the new agents will join the running integration process dynamically and initiatively if needed. Additionally, the related status information about certain agent can be looked up and displayed in AMS real-timely. Integration Control Tool is used to load and interpret the script-based integration control logic to control the system integration action. The interpreted integration control logic is distributed to the related agents who store the logic as their own regulation and act in terms of the regulation to accomplish the integration task. When the integration requirement changes, the integration control script is altered accordingly and then loaded to perform the new integration without interrupting the running system. Thus it achieves a smooth and flexible integration logic switch. Therefore, it is the strategy that can deal with the integration requirement transition flexibly and dynamically, while the traditional component-based system integration method can not.

Agent Warehouse takes charge of discovering and managing all the agents in the Run-time Support Platform. Script Design Tool is used to design the integration scripts in terms of integration requirements. And Run-time Support Platform provides service for the execution of system integration, which involves Platform Monitor Tool, Platform Configuration &&Management Tool, AMS, CRC, CMB and so on. Platform Monitor Tool timely monitors and displays the communication and collaboration amongst all the Agents which exist in the distributed system, and Platform Configuration&&Management Tool configures and manages the related platform information, such as IP and Ports of the computer. AMS is in charge of querying, managing and displaying the overall agents' information, CRC maintains a list of the agents' capabilities and can dynamically perceive the newly-loaded agents so as to record their capabilities; CMB provides service for the agent bidding mechanism based on Contract Net Protocol.

In the system integration process, the collaboration among agents adopts the federated acquaintance-first strategy. Once the capabilities of the agent can not accomplish the distributed integration task, it will try to find the appropriate agent through inviting public bidding based on Contract Net Protocol. The other agents who have the required capability will bid initiatively and provide service if chosen. It achieves the dynamic and intelligent collaboration among the agents. The biding process is monitored and the related agent information is displayed in CMB.

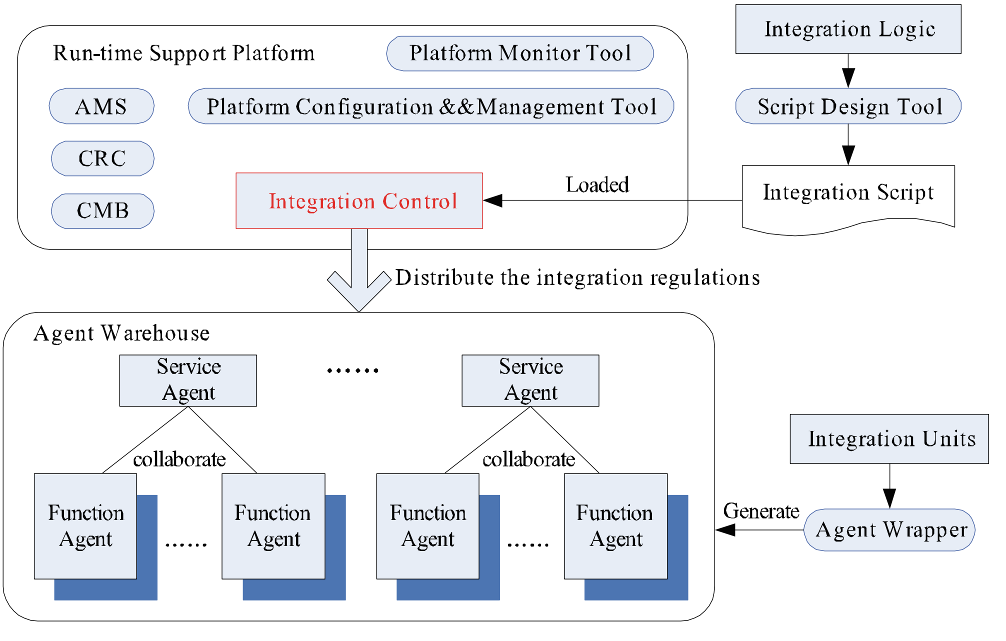


Fig. 6. Architecture of ASIDE

In ASIDE, the agents in Agent Warehouse can be discovered and perceived dynamically. With the help of CMB, the agents can gain or provide the required service in terms of bidding mechanism. Furthermore, when the integration requirement changes, the new integration logic is expressed as script, and without interrupting the running integration activities, the new script can be loaded into ASIDE to carry out the new integration. The dynamic switch of scripts leads to the integration transition, and it adapts to the frequently changing requirements flexibly and eventually achieves dynamic system integration.

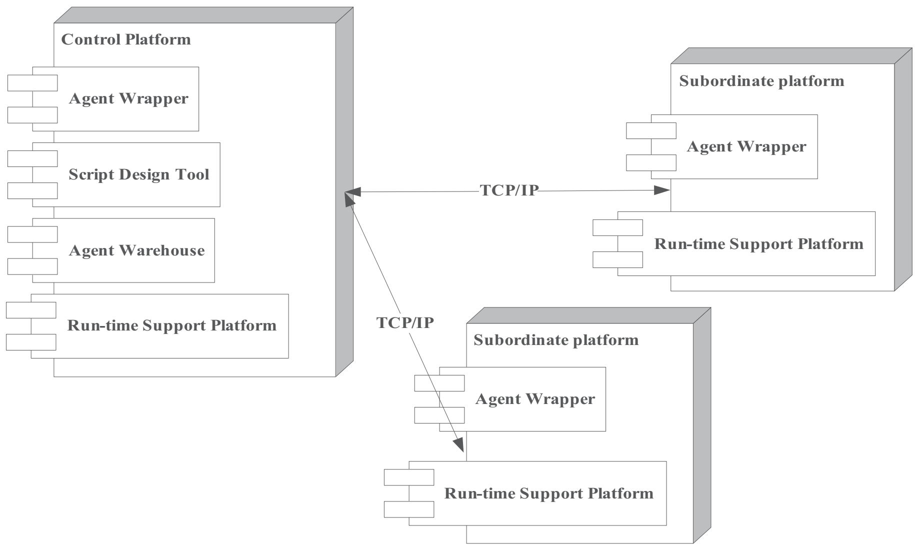


Fig. 7. Deployment Diagram of ASIDE

In addition, to support the distributed system integration, Control Platform (CP) and Subordinate Platform (SP) are developed, which are deployed in distributed computers. One Control Platform and several Subordinate Platforms are needed in the distributed integration and different tools of ASIDE are installed in them to support the integration, the deployment diagram is shown in Fig 7. ASIDE are deployed in Control Platform while only Agent Wrapper and Run-time Support Platform are deployed in Subordinate Platform. During the integration process, the agents are loaded and deployed in Control Platform and Subordinate Platforms, and the agents' information (especially the information of their capabilities) is immediately detected and recorded by CRC in Control Platform. After the integration script is loaded in Control Platform, the related agents are dynamically controlled and organized to accomplish the integration.

5.2 Agent-based system integration development process

The integration development process is divided into three phases as shown in Fig.8. Integration unit division is accomplished during requirement analysis phase, in integration design phase, agent design and integration logic design is completed. During run-time integration phase, the agents are deployed in Agent Warehouse and managed by Run-time Support Platform. After the script is loaded, the interpretation of script will lead to the occurrence that the integration regulations is distributed to the related agents, when the agents receives the regulation, they will immediately act in terms of the regulation to carry out the integration. In this phase, once the required capabilities by an agent are lost with the owner or lacked, the bidding mechanism will work to find the agent who is idle and owns the same capability, Meanwhile, the agents who satisfy the bidding demand will initiatively make a bid. Eventually the agent will choose the best one from the bidders. The bidding mechanism copes with the environment changes (the occurrence and disappearance of agents in the platform) dynamically and flexibly, which achieves the adaptive integration. Additionally, when the integration requirement changes, the interpretation of the new script controls the collaboration of the related agents to perform the new integration, this process does not need to interrupt the running system, and thus it achieves the dynamic and flexible adaptation to the requirement changes, which is the dynamic integration presented in this paper.

The system provides two types of interfaces, one faces to the general users, and the other faces to the integrated development users. The Integrated development users though the analysis of specific areas extracted the business processes of the field, use script language to define integrated rule according to the business processes, and add the rules into integration rules storage of this field, simultaneously abstracts the relatively complete functional module in the business processes to use as agent. For general users, firstly should input the integration needs in the man-machine interface, then the integrated development users invoke the integrated rules of the related field, map the integrating demand into integrating script .The control integrated agent controls service agent collaboration though explanting facility of integrated script, and achieves the goal of the system dynamic integration.

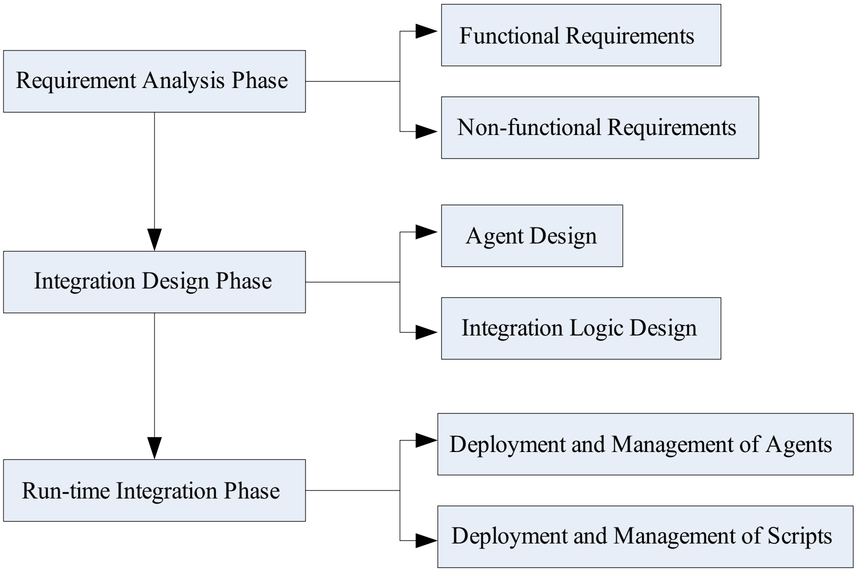


Fig. 8. Integration Development Process

After users input their requirement in human-computer interface, the integrated development staffs will call the integrated rules from corresponding field and map to integrated script based on the users' demands. The centralized control agent as a special agent calls the script interpreter to explain the integrated script. When running into a simple capability of the script, the centralized control agent will send a message to CRC looking for agents with the capability. After receiving the searching result from CRC, the centralized control agent will send messages to them to request executing the task. If the agent holding the capability cannot complete the task by itself, it will request collaboration with other agents through the contract net protocol based on agent active perception. After the agent completes the task, it will inform the centralized control agent of the result for continuous execution of the next task. In this framework, the centralized control agent, CRC, AMS, PIB as well as other agents communicate with each other through information mechanism, which can support synchronous and asynchronous communication, and can bring about convenience to build distributed systems.

5.2.1 Requirement analysis phase

Requirements analysis is critical to the success of a development project，Requirements are a description of how a system should behave or a description of system properties or attributes. The functional requirement of system integration is integrating the sub-systems to construct a new system which accomplishes a certain function and the non-functional requirement is that the integration should be dynamic and flexible so that it can cope with the changing integration requirement and environment.

5.2.2 Integration design phase

Two types of Agents are designed, which are Function Agent and Service Agent. Function Agent is a concrete entity which possesses some basic capabilities while Service Agent is an abstract entity which organizes the Function Agents together in terms of certain collaboration logic to provide a high level service.

After analyzing the system integration requirement, the basic function modules (if not exist) are designed and implemented to provide a specific function. Afterwards, With Agent Wrapper, single function module is packed into a function agent (as shown in Fig.9 and Fig.10) and then several function agents are packed into a service agent by being defined as the acquaintances of the service agent. In addition, the service flow which depicts the collaboration relation of the organized function agents is also defined in service agents. Moreover, the service flow checking is also accomplished in the wrapping process to ensure the reliability and integrity.

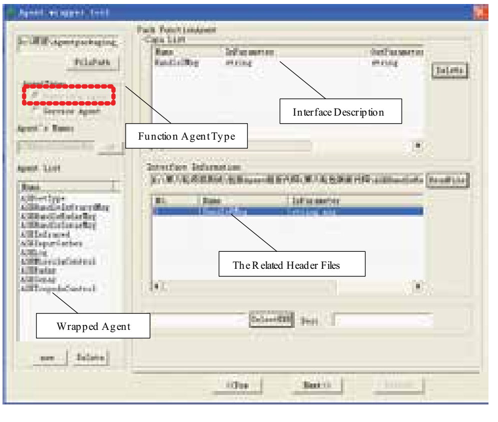


Fig. 9. Wrapping of Function Agent

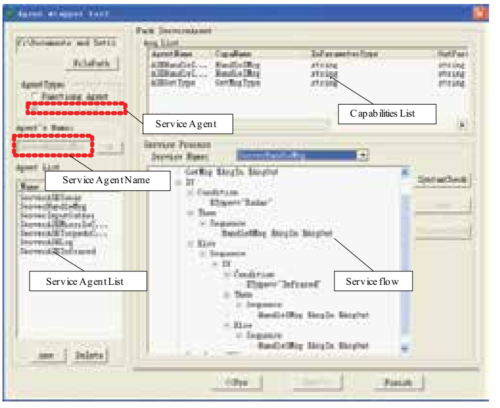


Fig. 10. Wrapping of Service Agent

DCISL (Dynamic Control Integration Script Language) is introduced in ASIDE, which depicts the integration control logic which will control the interaction and collaboration of the agents to accomplish the integration. Script Designer is the tool for designing the script in DCISL and it has Text Mode (as shown in Fig.11 and Fig.12) and Graphic Mode. Moreover, the semantics, syntax and integration logic checking are also involved in Script Designer to ensure the correctness, reliability and integrity of the integration logic.

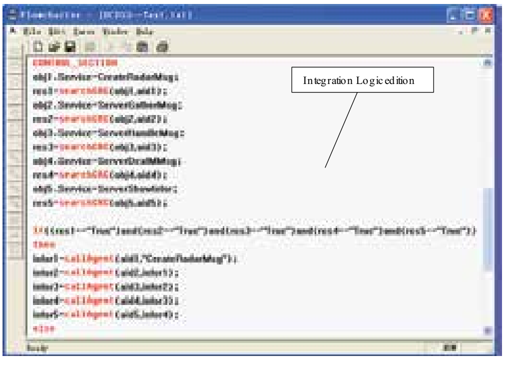


Fig. 11. Editing Script in Text Mode

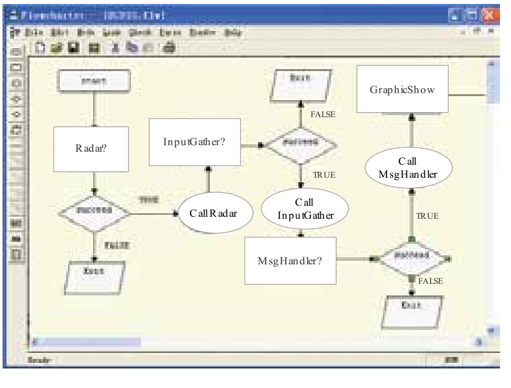


Fig. 12. Editing Script in Graphic Mode

Features like naming conflict, different format, and dissimilar structure of the data in different agents might cause inconvenience during the system integration process. To deal with these issues, Data Integration strategy is proposed in ASIDE, which involves data integration engine and plug-in, the extended functions are implemented by plug-ins which is produced by the data integration development tool.

In the system design phase, the data conversion relations are gained by analyzing the interface information of agents. With the data integration development tool, the data conversion regulation files are generated in terms of the data conversion relations (as shown in Fig.13), which are the plug-ins. After being deployed in the platform, the data conversion regulation files are extracted the related information by the engine to perform the data conversion among the agents.

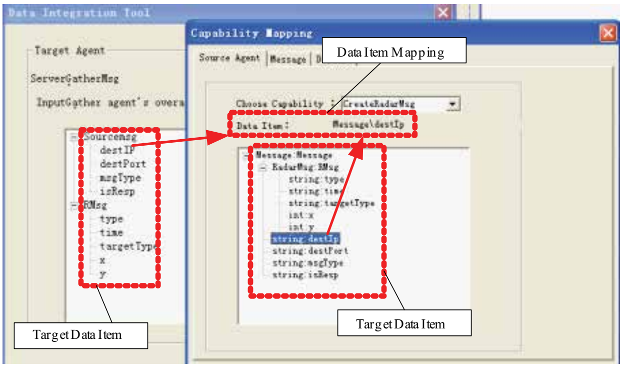


Fig. 13. Data Integration Development Tool

5.2.3 Run-time integration phase

With the tools in Run-time Support Platform, the intelligent and dynamic integration process is obtained. AMS (as shown in Fig.14), Integration Control and CMB (as shown in Fig.15) are the three critical tools among these tools to implement this process.

AMS is in charge of managing the overall agents which exists in CP and SP. When some new agents are loaded into Run-time Support Platform, AMS will perceive them immediately and allocate resource to them. Additionally, the related information of the agents can be queried real-timely in AMS.

Integration Control is used to load and interpret the script-based integration control logic to control the system integration. The interpreted integration control logic is distributed to the related agents and they store the logic as their own regulation and act according to the regulation to accomplish the integration task. When the integration requirement changes, the integration script is altered accordingly and then the new integration regulation is distributed to the agents, it replaces the old one and conducts the agents to accomplish the new integration.

In the integration process, the federated acquaintance-first strategy is applied in the collaboration among agents. Service Agent firstly obtains service from its acquaintance (function agent), if the required capability by the Service Agent is not possessed by its acquaintances, it will try to find the appropriate agent through inviting public bidding. The other function agents who have this capability will bid initiatively and provide service if chosen. The biding process is monitored and displayed real-timely in CMB.

Additionally, Monitor Tool monitors and displays the communication and interaction amongst all the Agents which exist in CP and SP. CRC deployed on CP maintains a list of overall Agents' capabilities. When the new agents are loaded, the related capabilities information is registered in CRC. The query function is also provided to inquire the ability information of certain agent which is loaded in certain platform.

Fig. 14. AMS

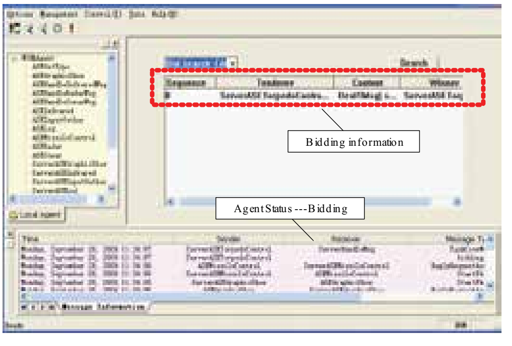


Fig. 15. CMB

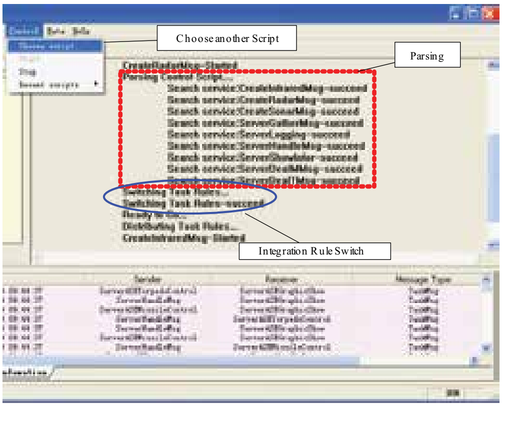


Fig. 16. Integration Logics Switch

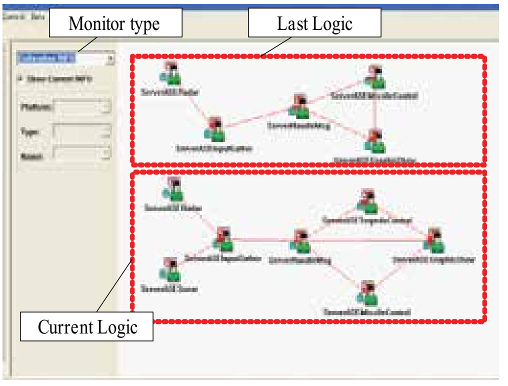


Fig. 17. Monitor the Scripts Switch

By choosing the first integration requirement script and loading it into the Control Platform, the script is interpreted and distributed to the agents who collaborate with each other in terms of script, as shown in Fig.16. With the system keeping running, the second script can be loaded into it and the new integration will be performed, the related switch information is shown in Fig.16.

When switch to the next script, the absence of ASETorpedoControl agent results in the suspending of the integration process. Therefore, the service agent ServerASETorpedoControl will bid for the capability of DealMsg, the bidding information is shown in Fig.15. After loading the ASETorpedoControl function agent, the required capability is provided by ASETorpedoControl agent and the process is activated to continue the integration. The monitor center displays the collaboration relation before and after switching in Fig.17.

6. Conclusion and future work

With the rapid development of computer software and hardware technology, the system scale is more gigantic and the function is more intricate, how to reuse and integrate the existing system efficiently to construct large-scale system is the emphasis of many researches. Though the traditional component-based system integration strategy brings some solutions towards this issue, it lacks flexibility and dynamic especially when the integration requirement keeps changing. To address this problem, the Agent-based software integration approach is put forward and implemented in this paper which applies the agent technology into the system integration to intelligent the integration process. With the agent wrapper, the function modules are wrapped into agents who are self-determined and self-adaptive. With the script-based control integration logic, the integration requirement switch is realized flexibly. With the bidding mechanism based on Contract Net Protocol, the required capabilities are provided by the agents who possess the ones so as to continue the integration process.

The contract net protocol is a simple and dynamic scheme but is communication intensive due to the broadcast of task announcements. Its performance degrades drastically when the number of communicating agents and the number of tasks announced increases. This limits its usability in a large-scale multi-agent system. In this thesis, the Contract Net Model based on Agent Initiative Perception is proposed. The functions of the perception coefficient and the degree of credibility are introduced to study the changes of environment and agent itself. The parameter for the perception coefficient represents the current overload of an agent, and the parameter for the degree of credibility shows the completion history of tasks having been finished by the agent. The contract tender decides to bid or not in terms of the bidding strategy, and the contract manager consigns tasks to agents based on the tendering strategy, that improve the system efficiency. However, when the number of tasks is small, the set of the intersection of these agents' capabilities is nearly empty. Each agent in the system will perceive tenders that increases the communication overhead of the system. As part of the future work, we intend to go further improving CNMAIP and using CNMAIP in the Agent-based System Integration Tool. At the same time, we will also further study how to motivate an agent when the tasks are finished by the agent successfully and how to punish an agent when the tasks are not complete on time.

7. Acknowledgement

This work is supported by the Defense Pre-Research Project of the 'Eleventh Five-Year-Plan' of China under contract number 513060401 and the Fund of C4ISR National Defense Science and Technology Key LAB under contract number 9140C8301011001; In addition, it is supported by the 'Fundamental Research Funds for the Central Universities'.

8. References

Rabelo, R. Camarinha-Matos, L., Afsarmanesh, H. (1998).Multi-agent Perspectives to Agile Scheduling. *Intelligent Systems for Manufacturing*, pp. 51-66. Kluwer, Netherlands.

Shen, W., Norrie, D. (1998). An agent-based approach for dynamic manufacturing scheduling. In *Proc. of the 3rd International conference on the practical applications of agents and multi-agent systems*, pp. 533-548. London, UK.

R.G. Smith, R. Davis. (1980) The contract net protocol: High level communication and control in a distributed problem solver. *IEEE Tran. Computers*, 1980, C-29(12): 1104—1113.

T. Sandholm. (1993) . An Implementation of the Contract Net Protocol Based on Marginal Calculations. In *Eleventh National Conference on Artificial Intelligence*, pages 256-262, 1993.

T. Sandholm, V. R. Lesser. (1996).Advantages of a Leveled Commitment Contracting Protocol. *In AAAI*, 126-133. 1996.

M. D'Inverno, M. luck. (1996). Formalizing the Contract Net as a Goal-directed System. In *Proc. 7th MAAMAW*, Eindhoven, Netherlands, LNAI Series, 1038, pages 72-85,.

E. Werner. (1989). Co-operating Agents: A unified theory of communication and social structure, In *Distributed Artificial Intelligence*, II, M. Huhns and L. Gasser, Eds. Pages 3-36. Kaufman and Pitman, London,.

M. Fisher, M. Wooldridge. (1994).Specifying and Executing Protocols for Cooperative Action. In S. M. Deen, editor, CKBS-94 --- *Proceedings of the Second Internateional Working Conference on Cooperating Knowledge-Based Systems*. Springer- Verlag,.

R. Brooks. (1991). Intelligence without Representation, *Artificial Intelligence*, 47(1-3), 139-159.

Rao A S, Georgeff M P. (1991). Modeling rational agents within a BDI-architecture. *Proceedings of the 2nd International Conference on Principles of Knowledge.* Representation and Reasoning, 473-484.

ZHANG Shuang-Min and SHI Chun-Yi. (2002). Modeling Agents Combined with the State of Environment. *Computer Research and Development*. Dec 2002, 39(12), 1587-1591.

WANG Yi-Chuan, SHI Chun-Yi. (2003). A Concurrent BDI-Agent Model. *Jounal of Software*. 2003, 14(3), 422-428.

DONG Ming-Kai, Zhang Hai-jun, SHI Zhong-Zhi. (2004). An Agent Model Based on Dynamic Description Logic. *Journal of Computer Research and Development*. May 2004, 41(5), 780-786.

Wooldridge M, Lomuscio A. (2000). Multi-Agent VSK logic. In: Ojeda-Aciego M, deGuzman IP, Brewka G, Pereira LM, eds. *Logics in Artificial Intelligence*: European Workshop, JELIA 2000. LNAI 1919, New York: Springer-Verlag, 300-312.

HS Nwana, DT Ndumu, LC Lee, JC Collis. (1999).ZEUS: a toolkit and approach for building distributed multi-agent systems. *Applied Artificial Intelligence*,.

Jennings N R, Sycara K, Wooldridge M. (1998). A Roadmap of Agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1998:1275-306.

Wekman K.(1991).Using negotiation and coordination systems. *Proceeding of 12th International Joint Conference on Artificial Intelligenc*e, 187-198.

R.G. Smith, R. Davis.(1980).The contract net protocol: High level communication and control in a distributed problem solver[J]. *IEEE Tran. Computers*, 29(12):1104-1113.

Haishun Yun, Qingshan Li ,Dan Jiang, He Liu, Shaojie Mao, Yuping Li. (2009). A Contract Net Protocol Based on Information Intermediary Service in Multi-agent System. In *proceedings of 2009 International Conference on Artificial Intelligence and Computational Intelligence*, IEEE-CS Press. pp.

Yingqiang Wang, Qingshan Li, Chenguang Zhao, Hua Xing. (2008). A Multi-Agent System Frame Model for Dynamic Integration. In *Proceeding of the 9th International Conference for Young Computer Scientists*, ICYCS 2008. IEEE CS Press. Zhangjiajie, China. p.1163-1168.

Dan Jiang, Qingshan Li, Haishun Yun. (2010) .Collaboration Algorithm of FSMAS. In *proceedings of the International Conference on Swarm Intelligence* (ICSI 2010). Springer LNCS, Vol. 6145, June 12-15, Beijing, China. pp.390-296.

Qingshan Li, Weihua Sun, Lili Guo. (2010). An Agent-based Dynamic Integration and Adaptive Evolving Model in Software Development. *Journal of Communications of SIWN*. June-July 2010. Vol.8-9.

Meisheng Wang, Qingshan Li, Chenguang Zhao. (2008). A contract net model based on agent active perception. In *Proceeding of the 32nd Annual IEEE International Computer Software and Applications Conference*, COMPSAC 2008, Turku, Finland. pp.511-516.

Chenguang Zhao, Qingshan Li, Meisheng Wang, Yingqiang Wang, Yuping Li. (2008). An agent based wrapper mechanism used in system integration. In *Proceeding of the 2008 International Conference on e-Business Engineering*. ICEBE 2008. IEEE CS Press. Xi’an, China. p.637-640.

Weihua Sun, Qingshan Li, He Liu, Lei Wang, Shaojie Mao, Yuping Li. A Scripted-Control Integration Strategy in Multi-agent System. In *proceedings of the 2009 International Conference on Artificial Intelligence and Computational Intelligence* (AICI'09), IEEE-CS Press. Nov. 2009, Shanghai, China. pp. 263-267.

Shanshan Hao, Qingshan Li, Meisheng Wang, Hua Xing, Chenguang Zhao. A Systems Integration Oriented Multi-Agent Hierarchy Cooperation Model. *International*

Conference Computer Science and Software Engineering, CSSE 2008. IEEE CS Press. Wuhan, China, 2008. 452-455

Wooldridge M, Jennings N R. Intelligent Agent: Theory and Practice. *Knowledge Engineering*, 1995, 10(2).

Ed Ort. Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools [M]. *SunMicrosystems*. 2005.

Cheyer A, Martin D. The Open Agent Architecture, *Journal of Autonomous Agents and Multi-Agent Systems*. March 2001. 143~148.

Chu WJ and Tolone, et al. Integrating Manufacturing Softwares for Intelligent Planning Execution: a CIIMPLEX Perspective.

Shen W, Norrie D H. A Hybrid Agent-Oriented Infrastructure for Modeling Manufacturing Enterprises. In: *Proc. of KAW'98*, Banff, Canada, 1998(Agents-5: 1~19).

14

The Fusion of Fuzzy Temporal Plans: Managing Uncertainty and Time in Decentralized Command and Control Systems

Mohamad K. Allouche and Jean Berger
Defence R&D Canada, Valcartier, 2459 Blvd. Pie XI North, Val Bélair, G3J 1X5 Canada

1. Introduction

Managing uncertainty in planning and plan execution activities is a key issue. This issue is even more critical in a network enabled environment where several tasks are distributed over the environment and can be carried out by different partners under different temporal constraints. In previous work (Allouche & Boukhtouta, 2009) a framework for distributed temporal plan fusion and monitoring has been proposed. A set of agents are tasked to coordinate the execution of different plans with different temporal constraints. Those plans are fused into one single plan, called coordinated plan. A coordinated plan can be executed and monitored by several agents while respecting the original temporal constraints of each agent's plan. The temporal constrains are set on tasks duration or/and between tasks. Each temporal constraint specifies the minimum and maximum authorized temporal distance between two events that typically represents the beginning or the end of execution of a task. In our opinion, the choice of such temporal constraints is not realistic in a distributed environment where different players must execute a common mission with limited and incomplete knowledge of their environment. The violation of a temporal constraint even with one unit of time will cause the plan execution to fail. For example, if a temporal constraint specifies that the duration of a task should last between 10 and 20 minutes, the fact that the task duration is 21 minutes is sufficient for the failure of any plan that contains this task. The fact that this situation is very likely to happen and that one minute late might be acceptable for a decision-maker, a new framework with degrading solutions for the problem is needed. In this work, we use fuzzy temporal constrains to maintain the execution of a plan with a degradation of its performance. In this context, the decision-maker decides whether the current execution is acceptable or not.
The following sections are organized as follows: Section 2 presents some related work. Section 3 presents a general framework for the fusion of fuzzy temporal plans. This framework is then applied to a Combat Search and Rescue (CSAR) mission in Section 4. Finally, Section 5 presents some conclusions and future work.

2. Related work

Classical requirements in supporting temporal reasoning for various application domains is the ability to handle and process quantitative information characterizing event duration, and its intrinsic dependencies in handling multiple event or activity constraints. Temporal networks offered a suitable formalism to capture and handle constraint information and relationships in which variables denote event times, typically defined by start and end time points over a given timeline, and constraints reflect possible temporal relations between them. A well-known approach to properly represent constraint-based quantitative temporal networks lied within the general realm of Temporal Constraint Satisfaction Problems (TCSPs) (Dechter et al., 1991). TCSP is a technology to represent and support queries about events and its existing temporal relations. It mainly aims at determining constraint consistency, and answer scenario-based what-if constraint satisfaction queries. Early work mainly conducted in the 90's on TCSPs has been devoted on problem classification tractability, exact and polynomial-time approximation search algorithms (Schwalb, 1998).

Driven by problem requirements, our work primarily relates to *Simple Temporal Problems* (STPs) (Dechter et al., 1991) a subdomain of TCSPs. In the general setting, dependencies between temporal variables are captured in a constraint directed graph in which nodes represent an event time variables and arcs connecting nodes reflect binary constraints expressed as single time intervals exhibiting event duration. The formalism is used for basic temporal problem expressivity and support reasoning about temporal constraints. It provides an inference mechanism to verify properties, check consistency and handle queries. A solution to a STP problem prescribes values to event/activity time variables in order to satisfy all temporal constraints defined in the network. As STP focuses on non-disjunctive temporal constraint (single time interval), TCSP deals with general disjunctive temporal constraints (multiple intervals) (Schwalb & Dechter, 1997; Venable, 2005). Despite this apparent weakness, STP proves to be quite valuable in many practical application domains trading-off problem modeling complexity and tractability (polynomial time solution computation). Recent research on simple temporal problems has been increasingly directed to the development of approaches with augmented semantics and expressivity, and enhanced capability to efficiently handle uncertainty and preferences (Rossi, et al., 2006). Proposed frameworks and extensions include Simple Temporal network (crisp constraint, with no uncertainty), Simple Temporal Problems with Preferences, Simple Temporal Problems with Uncertainty which is closely related to our proposed approach and, Simple Temporal Problems with Preferences and Uncertainty. A recent survey may be found in (Rossi, et al., 2006).

Simple Temporal Problems (STPs) (Dechter et al., 1991) have traditionally been limited to hard crisp constraint network lacking expressivity and flexibility. Fuzzy temporal constraint networks were then introduced (Vila & Godo, 1994) proposing a propositional temporal language based on fuzzy temporal constraints to express knowledge and, imprecision as a single type of uncertainty. It provides an inference mechanism involving rules to reason on fuzzy temporal constraints, and ultimately specifying the tightest constraints possible on event duration. In (Godo & Vila, 1995), Godo and Vila proposed a STP-based *Fuzzy Temporal Constraint Networks* in which each constraint representing single time interval, is related to a possibility distribution. Temporal uncertainty is managed using possibility theory (Zadeh, 1975), mainly describing uncertainty with temporal information available in terms of vagueness or imprecision. It provides temporal information consistency-checking to identify potential contradictions and possible scenarios induced by constraints. Even if the framework strictly focuses on possibilities and differs from other kind of uncertainty such as probability, ignoring preferences or exploratory inference on the impact of time-point contingencies on possible/probable variable instantiations, it offers a simple and useful framework to express basic knowledge with natural possibilistic semantics accounting for uncertainty induced by fuzzy temporal constraints. Fuzzy temporal constraints reasoning and time handling and emphasis on imprecision rather than ignorance to deal with uncertainty are naturally well-suited for our targeted application domain. This contrasts with subsequent frameworks described below, building on Simple Temporal Problems with Uncertainty and preferences, bringing unnecessary higher expressivity, query capability, semantic modeling or controllability checking for the problem at hand.

In the early 2000s, research efforts have been directed to further enrich existing frameworks in addressing limited expressiveness and flexibility to deal with preferences and uncertainty Original TCSPs exclusively model hard temporal constraints emphasizing full constraint satisfaction and solution feasibility over partial constraint satisfaction and quality. Accordingly, Khatib et al. (Khatib et al., 2001; Rossi et al., 2002) proposed a generalized TCSP framework introducing a function mapping degree of temporal constraint satisfaction to preferences. Simple Temporal Problems with Preferences (STPPs) (Khatib et al., 2001), tackle the lack of expressiveness of hard temporal constraints by introducing preferences. This is due to the fact that in some real application domains, problem goals may be driven by biases for some solution classes, giving rise to preference-based constraint satisfaction optimality in which computed solution quality is determined in terms of specified preferences. In parallel, Badaloni and Giacomin (Badaloni & Giacomin, 2000) introduced the *Flexible Temporal Constraints* framework. The approach relies on soft constraints to represent preferences among feasible solutions, and, prioritized constraints to characterize constraint satisfaction suitability.

Other researchers refined temporal constraint networks to independently deal with uncertainty by taking into account the contingent nature of some constraints, whose effective duration is dictated by external world events under which the decision support system has no control. This departs from the TCSP framework which assumes that all activities have durations under the control of the agent. The notion of controllability (*strong*, *weak*, and *dynamic*) refers to the agent's (decision-make) ability to control variables in assigning specific values (e.g. time point assignments) with respect to possible exogenous contingent events controlled by the external world (Vidal & Fagier, 1999). The Simple Temporal Problems with Uncertainty (STPUs) framework proposed by Vidal and Fargier, extends STPs incorporating *contingent* events controlled by "Nature", laying emphasis on controllability rather than traditional consistency. As there are no preferences stated explicitly, the focus is on controllability as opposed to optimality. As in STPs, activities durations in STPUs are modeled by intervals, whose start times (time-points) are determined by the agent. Recently, Venable (Venable, 2005) proposed Disjunctive temporal planning with uncertainty, an extension of the disjunctive temporal problem paradigm dealing with event contingency and similar controllability notions. The Probabilistic Simple Temporal Problems (PSTPs) framework has been introduced by Tsamardinos (Tsamardinos et al., 2003a) to handle temporal uncertainty. Similar ideas are presented in (Lau et al., 2005). In that setting, the occurrence of uncontrollable events is governed by a probability distribution rather than intervals. Alternatively, Dubois, HadjAli, and Prade (Dubois et al., 2003b) propose fuzziness in temporal reasoning to deal with uncertainty. They bring in *Fuzzy Allen Relations*, and as in Vila and Godo (Vila & Godo, 1994), consider on available information characterized by imprecision, and vagueness. Focusing on the notions of consistency and entailment, the authors ignore preferences and controllability. Similarly, Dubois, Fargier, and Prade (Dubois et al., 2003a) handle preferences and uncertainty using the fuzzy framework for the classical job-shop scheduling problem characterized by ill-known activity durations. Using possibility theory they consider precedence constraints, capacity constraints and due dates, and release time constraints.

An alternate research direction consists in addressing Simple Temporal Problems with Preferences and Uncertainty (STPPU). Inspired from Simple Temporal Problems with Uncertainty (STPUs) (Vidal & Fargier, 1999), Rossi et al. (Rossi et al., 2006) introduced a new formalism handling both preferences and uncertainty in Simple Temporal Problems. They generalized controllability notions integrating optimality (preferences) and controllability allowing an agent to execute controllable events in a consistent way to meet preferences. The framework provides a way to handle preferences and compute the best solution (rather than a feasible one) through controllability property checking algorithms, in polynomial time. Conditional Temporal Problems (CTP) (Tsamardinos et al., 2003b) has been extended to include preferences (CTPP) (Falda et al., 2007). In CTP, a Boolean formula is attached to each temporal variable, representing preconditions enabling event occurrence. The uncertainty on temporal and conditional plans mainly lies on the selection of temporal variable to be executed. CTPP is a generalization of CTPs in which preferences on temporal constraints are explicitly introduced as well as fuzzy thresholds to govern the occurrence of some events.

3. General framework for fuzzy temporal plan fusion

In (Allouche & Boukhtouta, 2009) a temporal plan is defined as a graph of temporal constraints. It corresponds to a set of temporally constrained actions. Each action is defined by two events: *start* and *end*. Those events form the nodes of the graph. The temporal constraints between those nodes are defined by intervals specifying the minimum and maximum authorized delays between those events. In our new framework, temporal constraints are defined with fuzzy intervals. A temporal plan p_i is defined as the following: $p_i \underset{def}{=} \{A_i, T_i\}$, where $A_i = \{e_1, \ldots, e_n\}$ is a set of action start/end event nodes, and T_i defines fuzzy temporal constraints between nodes.

3.1 Fuzzy temporal constraints

A fuzzy temporal constraint is set between actions start/end nodes. When a temporal constraint is set between the start and the end nodes of the same action, it specifies the minimum and maximum authorized duration for this action. A fuzzy temporal constraint can also synchronize different actions when it is set between their start/end nodes. A temporal constraint is represented by an interval of integers and a function π as shown in Figure 1. It is defined by the function T_i: $A_i \times A_i \rightarrow (I, \pi)$, where I is the set of all integer intervals and π: $I \rightarrow [0\ 1]$ is a *possibility distribution* over temporal distances. π associates a degree of possibility for each value in the interval, that is, it defines the degree of possibility to have a certain temporal distance between two nodes. Our previous definition of a temporal constraint becomes a particular instance of a fuzzy temporal constraint where all values (temporal distance) within the interval have the same degree of possibility equal to 1. A fuzzy temporal constraint can also be expressed by disjunction of several intervals (expressing alternative authorized durations), but this representation goes beyond the scope of this work.

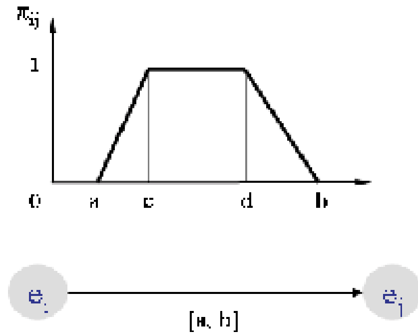


Fig. 1. Fuzzy temporal constraint set between two nodes e_i and e_j

In Figure 1, the fuzzy temporal constraint specifies that e_j must occur at least "*a*" units of time and at most "*b*" units of time after e_i. This is represented by an arrow from e_i to e_j. The function π_{ij} represents the degree of possibility for each element of the interval. *a* and *b* have the minimum degree of possibility. This degree increases as we move toward the center of the interval. In our previous framework the user would have to define the temporal constraint by the interval [*c d*] as shown in Figure 1. In the fuzzy definition of this temporal constraint all values between *a* and *c*, and between *d* and *b* are still acceptable but with decreasing degree of possibility as we move farther from *c* and from *d*. The same fuzzy constraint can be expressed differently by saying that e_i must occur at least "*a*" units of time and at most "*b*" units of time before e_j. Schematically, this is represented by an arrow from e_j to e_i, labelled by the interval [*-b, -a*]. π_{ji} is the symmetric function of π_{ij}, that is, $\pi_{ij}(x) = \pi_{ji}(-x)$ for all *x* in [*a b*]. The two constraints are equivalent, they coexist, and one is called the inverse of the other. It is possible to use $+\infty$ and $-\infty$ in order to define simple temporal precedence. For example, the interval [$0, +\infty$[may be used to specify that e_j must occur at the same time or after e_i. In this case the function π_{ij} should be defined over this interval. The inverse of this constraint is defined by the interval]$-\infty, 0$] and the symmetric function π_{ji}. The constraint defined by]$-\infty, +\infty$[is used to express temporal independence between e_i and e_j. It is called the *universal temporal constraint* and its function $\pi_{ij} = 1$. The null duration constraint is defined by the interval [*0 0*]. Its function π_{ij} is defined as follows: $\pi_{ij} = \begin{cases} 1 & if \quad d = 0 \\ 0 & otherwise \end{cases}$. For example, the null duration constraint will be set on atomic actions that have no duration and are represented by a single node, or on any node e_i to express that an event has no duration (π_{ii} is a null duration function).

The definition of fuzzy temporal actions allows the expression of all Allen's temporal relations (Allen, 1983) where each pair of action nodes is associated to an interval and a possibility function π. These expressions[1] are given in Fig. 2. In this figure, I^-, I^+, J^- and J^+ are action nodes. For the sake of clarity we didn't include the π_{ij} functions associated to the fuzzy temporal constraints. While Allen's relations are qualitative, the proposed representation allows qualitative as well as quantitative temporal relations by simply quantifying the interval of a fuzzy temporal constraint.

[1] In Fig. 2 only seven relations are represented. The converse of these relations (preceded-by, met-by, started-by, finished-by, overlapped-by and contains) are not represented.

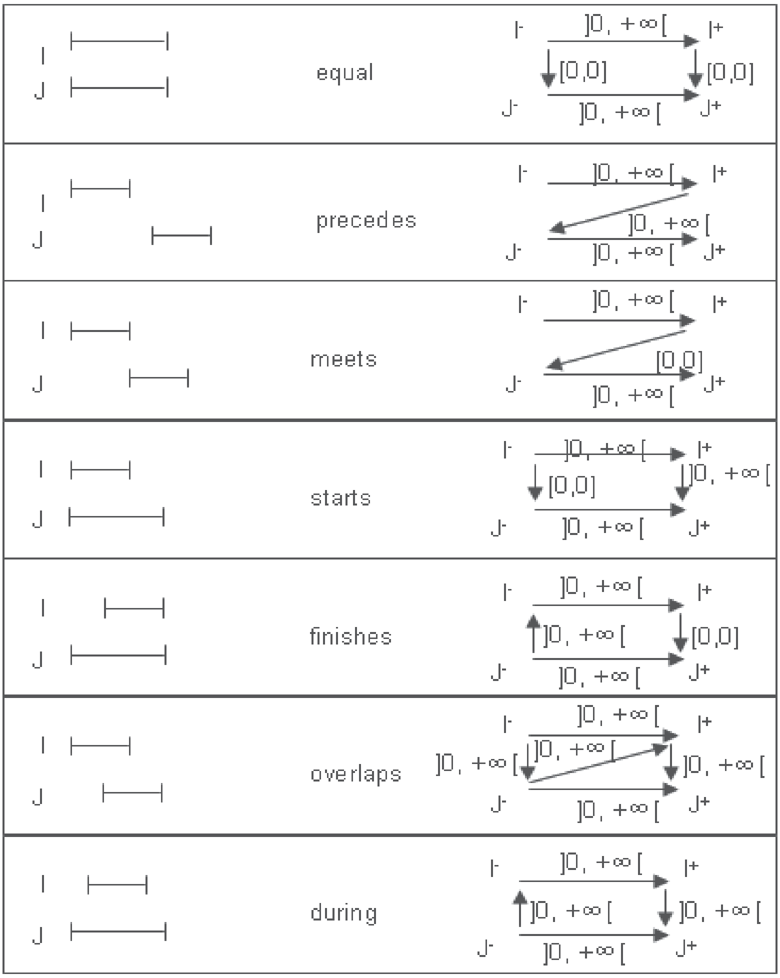


Fig. 2. Temporal constraints expressing the 13 temporal relations defined by Allen

3.2 Operations on fuzzy temporal constraints

As in the previous work, setting fuzzy temporal constraints between a set of events may lead to a temporally incoherent graph. An operation of graph minimization is necessary to check the coherence of the graph. This operation will be described in details in Section 3.3.1. To this end, two operations between fuzzy temporal constraints are needed.

3.2.1 Intersection

The intersection of two fuzzy temporal constraints $\{I_1, \pi_1\}$ and $\{I_2, \pi_2\}$ is also a fuzzy temporal constraint $\{I_1, \pi_1\} \cap \{I_2, \pi_2\} \underset{def.}{=} \{I_1 \cap I_2, \underset{I_1 \cap I_2}{\pi_1} \cap \underset{I_1 \cap I_2}{\pi_2}\}$, where $I_1 \cap I_2$ is the intersection between the two intervals I_1 and I_2, $\underset{I_1 \cap I_2}{\pi_i}$ is the restriction of π_i on the interval $I_1 \cap I_2$.

The intersection operation is illustrated in Figure 3.

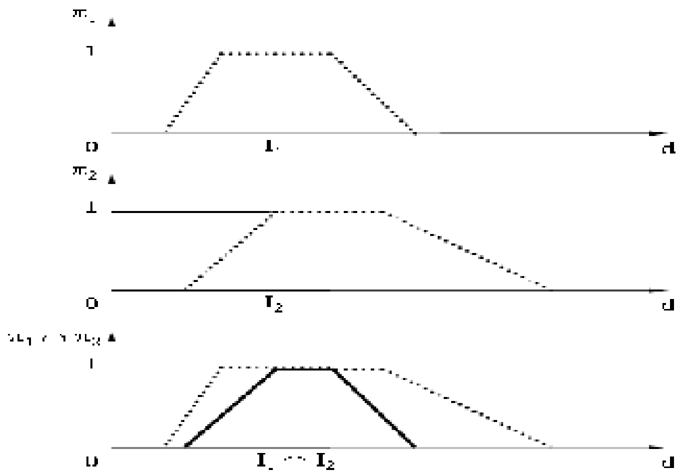


Fig. 3. Intersection of two fuzzy temporal constraints

3.2.2 Composition

The composition of two fuzzy temporal constraints is also a temporal constraint defined by the following: $\{I_1, \pi_1\} \oplus \{I_2, \pi_2\} \underset{def.}{=} \{I_1 + I_2, \underset{d=d_1+d_2}{Sup} \{min\{\pi_1(d_1), \pi_2(d_2)\}\}\}, \forall d \in I_1 + I_2$. The sum of two intervals is defined by the following: $I_1 + I_2 = [a_1, b_1] + [a_2, b_2] \underset{def.}{=} [a_1 + a_2, b_1 + b_2]$. Figure 4 shows the composition of two fuzzy temporal constraints.

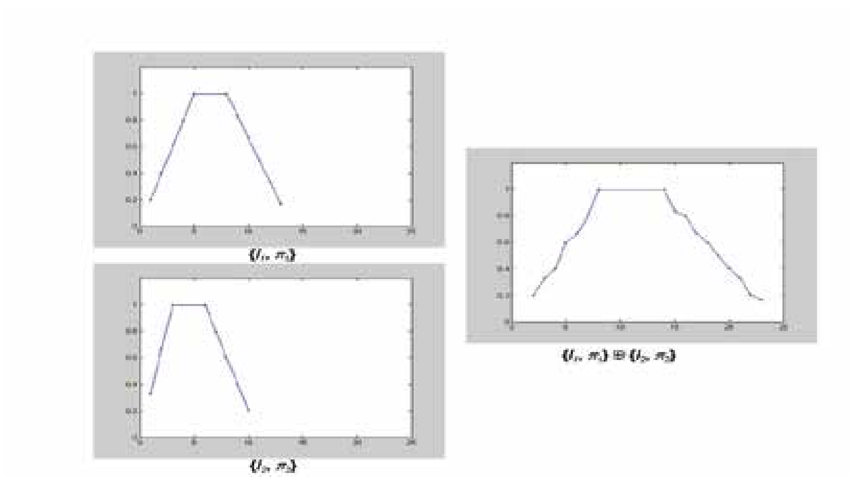


Fig. 4. Composition of two fuzzy temporal constraints

3.3 Operations on fuzzy temporal plans

This section focuses on different operations that may be performed on fuzzy temporal plans. The minimization operation is used to check the temporal consistence of a fuzzy temporal plan. The intersection, augmentation and fusion operations are performed on two different plans and the result is a new fuzzy temporal plan. All these operations are graph-based since fuzzy temporal plans have graph structure.

3.3.1 Minimization

The building of a fuzzy temporal plan may entail temporal incoherencies. Usually, the user will build the graph of fuzzy temporal constraints without checking if there is a feasible solution that will respect all these constraints. To check these potential incoherencies, a minimization operation propagates the fuzzy temporal constraints within the graph in order to obtain its minimal version. When successful, this minimization will generate the minimal version of the graph. The failure of this operation means that the original graph contains temporal incoherencies. Two graphs express the same fuzzy temporal constraints if they have the same minimal graph (if we apply the minimization operation to those graphs). As a direct result, a minimal graph is equal to its minimized version. The minimization operation applies only to complete graphs. For this reason, any graph of fuzzy temporal constraints must be completed with universal fuzzy temporal constraints before minimization. We use the function *comp*(p_i) to complete the graph of the plan p_i.
The algorithm of minimization is the following:

for k=1 to n
  for i=1 to n
    for j=1 to n
      $\{I_{ij}, \pi_{ij}\} \leftarrow \{I_{ij}, \pi_{ij}\} \cap (\{I_{ik}, \pi_{ik}\} \oplus \{I_{kj}, \pi_{kj}\})$
    end
  end
end

In Figures 5 and 6, we show a graph of fuzzy temporal constraints and its minimized version.

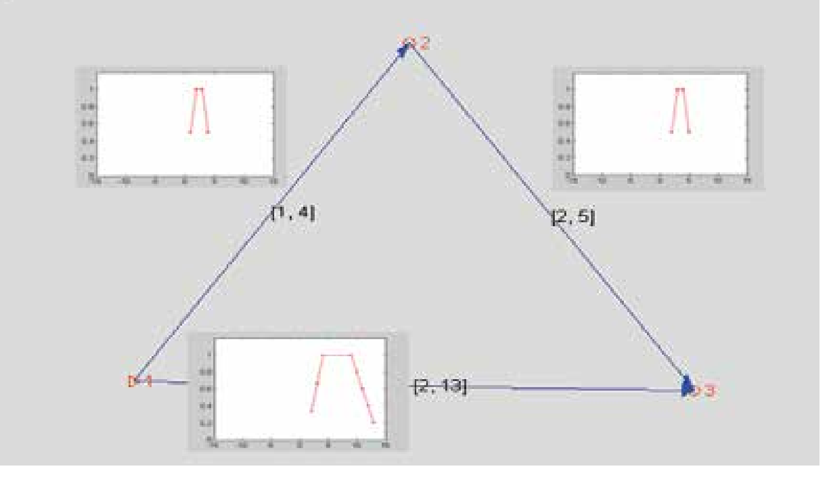


Fig. 5. Fuzzy temporal graph before minimization

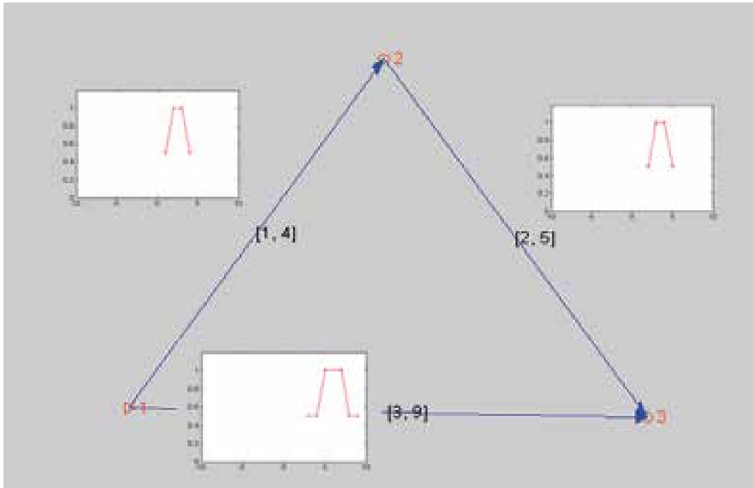


Fig. 6. Minimized version of the graph

Since a graph of fuzzy temporal constraints may be inconsistent, the minimization operation allows detecting a potential inconsistence as shown in Figure 7.

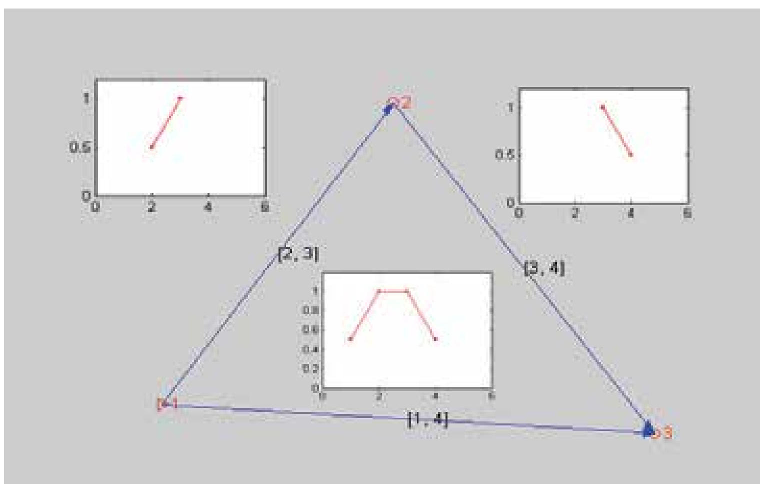


Fig. 7. Temporal inconsistence of a fuzzy temporal plan

In this figure, the composition of C_{12} = {[2 3], {0.5, 1}} and C_{23} = {[3 4], {1, .5}} is $C_{12} \oplus C_{23}$ = {[5 7], {.5, 1, .5}}. In the minimization algorithm, the expression: $\{I_{13}, \pi_{13}\} \leftarrow \{I_{13}, \pi_{13}\} \cap (\{I_{12}, \pi_{12}\} \oplus \{I_{23}, \pi_{23}\}) = \{I_{13}, \pi_{13}\} \leftarrow$ {[1 4], {.5, 1, 1, .5}} $\cap$ ({[2 3], {0.5, 1}} $\oplus$ {[3 4], {1, .5}} = $\{I_{13}, \pi_{13}\} \leftarrow$ {[1 4], {.5, 1, 1, .5}} $\cap$ {[5 7], {.5, 1, .5}}. The intersection between {[1 4], {.5, 1, 1, .5}} and {[5 7], {.5, 1, .5}} is empty. This due to the fact, that the sum of the minimum distances between e_1e_2 and e_2e_3 is greater than the maximum distance e_1e_3: 2+3 > 4. The complexity of the minimization operation is O(n^3), where n is the number of action nodes in the graph of the plan p_i. It is also important to mention that the modeling of a plan with several sub-plans will reduce the number of nodes in the graphs of the temporal plans, which can be executed and monitored in parallel by different agents.

3.3.2 Intersection

The intersection between two fuzzy temporal plans $p_i = \{A_i, T_i\}$ and $p_j = \{A_j, T_j\}$ is defined as follows:

$p_i \cap p_j \underset{def.}{=} \{A_i \cap A_j, \underset{A_i \cap A_j}{T_i} \cap \underset{A_i \cap A_j}{T_j}\}$, where $\underset{A_i \cap A_j}{T_i}$ is the restriction of T_i on $A_i \cap A_j$.

$\underset{A_i \cap A_j}{T_i} \cap \underset{A_i \cap A_j}{T_j} = \{C_{ab} \cap C_{cd} \mid C_{ab} = \{I_{ab}, \pi_{ab}\} \in T_i, C_{cd} = \{I_{cd}, \pi_{cd}\} \in T_j, e_a, e_b \in A_i, e_c, e_d \in A_j, \{e_a, e_b\} = \{e_c, e_d\}\}$. In Figure 8, we illustrate the intersection between two fuzzy temporal plans. It shows the intersection between fuzzy temporal constraints belonging to both plans.

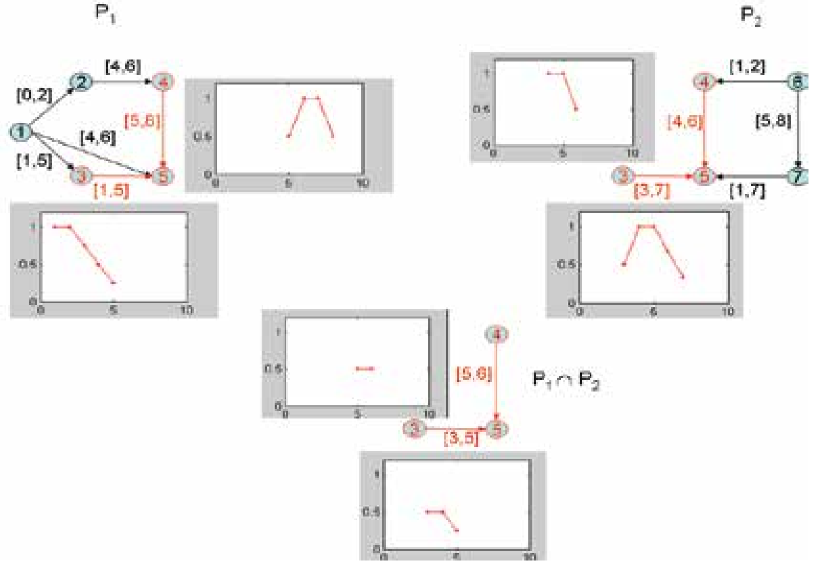


Fig. 8. Intersection of two fuzzy temporal plans

The computational complexity of the intersection operation is O($max(n,m)$), where n and m are the number of nodes in p_i and p_j respectively.

3.3.3 Augmentation

The augmentation operation adds to a plan, the nodes of another plan. This operation is necessary to perform binary operations on graphs of fuzzy temporal constraints:

$$\text{aug}(p_i, p_j) =_{\text{def.}} \text{comp}(\{A_i \cup A_j, T_i\})$$

This operation could also be defined as follows:

$$\text{aug}(p_i, p_j) =_{\text{def.}} \{A_i \cup A_j, \text{ext}(T_i, A_j)\}, \text{ where}$$

$\text{ext}(T_i, A_j)$ is the extension of T_i over A_j. It is defined by the following:

$$ext(T_i, A_j)(e_x, e_y) = \begin{cases} T_i(e_x, e_y) & if \quad e_x, e_y \in A_i \\ \{[-\infty, +\infty], 1\} & otherwise \end{cases}$$

The computational complexity of the augmentation operation is O(n+m).

3.3.4 Fusion of fuzzy temporal plans

The planning of a complex mission may require the participation of different planners (decision makers) from different backgrounds and with different perspectives. Very often this mission is decomposed into a set of sub-missions related with different types of relations. This decomposition draws links and dependencies between the different tasks that must be executed in order to fulfill the sub-missions. The planning of the different sub-missions is a process that can be carried out in parallel with different time constraints. The following example illustrates dependencies between two tasks. In the first task a truck must move from point A to point B. In the second task a tanker full of fuel must take another direction from C to D. If planned separately, the temporal constraints of these tasks should be independent. However, we add the following information: the truck does not have enough fuel to go from A to B and there is an intersection between the trajectory AB and CD. Based on this new information, it is clear that there must be a coordination between these two tasks and more specifically an adaptation of their temporal constraints to ensure their success. To this end, we introduce the fusion operation between two fuzzy temporal plans. The result of fusion is also a fuzzy temporal plan that can be executed by different players. It is called a coordinated plan. The fusion of several fuzzy temporal plans adapts their fuzzy temporal constraints in order to coordinate their execution. The fusion operation is a powerful tool that may be very useful when used in decentralized and distributed environments. It allows decentralized planning of activities and also a coordinated and distributed monitoring of their execution.
The fusion of two temporal plans p_i and p_j is defined by the following:

$$p_i \oplus p_j \underset{def.}{=} min(aug(p_i, p_j) \cap aug(p_j, p_i))$$

By developing this expression according to the definition of *aug* in Section 3.3.3 where $p_i = \{A_i, T_i\}$ and $p_j = \{A_j, T_j\}$, it becomes:

$$\begin{aligned} p_i \oplus p_j &= \min(\{A_i \cup A_j, ext(T_i, A_j)\} \cap \{A_j \cup A_i, ext(T_j, A_i)\}) \\ &= \min(\{A_i \cup A_j, ext(T_i, A_j) \cap ext(T_j, A_i)\}). \end{aligned}$$

The intersection $ext(T_i, A_j) \cap ext(T_j, A_i)$ is based on the intersection of fuzzy temporal constraints given in Section 3.2.1.
The fusion of two temporal plans is obtained by the union of their nodes and the merging of their fuzzy temporal constraints. Common fuzzy temporal constraints between p_i and p_j will be replaced by their intersection. In all other cases, the temporal constraints are unchanged. The coordinated plan must be minimized in order to detect a potential temporal inconsistence, which in this case means that it is not possible to coordinate the execution of the two fused plans and satisfy their corresponding temporal constraints. For this reason, the two temporal plans must be executed independently.
The computational complexity of the fusion operation is $O((n+m)^3)$.
The temporal plan fusion operation has two important properties that determine its context of use in a decentralized distributed environment.

Commutativity

The temporal fusion is commutative:

$$\begin{aligned} p_i \oplus p_j &= min(aug(p_i, p_j) \cap aug(p_j, p_i)) \\ &= min(aug(p_j, p_i) \cap aug(p_i, p_j)) \\ &= p_j \oplus p_i \end{aligned}$$

This property allows a group of players to fuse their plans regardless of the order in which the fusion is being performed.

Associativity

The fusion of temporal plans is associative:

$$\begin{aligned} p_i \oplus (p_j \oplus p_k) &= p_i \oplus (min(\{A_j \cup A_k, ext(T_j, A_k) \cap ext(T_k, A_j)\})) \\ &= min(p_i \oplus \{A_j \cup A_k, ext(T_j, A_k) \cap ext(T_k, A_j)\}) \\ &= min(\{A_i \cup A_j \cup A_k, ext(T_i, A_j \cup A_k) \cap ext(ext(T_j, A_k) \cap ext(T_k, A_j), A_i)\}) \\ &= min(\{A_i \cup A_j \cup A_k, ext(T_i, A_j \cup A_k) \cap ext(T_j, A_k \cup A_i) \cap ext(T_k, A_i \cup A_j)\}). \end{aligned}$$

It is easy to demonstrate in the same way that $(p_i \oplus p_j) \oplus p_k = min(\{A_i \cup A_j \cup A_k, ext(T_i, A_j \cup A_k) \cap ext(T_j, A_k \cup A_i) \cap ext(T_k, A_i \cup A_j)\})$.
This property allows a group of players to fuse any number of plans in any order to obtain the same coordinated plan.

3.4 Fuzzy temporal plan monitoring

The monitoring of a graph-structured fuzzy temporal plan differs from the monitoring of a threadlike plan. In the latter, each action is scheduled and executed according to its rank in the list. In graph-structured fuzzy temporal plans, each time an action is started/finished the fuzzy temporal constraints must be updated and then propagated throughout the graph by performing the minimization operation. This propagation tells if there still a solution for the execution of the plan while respecting the current fuzzy temporal constraints. It also gives the time left for an action to start or finish executing and the list of actions (candidates) that can be executed in the next step.

3.4.1 Timeout

The timeout is computed after each action start or end. Let e_i be the last occurred node in the graph, the timeout T is computed as the following: for all $C_{ij} = \{[a_{ij}, b_{ij}], \pi_{ij}\} \mid b_{ij} \geq 0$, $T = \min_j (b_{ij})$. b_{ij} is noted $max(C_{ij})$, hence, $T = \min_j (max(C_{ij}))$. It corresponds to the minimum of maximum authorized times for the next node e_j to occur. In fact, past this time at least one action will be considered as too late to be carried out after e_i.

3.4.2 Candidate list

The candidate list L contains all the nodes that are authorized to occur after e_i. Obviously, an action from this list must be executed before the expiration of the timeout. Suppose that C_{is} is the fuzzy temporal constraint that allowed the computing of the timeout after the occurrence of e_i. All the nodes e_j such as $C_{is} \cap C_{ij} \neq \emptyset$, can occur (candidates) after e_i. This is true because it is possible to execute the corresponding actions before the timeout expiration and without violating the corresponding temporal constraints.

3.4.3 Propagation of fuzzy temporal constraints

The propagation of fuzzy temporal constraints must be performed each time an action starts or finishes executing. After propagation, the timeout and the candidate list are recomputed. When a new node in the graph occurs (the execution of an action starts or finishes), the fuzzy temporal constraint between this node and the previous occurred node is updated by the new fuzzy temporal constraint {[*d d*], *v*} where *d* is the exact elapsed time between the two nodes and $v \in [0\ 1]$ is the possibility value associated to this distance. The choice of *v* can be user defined or based on different criteria. It is 1 when this distance is confirmed with certainty, and takes lower values otherwise. The constraint is propagated by minimizing the graph. The monitoring process is illustrated by the same example given in (Allouche & Boukhtouta, 2009) by adding fuzzy temporal constraints in the graph. The original minimal graph is illustrated in Figure 9. In this figure, the possibility functions of all fuzzy temporal constraints are represented in a 5×5 grid.

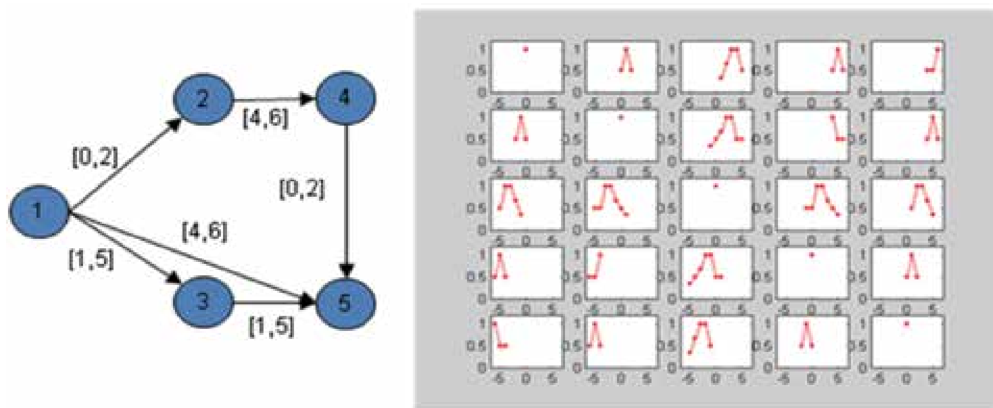


Fig. 9. Original minimal graph

In this example, we will not show the values of the timeout and candidate list since they are the same as in (Allouche & Boukhtouta, 2009). Only the change of fuzzy temporal constraints will be shown throughout the execution of the plan.

The execution of the plan starts with the occurrence of Node 1 at 15:06:12. Then Node 2 occurs at 15:06:13 with a possibility value = 1. The new graph is represented in Figure 10.

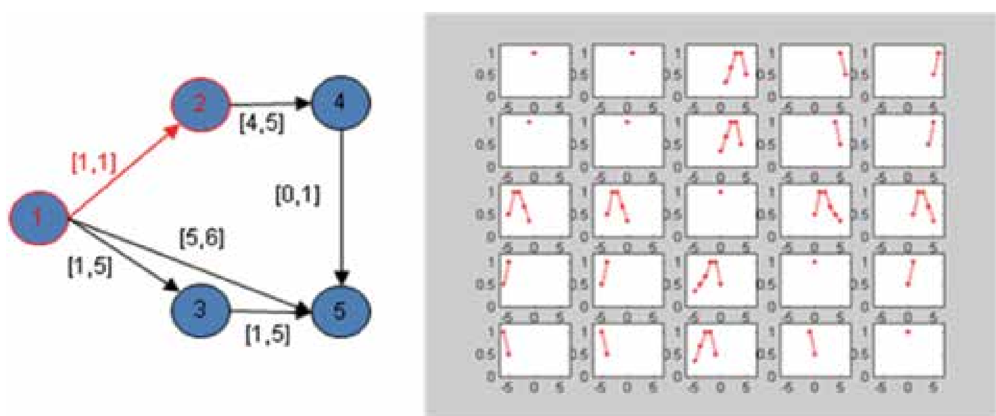


Fig. 10. Occurrence of the Node 2 at 15:06:13

Node 3 occurs at 15:06:15 with a possibility value = 0.5. The new graph is illustrated in Figure 11. It is important to see in this figure that all the fuzzy temporal constraints in the graph have now their possibility values at 0.5.

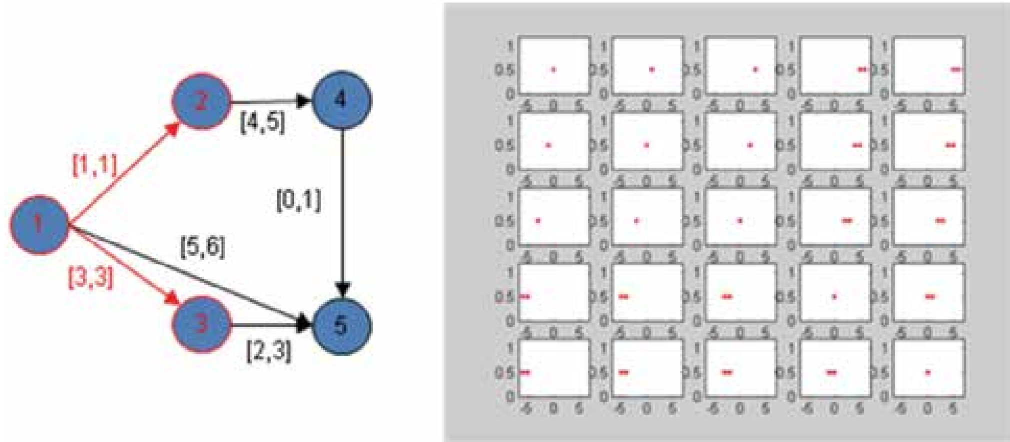


Fig. 11. Occurrence of Node 3 at 15:06:15

Finally Node 4 and Node 5 occur at 15:06:17 with possibility values .75 and 1 respectively. The result is depicted in Figure 12.

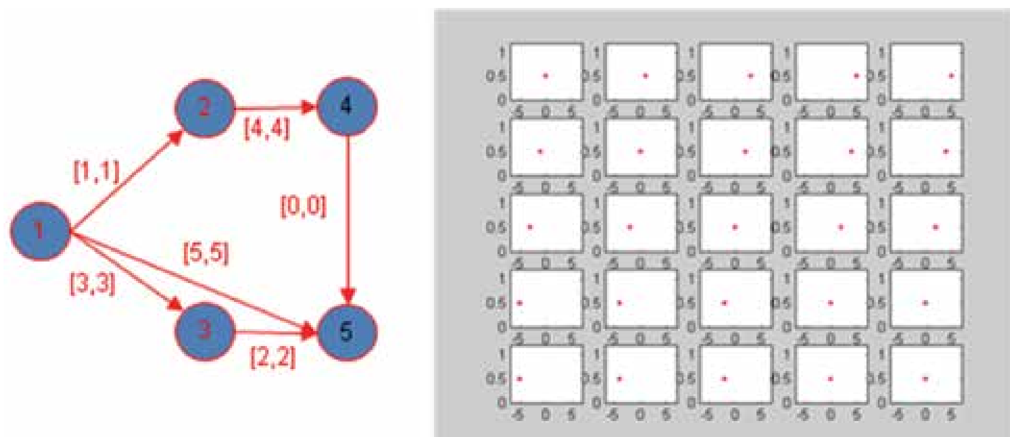


Fig. 12. Occurrence of Node 4 and Node 5 at 15:06:17

This example shows how the introduction of fuzzy information provides the monitoring process with more flexibility since the execution continued with possibility values less than 1. This flexibility should however be controlled by the decision-maker in order to give a meaning of a specific possibility value. In fact, the decision-maker is faced with two different problems. The first is to give a possibility value to the occurrence of a node. The second is to be able to qualify a possibility value in the graph that is less than 1. Mainly, the main question that the decision-maker would be eager to answer is: "should this specific possibility value be acceptable or not".

4. Application to CSAR

The fuzzy plan fusion is applied to a CSAR (Combat Search and Rescue) mission in the context of the North Atlantis scenario. This fictitious scenario was used in 2000 as an exercise by the Canadian Forces Command and Staff College (CFCSC) to teach the Canadian Forces Operational Planning Process and allow the sharing of operational knowledge and expertise among the CFC staff and students. The choice of this type of application is motivated by three main reasons: First, the temporal constraints are key elements in planning and the execution of CASR missions. Second, a CASR mission requires different types of assets distributed over the environment. Finally, a close coordination of the activities of the different involved assets is a key factor for the mission success.

A crisis has developed over the past 10 days on the continent of Atlantis. It is the result of years of growing tensions since the fall of 1999, and has now erupted into armed conflict. Individual country studies are provided as well as a document entitled "The Manghalour Peninsula Crisis," to provide the detailed background to the crisis.

As a result of the critical situation between ORANGELAND/REDLAND and BLUELAND, the UN requested the Alliance Council to consider a military response to help resolve the crisis.

On 12 June, the second day following the commencement of the Alliance joint operations to secure Blueland territory and expel any Coalition invasion forces, a UK Royal Air Force (RAF) Tornado call-sign HAWK27, conducting an electronic countermeasures and reconnaissance (ECR) mission, was shot down over the Celtic Straits by a surface-to-air missile (SAM) at 1608 hours. Shortly after the location of the downed crew was known, a CH-124 Sea King helicopter from Wahhabe Airbase, with a crew of five, was sent to recover and evacuate the Tornado aircrew. At approximately 1800 hours, in the process of extraction of the downed Tornado crew, the Sea King crashed.

A CSAR mission represents many dynamic challenges for the mission planners to locate and extract lost crew members in a hostile environment. Various elements must be taken into account, which may be predictable such as the friendly elements of detect and rescue, or unpredictable such as the enemy elements of detect and destroy. Usually, mission planners use air and ground picture inputs to make their decisions.

4.1 Tasks

The PC (Package Commander) designed a plan to meet two critical mission requirements: air superiority and CSAR extraction. This plan should also allow SEAD (Suppression of Enemy Air Defenses) and air superiority established at least 15 minutes ahead, ABR (Airborne Regiment) delay/harassment occurring approximately 10 minutes ahead and CAS (Close Air Support) in place five (5) minutes ahead of the CSAR helicopter extraction area TOT (Time On Target). The plan includes tasks assignment to allocated assets to counter the enemy threat for "efficiency and safety" and to gain and maintain local air superiority:

a. 4 x CF-18 – SIERRA 1-4 – sweep ingress and egress route and provide CAP (Combat Air Patrol) over CSAR pick-up area (above cloud);
b. 4 x CF-18 – ECHO 1-4 – escort CSAR helicopters inbound and outbound to the pick-up area (below cloud with assets);
c. 4 x CF-18 – BOMBER 1-4 – BAI (Battlefield Air Interdiction)(cluster munitions) pre-strike harass/delay of the LOC (Lines Of Communication) and ABR main body forward elements;
d. 4 x ECR Tornado – JAMMER 1-4 – SEAD of Eaglevista SAMs from five (5) minutes before to five (5) minutes after mission aircraft enter AOO (Area Of Operations);

e. 2 x ECR Tornado – ZAP 1-2 – SEAD of ABR SA 8 ahead of sweep aircraft and remain on station until all mission aircraft out of SA 8 range;
f. 2 x CH 53 – RESCUE 1-2 - each with maximum JTF2 (Joint Task Force)(less seven (7) for downed crews) such that each can carry out mission if other helicopter aborts;
g. 1 x AC-130 (Gunship) – GUNNER - for CAS in the target area;
h. 1 x Predator UAV – PREDATOR 1 – to locate and monitor pick-up area;
i. 1 x Predator UAV – PREDATOR 2 – to locate and assist in targeting ABR forward elements.

In this plan, the assets used such as the 4 x CF-18 in task c) for instance, correspond to agents that are responsible to execute this task. BOMBER 1-4 are the names of these agents. The organizational structure of agents and the means they have in order to form coalitions and help each other will not be discussed in this chapter.

4.2 Temporal constraints

From the description of the plan and related tasks, it is possible to deduce different temporal constraints between the different tasks.

- f) during b);
- f) during a);
- g) starts at least 5 min. before f) starts;
- c) starts at least 10 min. before f) starts;
- d) starts at least 15 min. before f) starts;
- e) starts at least 15 min. before f) starts;
- f) during h);
- e) during i);
- f) during e);
- e) overlaps a);
- d) starts at least 5 min. before and continues at least 5 min. after e) starts;
- f) must be performed between 10 and 20 min.

4.3 Temporal plans

The tasks described in Section 4.1 allow the definition of partial plans that must be executed by the different agents to insure the success of the CSAR mission. However, if the plans are executed separately, without taking into account the different temporal constraints, synchronisation problems may arise causing the whole CSAR mission to fail. For example, if f) is not executed during b), the extraction of the downed crew, at least in part, will be performed without escort, which will put the downed crew in danger. The idea is to define a plan for each agent or group of agents. Once the plans are defined, they must be fused to obtain a coordinated plan that can be executed by all the agents. Nine (9) plans can be defined from the tasks and related temporal constraints. They are shown in Figure 13. In these sub-plans the fuzzy constraints are those represented in figure 14. For each interval derived from the constraints defined in Section 4.2, we added 20% as a possibility distribution. The definition of this distribution is very important in order to cope with any unlikely delay that may alter the success of the mission if any of the temporal constraints defined in Section 4.2 is violated. In Figure 14, the fuzzy temporal constraint represented in (a) belongs to P3 and P7, (b) belongs to P4, (c) belongs to P6 and (d) belongs to P7 and P8.

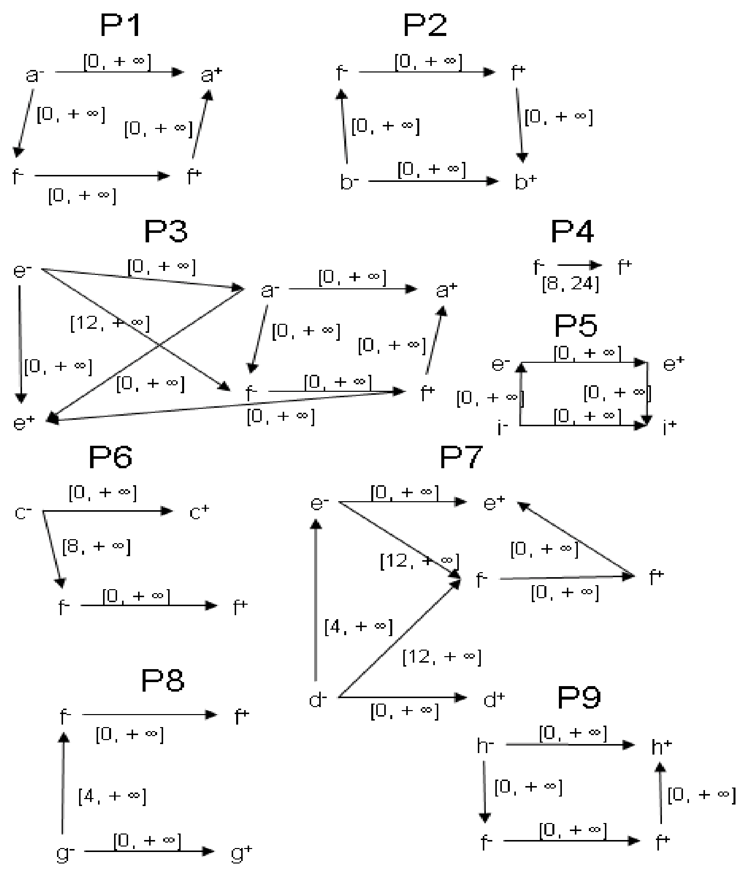


Fig. 13. Plans for CSAR mission

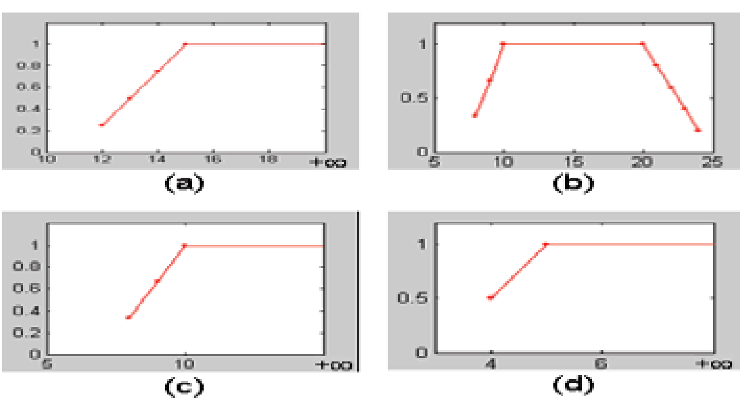


Fig. 14. Fuzzy temporal constraints in the sub-plans

4.4 Temporal fusion

The fusion of the nine plans is shown in Figure 15.

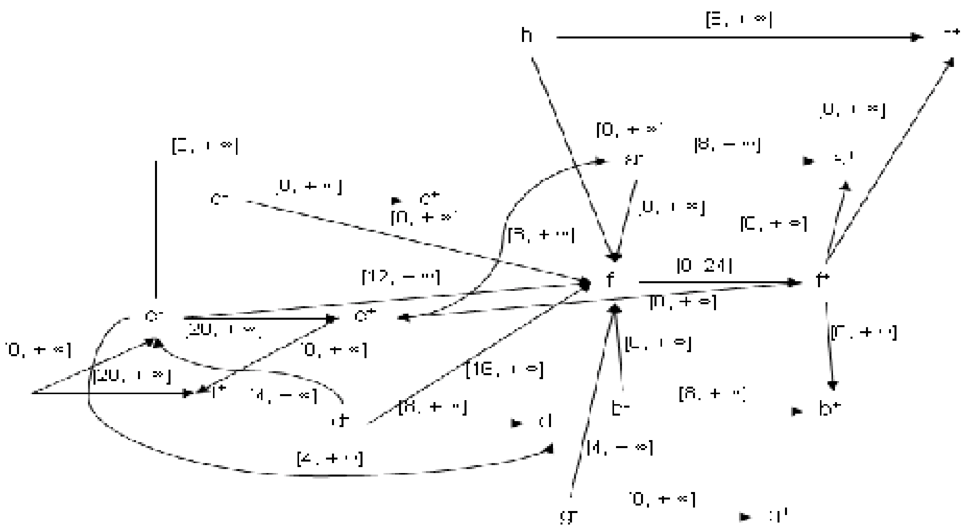


Fig. 15. Coordinated plan obtained by fuzzy temporal fusion of the nine plans

For the sake of clarity we show in Figure 16 only four fuzzy temporal constraints in the graph of the coordinated plan. Those constraints are the result of the propagation performed in the fusion process.

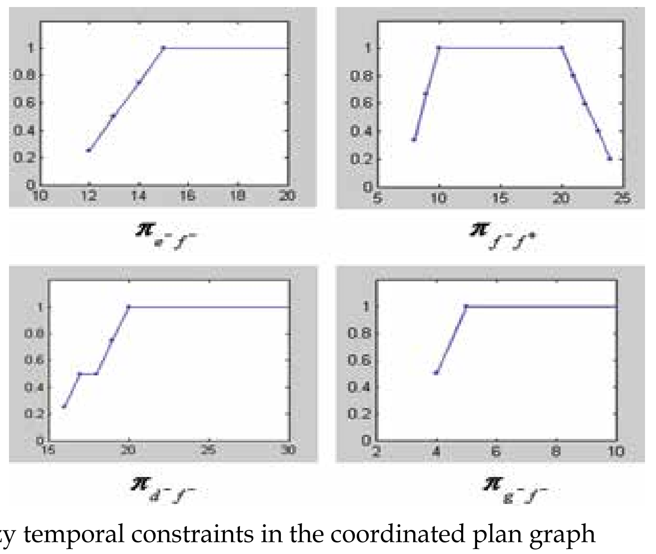


Fig. 16. Four fuzzy temporal constraints in the coordinated plan graph

It is also important to mention that for the sake of simplicity, details in the tasks such as take-offs, air-to-air refuelling, and landing were not taken into account in the plans. For example, the plan P4 is simply defined by the task f), which corresponds to the extraction of the downed crew. However, before the extraction, details such as the take-off of the rescue helicopters, the path followed before reaching the crash site, and finally the return to base, were not included in the plan. Also, all the communications between the PC, mission aircraft and other centres such as the Combined Air Operations Centre (CAOC) are not shown.

Some of the details will be shown in the timeline of the mission in Section 4.5. It is possible to use sub-plans to model such details. The use of sub-plans would allow different levels of abstraction of the problem.

4.5 Mission execution

The following is a general timeline giving a global picture of the execution of the mission. Each element of this timeline includes the hour, the aircraft name and a summarized description of its activity. As mentioned before, even if this timeline is too detailed for the coordinated plan, it still contains little details compared to the original mission timeline. We think that this level of details is sufficient to illustrate the fuzzy temporal fusion and monitoring of the execution of a distributed CSAR mission.

The nodes in the coordinated plan are shown as they occur within the following timeline. We have chosen to show the execution of the coordinated plan at three time points within the timeline: at 11:15, then at the end of the extraction and finally at the end of the mission. At each time point, the propagation of the fuzzy temporal constraints is shown as well as the positions of the different aircraft in the area of operations. In this timeline, we suppose to know exactly when a node occurs. However, the formalism allows the use of fuzzy temporal constraints to express an uncertainty about the occurrence of nodes.

06:00 Magic (1 x AWACS) on station north at 5800N/2600W

07:00 Predator 1 (1 x UAV) take-off from Bendeguz, Exxon 1 (1 x KC-135 AAR (Air-to-Air Refuel)) on station Track A 5830N 2400W

08:00

- **h- Predator 1 on station 5700N/2700W, detect and track downed aircrew, detect and track enemy forces**

09:00 Predator 2 (1 x UAV) take-off from Bendeguz

09:45 Echo 1-2 (2 x CF-18) take-off from Bendeguz

- **i- Predator 2 on station 5730N/2630W, airborne backup**

10:00 Spook (1 x JSTARS) on station 5720N/2400W, detect and track enemy forces (ABR and SA-8 TELs), Jammer 1-4 (4 x Tornado ECR) take-off from Bendeguz, Bomber 1-4 (4 x CF-18) take-off from Bendeguz, Exxon 2 (1 x KC-135 AAR) on station Track B 5830N 2400W

10:15 Echo join AAR Track B

10:30 Zap 1-2 (2 x Tornado ECR) take-off from Bendeguz, Jammer 1-4 AAR Track A, Echo 1-2 departs AAR Track B, Bomber joins AAR Track B, Gunner (1 x AC-130) take-off from Nitric

10:35 Rescue 1-2 (2 x CH-53) take-off from Nitric

10:45 Echo 3-4 (2 x CF-18) take-off Bendeguz

11:00 Jammer 1-4 departs AAR, Zap 1-2 joins AAR Track A, Sierra 1-4 (4 x CF-18) take-off from Bendeguz, Bomber 1-4 departs AAR Track B

11:10 Predator 2 departs north hold to reposition to 5640N/2700W, Predator 3 (1 x UAV) takes-off from Bendeguz

11:15

- Rescue 1-2 turn south along coast
- **b- Echo 1-2 join CSAR for close escort**
- Echo 3-4 join AAR Track B
- Zap 1-2 departs AAR Track A

Fig. 17. Assets positions at 11:15

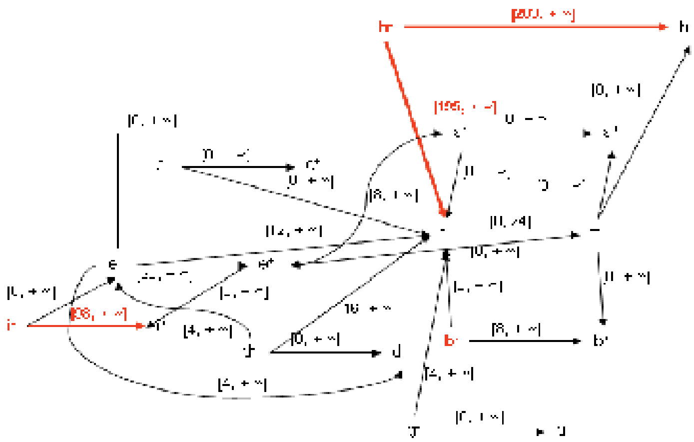


Fig. 18. Fuzzy temporal constraints of the coordinated plan at 11:15

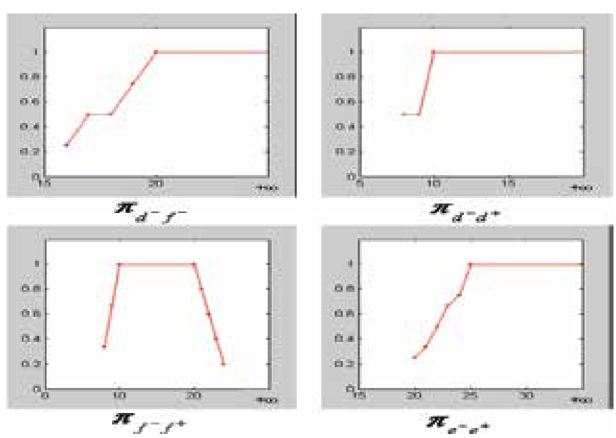


Fig. 19. Possiblity functions of four fuzzy temporal constraints of the coordinated plan at 11:15

<u>11:20</u>

d- Jammer 1-4 push from 5730N/2400W

<u>11:25</u> Jammer 1-4 (Low) ingress over Blueland, SEAD at Eaglevista, Zap 1-2 push from 5750N/2500W

<u>11:30</u> Sierra 1-4 Sweep, push from 5800N/2500W, Echo 3-4 depart AAR Track B, Predator 2 on station 5640N/2700W, Predator 1 repositions to 5700N/2730W

<u>11:35</u> Bomber 1-4 push from 5800N/2900W

<u>11:40</u> Gunner push from 5800N/2800W

<u>11:45</u>

e- Zap 1-2 engaged SEAD SA-8

j- Predator 3 on station 5730N/2630W, airborne backup

<u>11:50</u>

a- Sierra 1-4 on CAP bullseye 5700N/2700W (southwest)

c- Bomber 1-4 TOT BAI LOC Cluster munitions

<u>11:55</u>

g- Gunner TOT CAS

<u>12:00</u>

f- Rescue 1-2 TOT extraction begins

- Echo 1-2 provide top cover in target area
- Echo 3-4 arrive to provide top cover with Echo 1-2

c+ Bomber 1-4 RTB (Return To Base) Bendeguz

<u>12:22</u>

f+ Rescue 1-2 extraction complete

- Echo 1-2 RTB
- Echo 3-4 close escort CSAR egress

g+ Gunner RTB Nitric

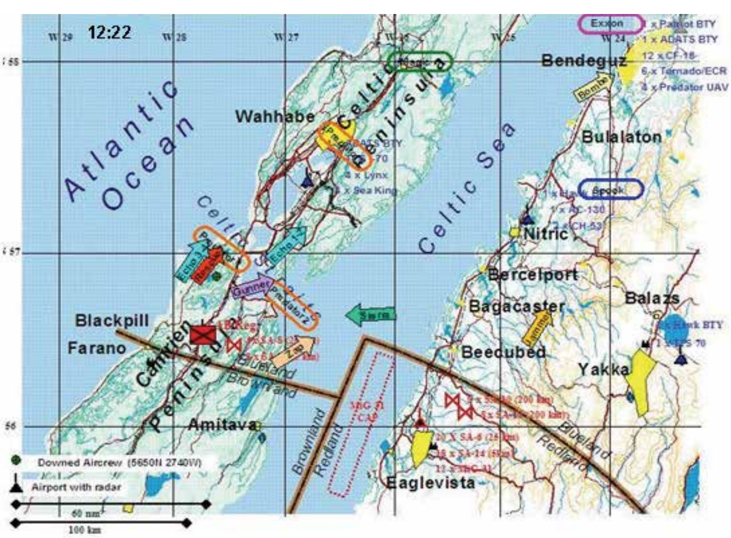


Fig. 20. Assets positions at 12:22

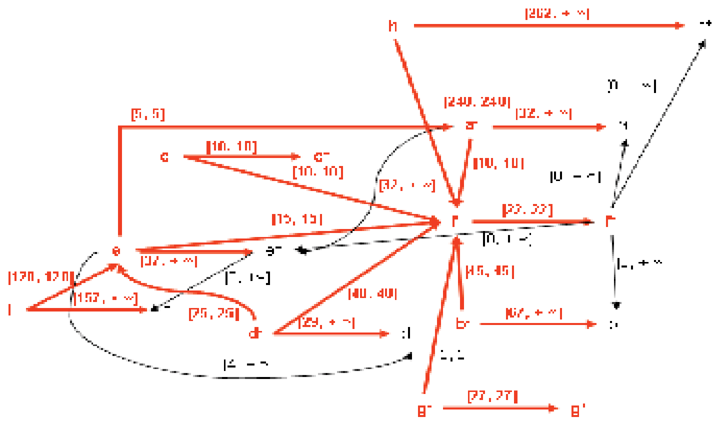


Fig. 21. Fuzzy temporal constraints of the coordinated plan at 12:22

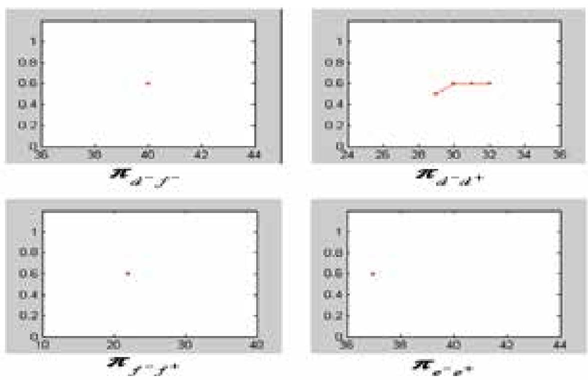


Fig. 22. Possiblity functions of four fuzzy temporal constraints of the coordinated plan at 12:22

<u>12:25</u>

d+ Jammer 1-4 RTB Bendeguz

<u>12:30</u>

a+ Sierra 1-4 RTB Bendeguz

e+ Zap 1-2 RTB Bendeguz

h+i+ Predator 1-2 RTB Bendeguz

<u>12:45</u>

- Rescue 1-2 lands at Wahhabe

b+ Echo 3-4 RTB Bendeguz

The propagation of the temporal constraints depicted in Figures 17-25, clearly show that the activity described in the timeline is compliant with the coordinated plan shown in Fig. 15. It is also important to mention that without the use of fuzzy temporal constraints the execution of the coordinated plan should fail. In fact, since the extraction task took 22 min, the possibility function of all the fuzzy temporal constraints has a value smaller than one as shown in Figure 22. For example, if the extraction of the downed crew had taken less than 20min, the propagation of temporal constraints would lead to possibility functions equal to 1. If the extraction had taken more than 24 min, it is then necessary either to change the plan (re-planning) or to adapt the temporal constraints. In both cases the coordination of agents' activities is necessary.

5. Conclusion and perspectives

In this work, we propose a general framework for distributed fuzzy temporal plan modelling and monitoring. We believe that the explicit representation of time in plan modelling needs also to take into account the representation of uncertainty. This is due to the fact that in distributed environments where different activities may take place at the same time, it is sometimes difficult to manage the synchronisation of tasks with precision.

Fig. 23. Assets positions at the end of the mission

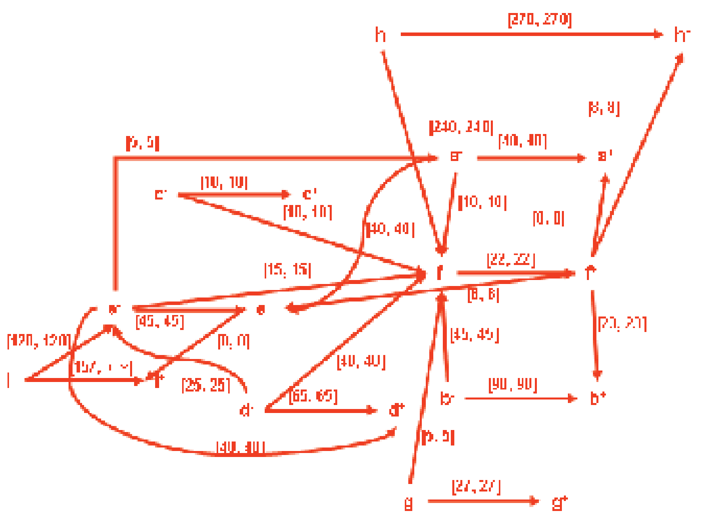


Fig. 24. Fuzzy temporal constraints of the coordinated plan at the end of the mission

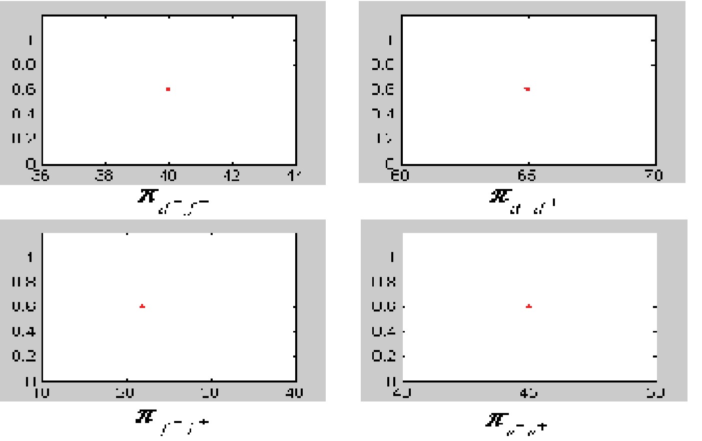


Fig. 25. Possibility functions of four fuzzy temporal constraints of the coordinated plan at the end of the mission

The fuzzy temporal plan model allows to model plans as a set of fuzzy and temporally constrained actions. Each action is modelled by two nodes (beginning and end). A fuzzy temporal constraint is defined by an interval where each value in this interval has a possibility value.

A propagation mechanism is defined in order to check the temporal consistence of the fuzzy temporal plan during execution.

In a distributed environment different activities are carried out simultaneously. This corresponds to the execution of different sub-plans by different players. A fusion operation is defined in order to fuse different sub-plans into a single plan called coordinated plan. A coordinated plan can be executed by different partners and the propagation mechanism is used to check its temporal consistence during the execution.

The approach only computes the possible solutions (coordinated plan) to execute distributed temporal sub-plans by different players. For instance, after each action execution, different actions may be candidate for execution. A decision must be made in order to choose the next action to be executed. In an ideal context, the proposed fusion and monitoring capability should be combined with a decision support capability and should keep the human operator in the loop to make decisions.

A CSAR mission was chosen to illustrate this work on a real-world scenario, where temporal aspects and uncertainty are key factors for the mission success.

One limitation of plan fusion is that each time a plan is fused with another plan, they become more temporally constrained. Hence, the fusion of large number of temporal plans tends to result in a temporally inconsistent coordinated plan. This indicates that plan fusion is useful in some coordination contexts but a re-planning activity may become unavoidable in some other cases. The introduction of fuzzy temporal constraints helps mitigate this problem since it is possible to extend the original temporal constraints with the appropriate possibility functions.

As future work, it would be interesting to give an interpretation of the possibility function values. For example, after the end of the extraction task, all the possibility functions have a maximum value of 0.6 as shown in Figure 22. This is due to the fact that the extraction task took 22 min, which exceeds the original definition of the corresponding temporal constraint given in Section 4.2. The question that needs to be answered is how a decision-maker should interpret a value lower than 1 of the possibility function. This value could be used to define a measure of performance for the mission execution.

6. References

Allen, J. F. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11) pp.832-843, Nov. 1983.

Allouche, M.K. and. Boukhtouta A. (2009). Multi-Agent Coordination by Temporal Plan Fusion - Application to Combat Search and Rescue, *Information Fusion*, doi:10.1016/j.inffus.2009.09.005

Badaloni, S., and Giacomin, M. (2000). Flexible temporal constraints. In *8th Conference on Information Processing and Management of Uncertainty in knowledge-Based System* (IPMU 2000), pp. 1262–1269.

Dechter, R., Meiri, I., and Pearl, J. (1991). Temporal constraint networks. *Artificial Intelligence*, 49(1-3), 61–95.

Dubois, D., Fargier, H., and Prade, H. (2003a). Fuzzy scheduling: Modelling flexible constraints vs. coping with incomplete knowledge. *European Journal of Operational Research*, 147, 231–252.

Dubois, D., HadjAli, A., and Prade, H. (2003b). Fuzziness and uncertainty in temporal reasoning. *Journal of Universal Computer Science*, 9(9).

Falda, M., Rossi, F., Venable , K.B. (2007). Dynamic consistency in fuzzy conditional temporal problems, *PROC. COPLAS* 2007, CP'07 WORKSHOP.

Godo L., Vila, L. (1995). Possibilistic Temporal Reasoning based on Fuzzy Temporal Constraints, *Proceedings of International Joint Conference on Artificial Intelligence* IJCAI.

Khatib, L., Morris, P., Morris, R. A., and Rossi, F. (2001). Temporal constraint reasoning with preferences. In Nebel, B. (Ed.), *17th International Joint Conference on Artificial Intelligence*, (IJCAI 2001), pp. 322–327. Morgan Kaufmann.

Lau, H. C., Ou, T., and Sim, M. (2005). Robust temporal constraint networks. In Proc. of 17th *IEEE Conf. on Tools with Artificial Intelligence* (ICTAI'05), pp. 82–88 Hong Kong.

Rossi, F., Sperduti, A., Venable, K., Khatib, L., Morris, P., and Morris, R. (2002). Learning and solving soft temporal constraints: An experimental study. In Van Hentenryck, P. (Ed.), Principles and Practice of Constraint Programming, *8th International Conference (CP 2002)*, Vol. 2470 of LNCS, pp. 249–263. Springer.

Rossi, F., and Venable, K.B., Yorke-Smith, N. (2006). Uncertainty in soft temporal constraint problems: a general framework and controllability algorithms for the fuzzy case, *Journal of Artificial Intelligence Research* 27.

Schwalb, E. (1998). Temporal Constraints: A Survey. *Constraints*, Volume 3, Numbers 2-3.

Schwalb, E., Dechter, R., (1997). Processing Disjunctions in Temporal Constraint Networks, *Artificial Intelligence*, 29-61.

Tsamardinos, I., Pollack, M. E., and Ramakrishnan, S. (2003a). Assessing the probability of legal execution of plans wit temporal uncertainty. In *Workshop on Planning Under Uncertainty and Incomplete Information of the Thirteenth International Conference on Automated Planning and Scheduling* (ICAPS 2003).

Tsamardinos, I., Vidal, T., Pollack, M. E. (2003b). CTP: A new constraint-based formalism for conditional, temporal planning. *Constraints*, 8(4), 365–388.

Venable, K.B. (2005). Disjunctive temporal planning with uncertainty, *19th International Joint Conference on Artificial Intelligence.*

Venable, K. B. (2005). Disjunctive temporal planning with uncertainty. *19th International Joint Conference on Artificial Intelligence* IJCAI.

Vidal, T., Fargier, H. (1999). Handling contingency in temporal constraint networks: from consistency to controllabilities. *Journal of Experimental and Theoretical Artificial Intelligence*, 11(1), 23–45.

Vila, L., and Godo, L. (1994). On fuzzy temporal constraint networks. *Mathware and Soft Computing*, 3, 315–334.

Zadeh, L. A. (1975). Calculus of fuzzy restrictions. *Fuzzy Sets and their Applications to Cognitive and Decision Processes*, 1–40.

15

Robust Consensus of Multi-agent Systems with Bounded Disturbances

Lin Wang[1] and Zhixin Liu[2]
[1]*Shanghai Jiao Tong University*
[2]*Academy of Mathematics and Systems Science, Chinese Academy of Sciences*
China

1. Introduction

Recent years have witnessed an increasing interest in the coordination of distributed multi-agent systems. In this area, one of the most fundamental problems is the consensus problem, which has broad connections with a wide range of disciplines including statistical decision theory [1, 2], distributed computing [3, 4], biology [5, 6], and cooperation of multi-agent systems[7, 8].

Consensus roughly speaking is characterized as a collection of agents with locally sensed information or limited inter-component communications seeking to reach a common value. A basic consensus protocol in the context of multi-agent systems usually consists of a information exchange network, in which each agent updates its state by forming a convex combination of the states of its neighbors and itself. Some sufficient consensus conditions for the heading angles of a group of agents modeled by Vicsek et al. [5] are presented in [9, 10]. Some less restrictive conditions were obtained in [12, 13], where consensus is ensured if there exists a spanning tree in the union of the information exchange networks. Many other generalizations have been devoted to different types of agent dynamics and different topologies of information exchange networks, such as nonlinear consensus protocol [14], consensus of agents modeled by double integrators [15], consensus algorithm with cohesion, separation and alignment rules [16, 17], consensus over random networks [18, 19], consensus of networked agents with time-delays [20, 21].

Most of the previously mentioned references used noise-free state iteration, that is they assume the information exchange between agents is accurate. This assumption would obviously be inappropriate in real distributed systems, since there are various kinds of noises during the sending, transmission and receiving of information. Consensus of distributed systems with noise disturbance is an important challenge, and now there are only a few results. The average-consensus control with fixed topology and additive input noises is investigated in [22], where the long term consensus error is minimized by a least mean square optimization method. [23] considered the consensus protocol with fixed topology and independent identically distributed noises, and used stochastic Lyapunov functions to establish mean square consensus. The extension to the case of time-varying topologies is carried in [24], where some sufficient conditions are given for mean square average-consensus and almost sure consensus. [11] further investigated decentralized adaptive synchronization for a stochastic model with uncertainties. Roughly speaking, the consensus algorithms in these researches are essentially distributed stochastic approximation type consensus protocols, and the consensus gains are essential for the consensus properties.

In this paper, we investigate the consensus problem of multi agent systems with bounded disturbances. The information exchange topology is time-varying, and the consensus protocol has a general form

$$x(t+1) = P(t)x(t) + w(t+1), t = 1, 2, \cdots,$$

where $x(t) \in \mathbf{R}^n$ is the information state, $w(t) \in \mathbf{R}^n$ is the noise disturbance, and $\{P(t)\}$ is a sequence of stochastic matrices. Here, the noise is bounded, and does not require the zero-mean property needed by the previous stochastic approximation type consensus protocols. For the system, it is expected that small disturbance of $\{w(t)\}$ would give rise to small discrepancy of the states of agents from each other, namely robust consensus in this paper.

We first investigate the case that the information exchange between the agents is bidirectional. From the well-known relationship between the external stability and the internal stability, we know that the exponential stability of the corresponding homogeneous dynamical equation will play a crucial role. Based on this observation, we establish the equivalence among the robust consensus, the positivity of the second smallest eigenvalue of the weighted Laplacian matrix, and the connectivity of the union graph, where several basic results on stability of time-varying systems [28], product of nonnegative stochastic matrices[27, 31, 32], algebraic graph theory [30] and Markov chains [29] are useful.

Since there are a variety of practical applications where information only flows in one direction, the case of interaction topologies with directed information exchanges is also studied. In this case, we show that the robust consensus can be achieved under dynamically changing interaction topology if and only if the union of the collection of interaction graphs across some time intervals has a spanning tree frequently enough.

The paper is partially based on the results in [25, 26]. Some new insights and simulations are added to illustrate the theoretical results. Section 2 recalls some basic notions and motivates the problem to be investigated; the case of undirected information exchange topology is studied in Section 3 while the directional case is studied in Section 4; Some simulation results are given in Section 5; and a brief conclusion is made in Section 6.

2. Preliminaries and problem statement

2.1 Preliminaries

The information exchange between agents can always be represented by directed/undirected graphs. A *directed graph* $G = (V, E)$ consists of a vertex set V and an edge set $E = \{(i, j)\}$, where $V = \{1, 2, \cdots, n\}$ is composed of the indices of all agents, and $E \subseteq V \times V$ is a set of ordered pairs of vertexes. As a comparison, an *undirected graph* means the pairs of vertexes in the edge set are unordered. If there is a directed edge from vertex j to i, then j is defined as a *parent* vertex and i is a *child* vertex, which means that i can receive the information from agent j. The *neighbors* of agent i are denoted by $N_i = \{j \subseteq V : (j, i) \in E\}$. A *path* that connects i and j in a directed/ undirected graph G is a sequence of distinct vertexes $i_0, i_1, i_2, \cdots i_m$, where $i_0 = i, i_m = j$ and $(i_l, i_{l+1}) \in E, 0 \leq l \leq m-1$. A graph is called *connected* (or *strongly connected* for directed graph) if for every pair of distinct vertexes there is a path connecting them. A digraph is said to have a *spanning tree* if and only if there exist a vertex $i \in V$, called *root*, such that there is a path from i to any other vertex. The *union* of a collection of graphs

$\{G_1, G_2, \cdots, G_h\}$ with the same vertex set V, is a graph G with vertex set V and edge set equaling the union of the edge sets of those graphs.

The *adjacency matrix* $A = [a_{ij}]$ of a weighted graph G is defined as $a_{ij} > 0$ if $(j, i) \in E$. The adjacency matrix of a weighted undirected graph is defined analogously except that $a_{ji} = a_{ji}$, $\forall i \neq j$, since the edge is unordered and the same for its two adjacent vertexes. The *degree matrix* $D = [d_{ij}]$ is a diagonal matrix with $d_{ii} = \sum_{j=1}^{n} a_{ij}$. The *Laplacian matrix* $L = [l_{ij}]$ is defined as $L = D - A$, which implies that 0 is one of its eigenvalues. Moreover, 0 is a simple eigenvalue if the graph is strongly connected [31]. For an undirected graph, L is symmetric positive semi-definite. For a positive semi-definite matrix B, we arrange all its eigenvalues in a nondecreasing order: $0 \leq \lambda_0(B) \leq \lambda_1(B) \leq \cdots \leq \lambda_{n-1}(B)$. In some cases, we are interested in the second smallest eigenvalue $\lambda_1(B)$.

Some notations from nonnegative matrix theory are important for investigating the consensus property [29, 31]. A matrix is *nonnegative (positive)* if all its entries are nonnegative (positive). Moreover, if the sum of each row satisfies $\sum_{j=1}^{n} a_{ij} = 1, i = 1, \cdots, n$, the matrix is called *stochastic*. A stochastic matrix P is said to be *indecomposable and aperiodic* (SIA) if $\lim_{k\to\infty} P^k = \mathbf{1}v^{\tau}$, where $\mathbf{1}$ is a column vector of all ones and v is some column vector. Define

$$\lambda(P) = \min_{i,j} \sum_{s=1}^{n} \min(P_{is}, P_{js}). \tag{1}$$

If $\lambda(P) > 0$, then P is called *scrambling matrix*. For a matrix $P = [p_{ij}]_{n\times n}$, its associated directed graph $\Gamma(P)$ is a directed graph on n nodes $1, 2, \cdots, n$ such that there is a directed arc in $\Gamma(P)$ from j to i if and only if $p_{ij} \neq 0$(cf. [31]).

2.2 Problem statement

Let $x_i(t) \in \mathbf{R}$, $i = 1, \cdots, n$ represent the information state of agent i at time t. As described in [1, 2, 7–10], a discrete-time consensus protocol can be summarized as

$$x_i(t+1) = \frac{1}{\sum\limits_{j\in N_i(t)} a_{ij}(t)} \sum_{j\in N_i(t)} a_{ij}(t) x_j(t), \tag{2}$$

where $a_{ij}(t) \geq 0$ represents the weighting factor, and $N_i(t) = \{j : a_{ij}(t) > 0\}$ is a set of agents whose information is available to agent i at time t.

In the real world, the outside interference and measurement error are unavoidable. Each agent receives in fact noisy information from its neighbors. Assume the resulting information of agent j received by agent i is the following form:

$$y^{ij}(t) = x_j(t) + e^{ij}(t),$$

where $e^{ij}(t)$ is the noise. The update law of agent i under the influence of noise can be described as

$$\begin{aligned} x_i(t+1) &= \frac{1}{\sum\limits_{j\in N_i(t)} a_{ij}(t)} \sum_{j\in N_i(t)} a_{ij}(t) y^{ij}(t) \\ &= \frac{1}{\sum\limits_{j\in N_i(t)} a_{ij}(t)} \sum_{j\in N_i(t)} a_{ij}(t) x_j(t) + \frac{1}{\sum\limits_{j\in N_i(t)} a_{ij}(t)} \sum_{j\in N_i(t)} a_{ij}(t) e^{ij}(t). \end{aligned} \tag{3}$$

Let $G_t = (V, E_t)$ represent the neighbor graph that $(j,i) \in E_t$ iff $j \in N_i(t)$. Let $A(t) = [a_{ij}(t)]$ be the adjacency matrix that $a_{ij}(t) > 0$ iff $j \in N_i(t)$. Let $D(t)$ be the associated degree matrix. Let $w_i(t) = \frac{1}{\sum_{j \in N_i(t)} a_{ij}(t)} \sum_{j \in N_i(t)} a_{ij}(t) e^{ij}(t)$. Then the matrix form of system (3) is

$$x(t+1) = D^{-1}(t)A(t)x(t) + w(t), \quad t = 1, 2, \cdots, \tag{4}$$

where $x(t) = [x_1(t), \cdots, x_n(t)]^\tau$ is the vector formed by the states of all agents, and $w(t) = [w_1(t), \cdots, w_n(t)]^\tau$ is the noise vector. Define $P(t) = D^{-1}(t)A(t)$. It's easy to check that $P(t)$ is a stochastic matrix, and $P(t) = I - D^{-1}(t)L(t)$ with $L(t)$ being the Laplacian matrix. The system (4) can be rewritten as

$$x(t+1) = P(t)x(t) + w(t), \quad t = 1, 2, \cdots. \tag{5}$$

In this paper, we propose the following assumption on the matrix $A(t) = [a_{ij}(t)]$, which is simple and easily satisfied.

Assumption $\mathbf{\Lambda}$:

(1) For each t, $A(t)$ has positive diagonal entries, i.e. $a_{ii}(t) > 0$;

(2) There exist two constants α, $\beta > 0$ such that $\alpha \le a_{ij}(t) \le \beta$ for all $a_{ij} \ne 0$.

Assumption $\mathbf{\Lambda}(1)$ means that each agent can sense its own information, and Assumption $\mathbf{\Lambda}(2)$ means that the information exchange between two neighboring agents has some bounds.

The purpose of this paper is to study the consensus property of system (4). Generally, by consensus we mean that for any two agents i and j, their states satisfy $\lim_{t \to \infty} \|x_i(t) - x_j(t)\| = 0$. In the presence of noise, we should not expect that the agents can reach consensus eventually. So, we introduce a concept— robust consensus to describe the influence of the noise to the behavior of the system. Define the distance between a vector x and a subspace $X \subset \mathbf{R}^n$ as

$$d(x, X) = \inf_{y \in X} d(x, y) = \inf_{y \in X} \|x - y\|, \tag{6}$$

where $\| \cdot \|$ is the standard Euclidean norm. In this paper, we take X as the space spanned by the vector $[1, 1, \cdots, 1]^\tau \in \mathbf{R}^n$, i.e., $X = span\{[1, 1, \cdots, 1]^\tau\}$, and denote the orthogonal complement space of X by M. Define a function set and a noise set as follows:

$$K_0 = \{f(\cdot) | f : R^+ \to R^+, f(0) = 0, f(\delta) \text{ decreases to 0 as } \delta \to 0\};$$
$$\mathcal{B}(\delta) = \Big\{ \{w(t)\} | \sup_{t \ge 0} d(w(t), X) \le \delta \Big\}.$$

Definition 2.1. System (4) is said to be robust consensus with noise, if there exist a function $f(\cdot) \in K_0$ and a constant $T > 0$ such that for any $\delta > 0, x(0) \in \mathbf{R}^n$, and any sequence $\{w(t)\} \in \mathcal{B}(\delta)$,

$$d(x(t), X) \le f(\delta), \ t \ge T. \tag{7}$$

Remark 2.1. If the noise vector $w(t) = c(t) \cdot \mathbf{1}$ with $c(t) \in \mathbf{R}$ being very large, then it may have strong influence on the states of the agents but have no influence on the consensus property, since the noise disturbance can be eliminated when considering the difference of the states between agents. This is the reason that we use $d(w(t), X)$ rather than $\|w(t)\|$ to describe the noise effect here.

3. Undirected information exchange topology

In this section, we assume the information exchange topology is undirected, that is the weighted adjacency matrix $A(t)$ is symmetric. In this case, the Laplacian matrix $L(t) = D(t) - A(t)$ is positive semi-definite.
The following theorem is our main result.

Theorem 1. *Consider System (4) under Assumption* $\mathbf{\Lambda}$*. If the weighted adjacency matrix* $A(t)$ *is symmetric, then the following three propositions are equivalent:*
(i) The system (4) is robust consensus.
(ii) For the Laplacian matrix $L(t)$*, there exists a constant* $q > 0$ *such that*

$$\inf_{t\geq 0} \lambda_1 \Big(\sum_{k=t+1}^{k=t+q} L(k) \Big) \neq 0. \tag{8}$$

(iii) There exists a constant $q > 0$ *such that for any* $t \geq 0$*, the union of the neighbor graphs* $\{G_{t+1}, G_{t+2}, \cdots, G_{t+q}\}$ *is connected.*

For readability, we divide the proof into the following three subsections.

3.1 The proof of (i) $\Rightarrow$ (ii)

By introducing a suitable projection operator, we can translate the distance between a vector and the subspace X into the norm of the projected vector, so the problem of robust consensus can be transformed into a certain robust stability in the subspace. We decompose the space $\mathbf{R}^n$ into two orthogonal subspaces X and $M = X^{\perp}$. As X and M are closed subspaces, we know that for any $x \in \mathbf{R}^n$, there exists a unique pair of vectors $x_0 \in M, x_1 \in X$ such that $x = x_0 + x_1$. Furthermore, according to the property of projection, we have $\|x_0\| = \|x - x_1\| = \inf_{y\in X} \|x - y\| = d(x, X)$. Denote by P_M the projector onto M. Then $P_M x = x$ if and only if $x \in M$, $P_M x = 0$ if and only if $x \in X$, and

$$\|P_M x\| = d(x, X). \tag{9}$$

Take a standard orthogonal base $e_1, e_2, \cdots, e_n$ in the space $\mathbf{R}^n$, where $e_n = [\frac{1}{\sqrt{n}}, \cdots, \frac{1}{\sqrt{n}}]^{\tau}$. Then $X = span\{e_n\}$ and $M = span\{e_1, e_2, \cdots, e_{n-1}\}$.We can get a detailed form of the projector P_M as follows:

$$P_M = Q \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix} Q^{\tau} = I - \frac{1}{n}\mathbf{1} \cdot \mathbf{1}^{\tau},$$

where $Q = [e_1, e_2, \cdots, e_n]$. The projector P_M has the following property.

Lemma 1. *If* P_1, P_2 *are stochastic matrices, and* P_M *is the projector onto M, then we have*

$$P_M P_1 P_M P_2 = P_M P_1 P_2. \tag{10}$$

Proof. By using the expression of the projector P_M and the properties of the stochastic matrix, we have

$$P_M P_1 P_M P_2 = P_M P_1 (I - \frac{1}{n}\mathbf{1}\cdot\mathbf{1}^\tau)P_2 = P_M(P_1 - \frac{1}{n}\mathbf{1}\cdot\mathbf{1}^\tau)P_2 = P_M P_1 P_2.$$

■

Moreover, if $\{P_i, i = 1, \cdots\}$ are stochastic matrices, by the above lemma and the fact that the product of stochastic matrices is still stochastic, we can deduce that

$$\prod_{i=1}^{n} P_M P_i = P_M \prod_{i=1}^{n} P_i. \tag{11}$$

Now, letting the projector P_M act on both sides of the system (5), we have

$$P_M x(t+1) = P_M P(t)x(t) + P_M w(t). \tag{12}$$

Set $\eta(t) = P_M x(t)$, $\nu(t) = P_M w(t)$, then $\eta(t), \nu(t) \in M$, and

$$\begin{aligned} & P_M P_t \eta_t = P_M P_t P_M x(t) = P_M P_t\big(I - \tfrac{1}{n}\mathbf{1}\cdot\mathbf{1}^\tau\big)x(t) \\ = \;& P_M P_t x(t) - P_M P_t \tfrac{1}{n}\mathbf{1}\cdot\mathbf{1}^\tau x(t) = P_M P_t x(t) - \tfrac{1}{n}P_M\mathbf{1}\cdot\mathbf{1}^\tau x(t) \\ = \;& P_M P_t x(t). \end{aligned} \tag{13}$$

So system (5) is equivalent to

$$\eta(t+1) = P_M P(t)\eta(t) + \nu(t), \tag{14}$$

where $\eta(t), \nu(t) \in M$. Set $\nu = \{\nu(t)\}_{t=1}^{\infty}$, $U_M(\delta) = \{\nu : \|\nu(t)\| \le \delta,\ \nu(t) \in M,\ \forall t \ge 1\}$. Here, system (14) is said to be robust stable on the subspace M, if for any $\eta(1) \in M$ and any $\varepsilon > 0$, there exist constants $\delta = \delta(\varepsilon, \eta(1)) > 0$, $T = T(\varepsilon, \eta(1)) > 0$ such that $\sup\limits_{\nu\in U_M(\delta)} \sup\limits_{t\ge T} \|\eta(t)\| \le \varepsilon$.
Thus, robust consensus of system (5) has been transformed into the robust stability of system (14) on the subspace M.
We define $\Phi(k, i)$ as the state transition matrix of (14), that is

$$\Phi(k+1, i) = P_M P(k)\Phi(k, i), \quad \Phi(i, i) = I, \quad \forall k \ge i \ge 0. \tag{15}$$

Then, we have

$$\eta(t+1) = \Phi(t+1, 0)\eta(0) + \sum_{i=1}^{t} \Phi(t+1, i+1)\nu(i). \tag{16}$$

To motivate further study, we introduce the following exponential stability lemma.

Lemma 2. *Consider system (14) with $P(t)$ being stochastic matrix. If the system is robust stable on the subspace M, then there exist constants $N > 0$ and $\lambda \in (0, 1)$ such that*

$$\| \Phi(k+h, k) \| \le N\lambda^h, \quad \forall k \ge 0, \quad \forall h \ge 1. \tag{17}$$

The proof is in Appendix 7.
The following lemma will establish the relationship between the exponential stability and the second smallest eigenvalue of $\sum L(t)$.

Lemma 3. *Consider system (14) with $P(t) = D^{-1}(t)A(t)$. Assume the weighted adjacency matrix $A(t)$ satisfies Assumption $\mathbf{\Lambda}$ and is symmetric. If there exist constants $N > 0$ and $\lambda \in (0,1)$ such that $\|\Phi(k+h,k)\| \leq N\lambda^h$ for any $k \geq 0, h \geq 0$. Then, there must exist a constant $q > 0$ such that*

$$\inf_{k\geq 0} \lambda_1\Big(\sum_{k}^{k+q-1} L(t)\Big) \neq 0,$$

where $L(t) = D(t) - A(t)$ is the Laplacian matrix.

In order to maintain the integrity of the context, the proof of this Lemma is also presented in Appendix 7.
Finally, by combining Lemma 2 with Lemma 3, we can finish the proof of (i) $\Rightarrow$ (ii).

3.2 The proof of (ii) $\Rightarrow$ (iii)

The following theorem provides the relationship between the connectivity of the graph and the eigenvalues of the weighted Laplacian matrix.

Theorem 2. *Let G be a graph and L be its weighted Laplacian matrix. Then, the eigenvalue 0 of L is simple if G is connected. Moreover, if the algebraic multiplicity of the eigenvalue 0 of L is c, then the graph G has exactly c connected components.*

Proof. The first part can be obtained from Lemma 13.9.1 of ref. [30]; The second part follows from the fact that the union of two disjoint graph has as its spectrum the union of the spectra of the original graphs. ■

Since $\sum_{k=t+1}^{k=t+q} L(k)$ is a weighted Laplacian matrix of the union graph of $\{G_{t+1}, G_{t+2}, \cdots, G_{t+q}\}$, by Theorem 2, we know that $\lambda_1(\sum_{k=t+1}^{k=t+q} L(k)) \neq 0$ is equivalent to the connectivity of the union graph. Thus the Proposition (ii) $\Rightarrow$ (iii) of Theorem 1 is true.

3.3 The proof of (iii) $\Rightarrow$ (i)

The union of graphs is closely related to the product of stochastic matrices. The following theorem provides a relationship between the connectivity of the union graph of $\{G_{t_1}, \cdots, G_{t_m}\}$ and the matrix products $P(t_1)P(t_2)\cdots P(t_m)$.

Lemma 4. *[9] Let $\{P(t)\}$ be stochastic matrix with positive diagonal entries, and G_t be the associated graph. If $\{G_{t_1}, \cdots, G_{t_m}\}$ is jointly connected, then the product of matrix $P(t_1)P(t_2)\cdots P(t_m)$ is SIA.*

Let $\mathcal{P}$ be a matrix set. By a word (in the $\mathcal{P}$) of length m we mean the product of m $\mathcal{P}'s$ (cf. [27]). From the proof of Lemma 4 in ref. [27], we can see that the following result is also true.

Lemma 5. *Let $\{P_i\}_{i=1}^{\infty}$ be a stochastic matrix sequence denoted by $\mathcal{P}$. If any word in the $\mathcal{P}$'s is SIA, then there exists a constant $T^* > 0$ such that all words in the $\mathcal{P}$'s of length $\geq T^*$ are scrambling matrices, where T^* only depends on the dimension of the matrix.*

For a stochastic matrix $P = \{P_{ij}\}$, define

$$\tau(P) = \frac{1}{2}\max_{i,j} \sum_{s=1}^{n} |P_{is} - P_{js}|. \tag{18}$$

From ref. [29], we know that $\tau(P) = 1 - \lambda(P)$, and for any stochastic matrices $P(1)$ and $P(2)$, we have

$$\tau(P(1)P(2)) \le \tau(P(1))\tau(P(2)). \tag{19}$$

Furthermore, the function $\tau(\cdot)$ also has the following property.

Lemma 6. *[29] Let $y = [y_1, \cdots, y_n]^\tau \in \mathbf{R}^n$ be an arbitrary vector and $P = [P_{ij}]_{n\times n}$ be a stochastic matrix. If $z = Py,\ z = [z_1, \cdots, z_n]^\tau$, then we have*

$$\max_{s,s'} |z_s - z_{s'}| \le \tau(P) \max_{j,j'} |y_j - y_{j'}|.$$

We also need the following simple lemma, whose proof is in Appendix 7.

Lemma 7. *Let $z \in \mathbf{R}^n$, $\Delta z = \max_{i,j} |z_i - z_j|$, P_M be the projector onto M. Then*

$$\frac{\sqrt{2}}{2}\Delta z \le \|P_M z\| = d(z, X) \le \sqrt{n}\Delta z.$$

The proof of (iii) $\Rightarrow$ (i) :
For system (5), we define the state transition matrix as follows

$$\Phi^*(k+1, i) \quad = \quad P(k)\Phi^*(k, i), \Phi^*(i, i) = I, \quad \forall k \ge i \ge 0,$$

then we have

$$x(t+1) = \Phi^*(t+1, 0)x(0) + \sum_{i=1}^{t} \Phi^*(t+1, i+1)w(i). \tag{20}$$

By applying Lemma 6, we have

$$\Delta x(t+1) \le \tau(\Phi^*(t+1, 0))\Delta x(0) + \sum_{i=1}^{t} \tau(\Phi^*(t+1, i+1))\Delta w(i).$$

From Lemma 4, we know that $\Phi^*(t+q+1, t+1)$ is SIA for any t. Furthermore, by combining with Lemma 5, for any $t \ge 0$, we have

$$\Phi^*(t+L, t) = \prod_{k=t}^{k=t+L-1} P(k) \text{ is a scrambling matrix,}$$

where $L = qT^*$ is a constant. From (2) of Assumption $\mathbf{\Lambda}$, we know that there exists a constant $\bar{\alpha} > 0$ such that all the non-zero entries of $P(t)$ are larger than or equal to $\bar{\alpha}$. Then we have

$$\lambda(\Phi^*(t+L, t)) \ge \bar{\alpha}^L.$$

Hence

$$\tau(\Phi^*(t+L, t)) = 1 - \lambda(\Phi^*(t+L, t)) \le 1 - \bar{\alpha}^L = \sigma, \quad \forall t \ge 0.$$

For any $t \ge 0$ and $h \ge 0$, there exists an integer $k_0 \ge 0$ such that $k_0 L < h \le (k_0+1)L$. By (19), we have

$$\begin{aligned} \tau(\Phi^*(t+h, t)) \quad &\le \quad \tau(\Phi^*(t+h, t+k_0L)) \cdot \tau(\Phi^*(t+k_0L, t+(k_0-1)L) \cdots \Phi^*(t+L, t)) \\ &\le \quad \sigma^{k_0} \le \sigma^{\frac{h}{L}-1} = \sigma^{-1}(\sigma^{\frac{1}{L}})^h. \end{aligned}$$

Define $N = \sigma^{-1},\ \lambda = \sigma^{\frac{1}{L}}$. Then $0 < \lambda < 1$ and

$$\tau(\Phi^*(t+h,t)) \leq N\lambda^h, \quad \forall t \geq 0, h \geq 0, \tag{21}$$

where N and λ are independent of t and h. From (21), we have

$$\lim_{t\to\infty} \tau(\Phi^*(t+1,0)) \leq \lim_{t\to\infty} N\lambda^{t+1} = 0. \tag{22}$$

Moreover, for any $t \geq 0$, we have

$$\sum_{i=1}^{t} \Phi^*(t+1,i+1) \leq \sum_{i=1}^{t} N\lambda^{t-i} < \frac{N}{1-\lambda}.$$

Thus

$$\sup_{t\geq 0} \sum_{i=1}^{t} \Phi^*(t+1,i+1) \leq \frac{N}{1-\lambda}. \tag{23}$$

By (22), for any $\delta > 0$, there exists $T > 0$ such that

$$\tau(\Phi^*(t,0)) \leq \frac{\delta}{\Delta x_0}, \quad \forall t \geq T. \tag{24}$$

For any $\{w(t)\}$ satisfying $\sup_{t\geq 0} d(w(t), X) \leq \delta$, by Lemma 7, we have $\sup_{t\geq 0} \Delta w(t) \leq \sqrt{2}\delta$, which, in conjunction with (23) and (24), yields

$$\Delta x(t) \leq \delta + \frac{\sqrt{2}N}{1-\lambda}\delta = (1 + \frac{\sqrt{2}N}{1-\lambda})\delta, \quad \forall t \geq T.$$

By taking $f(s) = \sqrt{n}(1 + \frac{\sqrt{2}N}{1-\lambda})s$, obviously we have $f(\cdot) \in K_0$, and

$$d(x(t), X) \leq f(\delta), \quad \forall t \geq T.$$

Thus, we complete the proof of (iii) ⇒ (i) of Theorem 1. ■

Remark 1. *From Lemma 2, we know that the exponential stability of the projected system (14) is essential for the robust consensus. However, to the best of our knowledge, almost all the existing results only analyzed the asymptotic stability of the projected system, which might not be powerful enough for dealing with the influence of noise.*

4. Directed information exchange topology

In this section, we will generalize the above results to the case of directed information exchange. The information exchange with directed topology can be found in many biological, social, and engineering systems, such as the "leader-follower" model where the leader can influence the followers while the followers can not influence the leader.
The main result of this section are summarized in the following theorem.

Theorem 3. *Consider the system (4) under Assumption* **A**. *Then it is robust consensus if and only if there exists a constant $q > 0$ such that for any $t \geq 0$ the union of neighbor digraphs $\{G_{t+1}, G_{t+2}, \cdots, G_{t+q}\}$ has a spanning tree.*

To prove the main result, we need the following lemmas which is about the algebraic multiplicity of the eigenvalue 1 of stochastic matrix.

Lemma 8. *Let P be a stochastic matrix with P_M being its projector onto M. If $\rho(P_M P) < 1$, then 1 is a simple eigenvalue of P, where $\rho(A)$ is the spectral radius of a matrix A.*

Proof From the theorem of finite dimensional Markov chain (see [32]), we know that there exists a matrix K such that

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} P^i = K.$$

Thus

$$\lim_{n\to\infty} P_M \frac{1}{n}\sum_{i=1}^{n} P^i = P_M K.$$

On account of $\rho(P_M P) < 1$, there exists a matrix norm $\|\cdot\|_l$ such that

$$\|P_M P\|_l \doteq \mu < 1.$$

By (11),

$$\|P_M \frac{1}{n}\sum_{i=1}^{n} P^i\|_l = \|\frac{1}{n}\sum_{i=1}^{n}(P_M P)^i\|_l \leq \frac{1}{n}\sum_{i=1}^{n}\|P_M P\|_l^i \leq \frac{1}{n}\sum_{i=1}^{n}\mu^i.$$

Thus

$$\lim_{n\to\infty} \|\frac{1}{n}\sum_{i=1}^{n}(P_M P)^i\|_l = 0,$$

$$P_M K = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}(P_M P)^i = 0.$$

Since the rank of P_M is $n-1$, the rank of K is not greater than 1.
1 is an eigenvalue of P and $\rho(P) \leq 1$. Now, we prove 1 is simple. If not, from the Jordan canonical form, we have a nonsingular matrix T such that

$$P = T\begin{pmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ * & \lambda_2 & 0 & \cdots & 0 \\ 0 & * & \lambda_3 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & * & \lambda_n \end{pmatrix} T^{-1},$$

where $|\lambda_i| \leq 1$ and $\lambda_1 = \lambda_2 = \cdots = \lambda_k = 1$ with $k \geq 2$. Thus

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n} P^i = T\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ * & 1 & 0 & \cdots & 0 \\ * & * & * & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ * & \cdots & * & * & * \end{pmatrix} T^{-1},$$

which implies that the rank of K is larger than 1. This is a contradiction, so 1 is a simple eigenvalue of P. ∎

Lemma 9. *[13] For a stochastic matrix P, the eigenvalue 1 is simple if and only if its associated digraph $\Gamma(P)$ has a spanning tree.*

Lemma 10. *[13] Let $\{P(t)\}$ be a sequence of stochastic matrix with positive diagonal entries, and with associated digraphs denoted by G_t. If for any sequence $\{i_1, i_2, \cdots, i_m\}$, the union of the directed graphs $\{G_{i_1}, \cdots, G_{i_m}\}$ has a spanning tree, then the matrix product $P(i_m) \cdots P(i_2)P(i_1)$ is SIA.*

Proof of Theorem 3:
Sufficiency: By using Lemma 10, and following the same analysis in the proof of (iii)⇒ (i) of Theorem 1, one can get the desired conclusion directly.
Necessity: As we have seen in the above section, the robust consensus of the system (5) can be transformed to the stability of the system (14) on the subspace M. Define

$$P^{(t,r)} = P(t+r-1)\cdots P(t+1)P(t),$$

from Lemma 2, we know that for any $t \geq 0$ there exists a constant integer q such that

$$\|P_M P^{(t,q)}\| < 1. \tag{25}$$

Let $P^{(t)}$ denote $P^{(t,q)}$, $D^{(t)}$ denote the associated directed graph of $P^{(t)}$, G_k denote the neighbor graph associated with $P(k)$, $G^{(t)}$ denote the union of digraphs $\{G_t, G_{t+1}, \cdots, G_{t+q}\}$.
From Lemma 8 and (25), we know 1 is a simple eigenvalue of $P^{(t)}$. Furthermore, the digraph $D^{(t)}$ has a spanning tree according to Lemma 9. To complete the proof of the sufficiency part of Theorem 3, we need to show that $G^{(t)}$ has a spanning tree.
Now, let vertex r be the root of the graph $D^{(t)}$, then for any other vertex $j \in V$, there is a $r \to j$ path in digraph $D^{(t)}$, i.e. there is a sequence of arcs (r, i_1), (i_1, i_2), $(i_2, i_3), \cdots$, (i_{m-2}, i_{m-1}), (i_{m-1}, j) in $D^{(t)}$ connecting r to j. By the relations between $D^{(t)}$ and $P^{(t)}$, we know that the following elements of $P^{(t)}$: $P^{(t)}_{i_1,r}, P^{(t)}_{i_2,i_1}, \cdots, P^{(t)}_{i_{m-1},i_{m-2}}, P^{(t)}_{j,i_{m-1}}$ are all nonzero. Now, we consider each nonzero entry $P^{(t)}_{i_{s+1},i_s}$, $s = 0, 1, \cdots, m-1$(i_0 denotes r, i_m denotes j), by using the property of matrix product, we know that there exists a nonzero item $P_{i_{s+1},k_{q-1}}(t+q-1)P_{k_{q-1},k_{q-2}}(t+q-2)\cdots P_{k_1,i_s}(t)$ for some $k_1, k_2, \cdots, k_{q-1}$. Note that $P_{i,j}(k) \neq 0$ means there is an arc (j,i) in digraph G_k, thus the item $P_{i_{s+1},k_{q-1}}(t+q-1)P_{k_{q-1},k_{q-2}}(t+q-2)\cdots P_{k_1,i_s}(t) \neq 0$ means there is a path in the union digraph $G^{(t)}$ connecting i_s with i_{s+1}. Hence, for each nonzero entry $P^{(t)}_{i_{s+1},i_s}$, $s = 0, 1, \cdots, m-1$, there is a path $i_s \to i_{s+1}$ in digraph $G^{(t)}$, and so it is obvious that there is a path in digraph $G^{(t)}$ connecting r with j. According to the arbitrariness of vertex j, we know that the digraph $G^{(t)}$ has a spanning tree rooted at r, i.e. the union of neighbor digraphs $\{G_t, G_{t+1}, \cdots, G_{t+q-1}\}$ has a spanning tree. The proof of Theorem 3 is complete. ■

5. Simulations

In this section, we consider the consensus behavior of four agents with switching topologies. Three representative examples are given for illustration. In the simulations, the initial states of the agents are chosen randomly from $[0, 3]$, the adjacency matrix $A(t)$ associated with the network topology is $0-1$ matrix, i.e. $a_{ij}(t) = 1$ iff $j \in N_i(t)$ and otherwise $a_{ij}(t) = 0$. We assume each agent treats itself as a neighbor, and can access its own information accurately at any time. That is $x_i(t+1) = x_i(t)$ if agent i has no other neighbors except itself at time t.

Example 1. *Consider a dynamical network of four agents with undirected information exchange topologies. The agent applies the consensus protocol (4). The information exchange topologies are changed as follows: when $t = 3k$, $k = 0, 1, \cdots$, it has the structure as shown in Figure 1.(a), that is $a_{12} = a_{21} = 1$; when $t = 3k + 1$, it is described by Figure 1.(b); when $t = 3k + 2$, it is Figure 1.(c).*

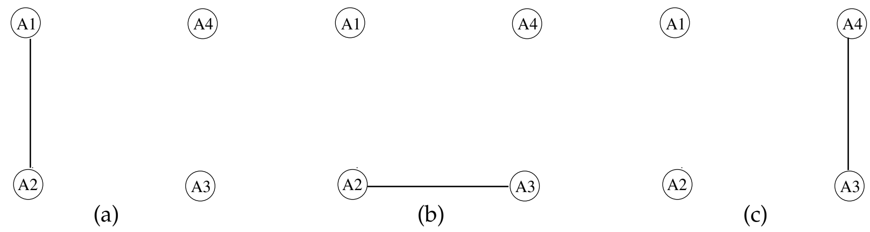


Fig. 1. Three topologies of undirected information exchange.

Figures 2.(a) and (b) show the evolution of protocol (4) with noise chosen randomly from $[-0.5, 0.5]$. Figure 2.(a) depicts the information states $x(i)$, $i = 1, 2, 3, 4$. Figure 2.(b) depicts the maximal difference of information states indexed by $\max_{ij} |x_i - x_j|$. Figure 2.(c) shows the relationship between the difference of the states and the intensity of the noise, where the point represent the mean difference of the states over one running indexed by $\frac{1}{T}\sum_{t=1}^{T} \max_{ij} |x_i(t) - x_j(t)|$ with $T = 50$ in the simulations, and the range of the noise add 0.1 at each running. Figure 2.(c) shows that the higher the noise intensity is, the higher the difference of the states is. From Figure (2), we can see that the system in Example 1 is robust consensus.

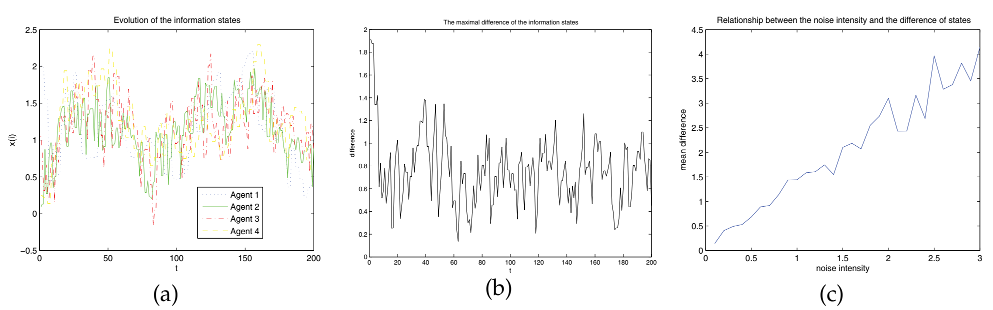


Fig. 2. Evolution of the protocol (4) with the undirected topologies given by Figure 1.

Example 2. *Consider a dynamical network of four agents with directed information exchange topologies as shown in Figure 3. The agent applies the consensus protocol (4). When $t = 3k$, $k = 0, 1, \cdots$, the information exchange topology has the structure as shown in Figure 3.(a), that is $a_{21} = 1$ and others equal zero; when $t = 3k + 1$, it is described by Figure 3.(b); when $t = 3k + 2$, it is Figure 3.(c).*

Figure 4 shows the evolution of the protocol (4) with the above directed topologies. Figures 4.(a),(b) and (c) show the same items as those in Figures 2.(a),(b) and (c). In view of Figure 3, agent 1 does not have other neighbors except itself during the evolution. Thus, in Figure 4.(a), agent 1 keep its own state all the time, and other agents move near it.

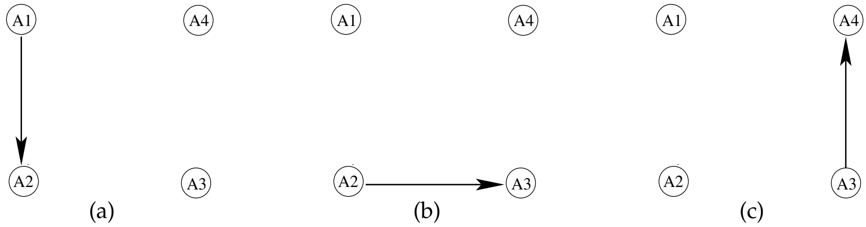


Fig. 3. Three topologies of directed information exchange.

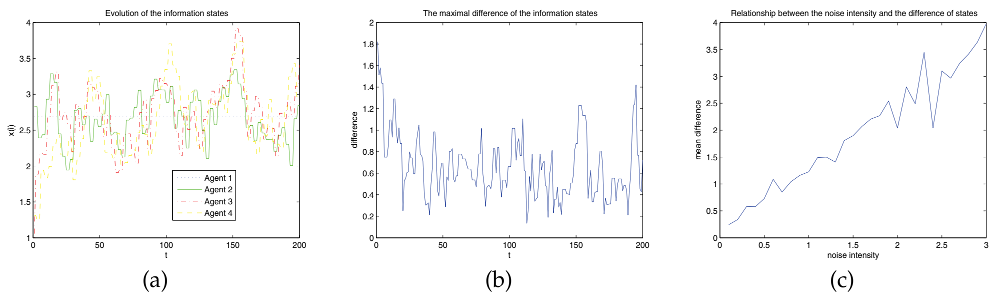


Fig. 4. Evolution of the protocol (4) with the directed topologies given by Figure 3.

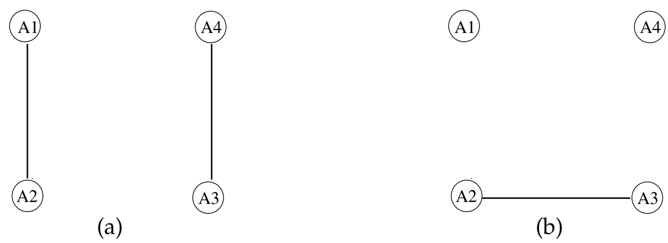


Fig. 5. Two topologies of undirected information exchange.

Example 3. *In this example, we will present two experiments to show that the periodical connectivity of the neighbor graphs is essential for the robust consensus of the agents. In the following experiments, the four agents apply the consensus protocol (4) with the noise chosen randomly from* $[-0.3, 0.3]$, *and the neighbor graphs have the topologies as shown in Figure 5.*

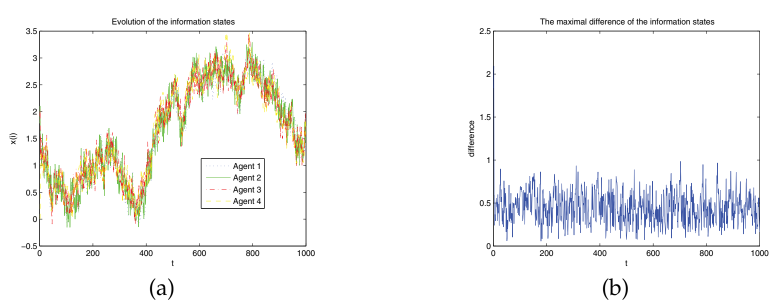
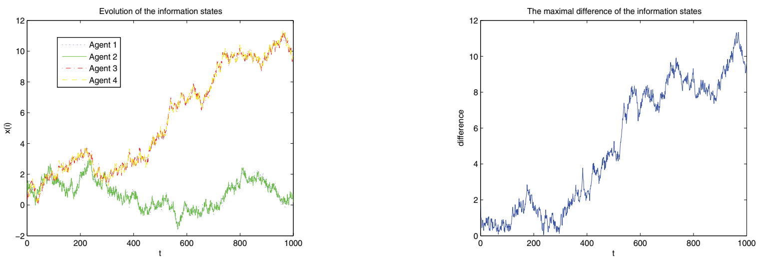


Fig. 6. Evolution with periodically connected information exchange topologies.

In the first experiment, the dynamics of the information exchange topology is defined as follows: when $t = 2k$, $k = 0, 1, \cdots$, the topology has the structure as shown in Figure 5.(a), that is $a_{12} = a_{21} = 1$ and $a_{34} = a_{43} = 1$; when $t = 2k + 1$, it has the structure as shown in Figure 5.(b). Figure 6.(a) shows the information states $x(i)$, and Figure 6.(b) depicts the maximal difference of the information states. From these figures, we can see that the system is robust consensus.



(a) (b)

Fig. 7. Evolution with infinitely connected information exchange topologies.

The second experiment is carried out as follows: when $t = 2^k$, $k = 0, 1, \cdots$, the information exchange topology has the structure as shown in Figure 5.(b), otherwise it has the structure as shown in Figure 5.(a). Therefore, for any time T, the union graph of the neighbor graphs $\{G_t : t \geq T\}$ is connected (called infinitely connected for short). However, from Figure 7 we know that the infinite connectivity is not enough for the robust consensus.

6. Conclusions

This paper studies the robust consensus of multi-agent systems with bounded noises. The information exchange topologies are time-varying. For the case of undirected topologies, the equivalence among the robust consensus, the positivity of the second smallest eigenvalue of the Laplacian matrix, and the periodically joint connectivity of the neighbor graphs are established. For the case of directed topologies, the robust consensus can be achieved if and only if there exists a spanning tree in the union of the neighbor graph periodically. From the theoretical analysis and the simulation results, we can see that the periodical connectivity of the neighbor graphs is essential for the robust consensus of the agents.

7. Appendix

To prove Lemma 2, we need the following theorem.

Theorem 4. *[28] Consider the following time-varying system:*

$$y_{k+1} = B_k y_k, \qquad k \geq 0, \tag{26}$$

where $B_k \in \mathbf{R}^{n \times n}$ and $y_k \in \mathbf{R}^n$. Let $\Psi(k,i)$ be the state transition matrix of system (26): $\Psi(k+1,i) = B_k \Psi(k,i)$, $\Psi(i,i) = I$, $\forall k \geq i \geq 0$. Then system (26) is exponentially stable in the sense that there exist constants $M_0 > 0$ and $c > 0$ such that

$$\|\Psi(k+h,k)\| \leq M_0 \exp\{-ch\} \; \forall k \geq 0, \; \forall h \geq 0,$$

if and only if $\sum_{i=1}^{k} \|\Psi(k,i)\| \leq M_1$, $\forall k \geq 0$, where M_1 is independent of k.

Proof of Lemma 2 Set $\nu = \{\nu_t\}_{t=1}^{\infty}$, $U = \{\nu : \|\nu_t\| \le \delta_0,\ \nu(t) \in M,\ \forall t \ge 1\}$. According to the assumption, we know that there exist $f(\delta_0) > 0,\ T_0 > 0$ such that for any $\nu \in U$ and $\eta_0 = 0$, $\|\eta_t\| \le f(\delta_0),\ t \ge T_0$. By (16), for any $\nu \in U$, we have

$$\sup_{t \ge T_0 - 1} \|\sum_{i=1}^{t} \Phi(t+1, i+1)\nu_i\| \le f(\delta_0) = \varepsilon_0. \tag{27}$$

We want to prove that there exists a constant $N_1 > 0$ such that for any $\nu \in U$

$$\sup_{t \ge T_0 - 1} \sum_{i=1}^{t} \|\Phi(t+1, i+1)\nu_i\| \le N_1. \tag{28}$$

We will prove this by reduction to absurdity. Assume that there exist $t_0 \ge T_0 - 1$ and $u \in U$ such that $\sum_{i=1}^{t_0} \|\Phi(t_0+1, i+1)u_i\| \ge 2n\varepsilon_0$, where n is the dimension of space. Let $\Gamma_i(s,j)$ be the (s,j) entry of matrix $\Phi(t_0, i+1)$, and let $u_i(j)$ be the jth element of u_i. Then there must exist an integer $s \in [1, n]$ such that

$$\sum_{i=1}^{t_0} |\sum_{j=1}^{n} \Gamma_i(s,j)u_i(j)| \ge 2\varepsilon_0. \tag{29}$$

Define $\hat{u}_i(j) = |u_i(j)| sgn\{\Gamma_i(s,j)\}, 1 \le j \le n$, where $sgn(\cdot)$ is the symbolic function. Obviously $\sup_{t \ge 0} \|\hat{u}_t\| = \sup_{t \ge 0} \|u(t)\| \le \delta_0$. By (29), we have

$$\|\sum_{i=1}^{t_0} \Phi(t_0+1, i+1)\hat{u}_i\| \ge |\sum_{i=1}^{t_0}\sum_{j=1}^{n} \Gamma_i(s,j)\hat{u}_i(j)| = \sum_{i=1}^{t_0}\sum_{j=1}^{n} |\Gamma_i(s,j)||u_i(j)| \ge 2\varepsilon_0. \tag{30}$$

Decompose $\{\hat{u}_i,\ i = 1, \cdots, t_0\}$ into two orthogonal vectors α_i and β_i, such that $\hat{u}_i = \alpha_i + \beta_i,\ \alpha_i \in M,\ \beta_i \in X$. Then $\|\hat{u}_i\| = \|\alpha_i\| + \|\beta_i\|$; thus $\|\alpha_i\| \le \|\hat{u}_i\|$. Noting that $P(t)$ is stochastic, we have $P_M P(t)\beta_i = 0,\ \forall t \ge 0,\ \forall i \in [1, t_0]$. Thus by (30), we have

$$\|\sum_{i=1}^{t_0} \Phi(t_0+1, i+1)\alpha_i\| = \|\sum_{i=1}^{t_0} \Phi(t_0+1, i+1)(\hat{u}_i - \beta_i)\| = \|\sum_{i=1}^{t_0} \Phi(t_0+1, i+1)\hat{u}_i\| \ge 2\varepsilon_0. \tag{31}$$

Now define a new sequence as

$$u^* = \begin{cases} \alpha_i, & i \in [1, t_0] \\ u_i, & i > t_0 \end{cases}$$

Obviously $u^* \in U$, and by (31), we have

$$\sup_{t \ge T_0 - 1} \|\sum_{i=1}^{t} \Phi(t+1, i+1)u_i^*\| \ge \|\sum_{i=1}^{t_0} \Phi(t_0+1, i+1)\alpha_i\| \ge 2\varepsilon_0.$$

It is opposite to (27). Thus (28) holds.
For any vector $\xi \in \mathbf{R}^n$ with $\|\xi\| \le \delta_0$, do orthogonal decomposition on ξ: $\xi = \xi_1 + \xi_2,\ \xi_1 \in M,\ \xi_2 \in X$. Then $\|\xi\| = \|\xi_1\| + \|\xi_2\|,\ \|\xi_1\| \le \delta_0$, and $P_M P(t)\xi_2 = 0,\ \forall t \ge 0$. By (28), we have

$$\sup_{t \ge T_0 - 1} \sum_{i=1}^{t} \|\Phi(t+1, i+1)\xi\| = \sup_{t \ge T_0 - 1} \sum_{i=1}^{t} \|\Phi(t+1, i+1)\xi_1\| \le N_1.$$

On account of the arbitrariness of ξ, there must exist a constant $N_2 > 0$ such that

$$\sup_{t \geq T_0 - 1} \sum_{i=1}^{t} \|\Phi(t+1, i+1)\| \leq N_2. \tag{32}$$

Because $\{P(t)\}_{t=1}^{t=T_0}$, P_M and T_0 are bounded, we can choose a constant $N_3 > 0$ such that

$$\sup_{t \leq T_0 - 1} \sum_{i=1}^{t} \|\Phi(t+1, i+1)\| \leq N_3. \tag{33}$$

Taking $\bar{N} = \max\{N_2, N_3\}$ which is independent of t, by (32)(33), we have $\sum_{i=1}^{t} \|\Phi(t,i)\| \leq \bar{N}$, $\forall t \geq 0$. According to Theorem 4, there exist constants $N > 0$ and $\lambda \in (0,1)$ such that $\| \Phi(k+h,k) \| \leq N\lambda^h$, $\forall k \geq 0$, $\forall h \geq 1$. ■

Proof of Lemma 3 According to the assumptions of this lemma, we can choose an integer $q > 0$ such that

$$\|\Phi(k+q,k)\| = \|P_M \prod_{t=k}^{k+q-1} P(t)\| \leq \frac{1}{2}. \tag{34}$$

Let $\rho_k = \lambda_1(\sum_{k}^{k+q-1} L(t))$ and x_k^* be the corresponding unit eigenvector. Then we have $\rho_k = \sum_{k}^{k+q-1} x_k^{*\tau} L(t) x_k^*$. Since $L(t) \cdot \mathbf{1} = 0$ for all $t \geq 1$, x_k^* can be chosen from the subspace M.

For any integers $t_j \in [k, k+q-1]$, $j = 1, \cdots, s \leq q$, we have

$$\begin{aligned} & x_k^{*\tau} P_M D^{-1}(t_1)L(t_1)\cdots D^{-1}(t_s)L(t_s)x_k^* \\ \leq\ & \|x_k^{*\tau} P_M D^{-1}(t_1)L(t_1)\cdots D^{-1}(t_s)L(t_s)^{\frac{1}{2}}\| \cdot \|L(t_s)^{\frac{1}{2}} x_k^*\| \\ \leq\ & \beta^{2q}(x_k^{*\tau} L(t_s)x_k^*)^{\frac{1}{2}} \leq \beta^{2q}\rho_k^{\frac{1}{2}}. \end{aligned} \tag{35}$$

Let b be the uniform upper bound of $\| L(t) \|$ and $\| D^{-1}(t) \|$ in view of Assumption Λ. From (34),(35) and the Schwarz inequality, it follows that

$$\begin{aligned} \frac{1}{2} \ \geq\ & \|\Phi(k+q,k)\| = \|P_M \prod_{t=k}^{k+q-1} P(t)\| = \|P_M \prod_{t=k}^{t=k+q-1} (I - D^{-1}(t)L(t))\| \\ \geq\ & x_k^{*\tau} P_M \prod_{t=k}^{t=k+q-1} (I - D^{-1}(t)L(t))x_k^* \\ =\ & 1 - \sum_{s=1}^{q} \sum_{k \leq t_1 \leq \cdots \leq t_s \leq k+q-1} x_k^{*\tau} P_M D^{-1}(t_1)L(t_1)\cdots D^{-1}(t_s)L(t_s)x_k^* \\ \geq\ & 1 - \sum_{s=1}^{q} \binom{q}{s} \beta^{2q}\rho_k^{\frac{1}{2}}. \end{aligned}$$

So, $\rho_k^{\frac{1}{2}} \geq \dfrac{1}{2\sum_{s=1}^{q} \binom{q}{s}\beta^{2q}}$, $\forall k \geq 0$, which means that $\inf_{k \geq 0} \lambda_1(\sum_{k}^{k+q-1} L(t)) \neq 0$. Thus Lemma 3 is true. ■

Proof of Lemma 4 On the one hand, from $\min_i z_i \le \frac{\sum_i z_i}{n} \le \max_i z_i$, we have, for any z_j,

$$z_j - \max_i z_i \le z_j - \frac{\sum_i z_i}{n} \le z_j - \min_i z_i.$$

So, $|z_j - \frac{\sum_i z_i}{n}| \le \max\{\max_i z_i - z_j, z_j - \min_i z_i\} \le \Delta z$. Thus

$$\|P_M z\| = \|P_M(z - \frac{\mathbf{1}^\tau z \mathbf{1}}{n})\| = \|z - \frac{\mathbf{1}^\tau z \mathbf{1}}{n}\| \le \sqrt{n}\Delta z.$$

On the other hand,

$$\begin{aligned} |z_i - z_j| &= |(z_i - \frac{\sum_s z_s}{n}) - (z_j - \frac{\sum_s z_s}{n})| \le |z_i - \frac{\sum_s z_s}{n}| + |z_j - \frac{\sum_s z_s}{n}| \\ &\le \sqrt{2}\sqrt{(z_i - \frac{\sum_s z_s}{n})^2 + (z_j - \frac{\sum_s z_s}{n})^2} \le \sqrt{2}\|z - \frac{\mathbf{1}^\tau z \mathbf{1}}{n}\| = \sqrt{2}\|P_M z\|, \end{aligned} \tag{36}$$

Thus, $\Delta z \le \sqrt{2}\|P_M z\|$. From (9) and the above analysis, it follows that

$$\frac{\sqrt{2}}{2}\Delta z \le \|P_M z\| = d(z, X) \le \sqrt{n}\Delta z.$$

■

8. Acknowledgements

This work was supported by the NSF of P. R. China under Grant Nos. 60804043, and a grant from the Major State Basic Research Development Program of China (973 Program) (No. 2010CB731400).

9. References

[1] De Groot, M.H. (1974). Reaching a consensus, *J. Amer. Statist. Assoc.*, Vol.69, 118-121
[2] Chatterjee, & S., Seneta, E. (1977). Towards consensus: some convergence theorems on repeated averaging, *J. Appl. Prob*, Vol.14, 89-97
[3] Fischer M. J. (1983). The consensus problem in unreliable distributed systems (a brief survey), In *Foundations of Computation Theory*, M. Karpinsky ed., Lecture Notes in Comput, 127-140, Springer-Verlag, New York
[4] Lynch, N. (1996). *Distributed Algorithms*, Morgan Kaufmann, San Mateo, CA
[5] Vicsek, T., Czirok, A., Ben Jacob, I., Cohen, I., & Schochet, O. (1995). Novel type of phase transitions in a system of self-driven particles, *Phys. Rev. Lett.*, Vol.75, 1226-1229
[6] Visscher, P. K. (2003). How self-organization evolves, *Nature*, Vol.421, 799-800
[7] Shamma, J. (2008). *Cooperative Control of Distributed Multi-Agent Systems*, Wiley-Interscience, New York
[8] Ren, W., & Beard, R.W. (2008). *Distributed Consensus in Multi-vehicle Cooperative Control*, Communications and Control Engineering Series, Springer-Verlag, London
[9] Jadbabaie, A., Lin, J., & Morse, A.S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules, *IEEE Trans. Auto. Contr.*, Vol.48, No.6, 988-1001

[10] Liu, Z.X., Han, J., & Guo, L. (2006). Analysis and Regulation of Dynamical Networks Based on Vicsek's Model, *Complex Networks*, Guo, L. & Xu, X.X.(Ed.), 49-66, Shanghai Scientific & Technological Education press, Shanghai

[11] Ma, H. B. (2009). Decentralized Adaptive Synchronization of a Stochastic Discrete-Time Multiagent Dynamic Model, *SIAM J. Control Optim.*, Vol. 48, 859-880

[12] Cao, M., Morse, A.S., & Anderson, B.D.O. (2008). Reaching a Consensus in a dynamically changing environment: a graphical approach, *SIAM Journal on Control and Optimization*, Vol.47, 575-600

[13] Ren, W., & Beard, R. W. (2005). Consensus seeking in multi-agent systems under dynamically changing interaction topologies, *IEEE Trans. Auto. Contr.*, Vol.50, No.5, 655-661

[14] Moreau, L.(2005). Stability of multi-agent systems with time-dependent communication links, *IEEE Trans. Auto. Contr.*, Vol.50, No.2, 169-182

[15] Hong, Y., Gao, L., Cheng, D., & Hu, J. (2007). Lyapunov-based approach to multi-agent systems with switching jointly-connected interconnection, *IEEE Trans. Auto. Contr.*, vol.52, 943-948

[16] Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: algorithms and theory, *IEEE Trans. Auto. Contr.*, Vol.51, 401-420

[17] Su, H., Wang, X., & Lin, Z. (2009). Flocking of multi-agents with a virtual leader, *IEEE Trans. Auto. Contr.*, Vol.54, 293-307

[18] Tahbaz-Salehi, A., & Jadbabaie, A. (2008). A necessary and sufficient condition for consensus over random networks, *IEEE Trans. Auto. Contr.*, Vol.53, No.3, 791-795

[19] Hatano, Y., & Mesbahi, M. (2005). Agreement over random network, *IEEE Trans. Auto. Contr.*, Vol.50, No.11, 1867-1872

[20] Olfati-Saber, R., & Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays, *IEEE Trans. Auto. Contr.*, Vol.49, 1520–1533

[21] Tian, Y., & Liu, C. (2008). Consensus of multi-agent systems with diverse input and communication delays, *IEEE Trans. Auto. Contr.*, Vol.53, No.9, 2122-2128

[22] Xiao, L., Boyd, S., & Kim, S.J. (2007). Distributed average consensus with least-mean square deviation, *Journal of Paralell and Distributed Computing*, Vol.67, No.1, 33-46

[23] Huang, M., & Manton, J. H. (2009). Coordination and consensus of networked agents with noisy measurement: stochastic algorithms and asymptotic behavior, *SIAM Journal on Control and Optimization*, Vol.48, No.1, 134-161

[24] Li, T., & Zhang, J.F. (2010). Consensus conditions of multi-agent systems with time-varying topologies and stochastic communication noises, *IEEE Trans. Auto. Contr.*, Vol.55, No.9, 2043-2057

[25] Wang, L., & Liu, Z.X. (2009). Robust consensus of multi-agent systems with noise, *Science in China*, Vol.52, No.5, 824-834

[26] Wang, L., & Guo, L. (2008). Robust consensus and soft control of multi-agent systems with noises, *Journal of Systems Science and Complexity*, Vol.21, 406-415

[27] Wolfowitz, J. (1963). Products of indecomposable, aperiodic, stochastic matrices, *Proceedings of the American Mathematical Society*, Vol.15, 733-737

[28] Guo, L. (1993). *Time-Varying Stochastic Systems*, Jilin Science and Technology Press, Jilin

[29] Seneta, E. (1984). *Non-negative Matrices and Markov Chains*(2nd ed.), Springer, New York

[30] Godsil, C., & Royle, G. (2001). *Algebraic Graph Theory*, Springer-Verlag, New York

[31] Horn, R. A., & Johnson, C. R. (1985). *Matrix Analysis*, Cambridge University, New York

[32] Doob, J. L. (1953). *Stochastic Processes*, John Wiley, New York

Part 3

Multi-Agent Systems Programming

16

Principles of Agent-Oriented Programming

André Filipe de Moraes Batista, Maria das Graças Bruno Marietto, Wagner Tanaka Botelho, Guiou Kobayashi, Brunno dos Passos Alves, Sidney de Castro and Terry Lima Ruas
Centre of Mathematics, Computation and Cognition, Federal University of ABC (UFABC) Brazil

1. Introduction

In the early of 1970s the Artificial Intelligence (AI) was defined as a system based on the Von Neumann model (with a single control center) and the concepts of traditional Psychology. In the late of 1970s the idea of individual behavior was tested from several works that required a distributed control. For example, works related to blackboards (Fennel & Lesser, 1977) and actors (Hewitt, 1977) allowed the modeling of classical problems considering concepts such as cooperation, communication and distribution. Therefore, the researchers started to investigate the interaction between systems, trying to solve distributed problems in a more social perspective.

In order to find solutions for distributed systems, the Distributed Artificial Intelligence (DAI) started in the early of 1980s to be investigated. It combines the theoretical and practical concepts of AI and Distributed Systems (DS). The solution is also based on social behaviors where the cooperative behavior is utilized to solve a problem. The DAI is different of DS because (i) it is not based on the client-server model, and (ii) the DAI area does not address issues related to distributed processing, aiming to increase the efficiency of computation itself (transmission rate, bandwidth, etc). However it aims to develop technical cooperation between entities involved in a system. Also, the DAI differs from AI because it brings a new and broader perspectives on knowledge representation, planning, problem solving, coordination, communication, negotiation, etc.

The Multi-Agent Systems (MAS) is one of the research areas of DAI, and uses autonomous agents with their own actions and behaviors. The agents in a MAS are designed to act as experts in a particular area. The main characteristic is to control their own behaviors and, if necessary, to act without any intervention of humans or other systems. The focus of the designer is to develop agents working in an autonomous or social way, as well as systems of communication and cooperation/collaboration, so that the solution arises from the interactions. This bottom-up approach usually leads to an open architecture, where agents can be inserted, deleted, and reused. According to Sawyer (2003), the Internet is an example of MAS because it is constituted by thousands of independent computers, each on running autonomous software programs that are capable of communication with a program running on any other node in the network.

The term agent is used frequently in AI, but also outside its field, for example in connection with databases and manufacturing automation. When people in AI use the term, they are referring to an entity that functions continuously and autonomously in an environment where other processes take place and other agents exist. The sense of autonomy is not precise, but the term is taken to mean that the agent activities do not requires constant human guidance or intervention (Shoham, 1993). There are a number of good reasons for supposing that agent technology will enhance the ability of software engineers to construct complex and distributed applications. It is a powerful and natural metaphor to conceptualize, design and implement many systems.

This chapter makes an overview of theoretical and technical concepts of Agent-Oriented Programming (AOP). Historically, the AOP appears after the Object-Oriented Programming (OOP). However the differences between them are not clear in the research and development community. Section 2 discusses the differences between objects and agents and also the evolution of the programming language paradigms. Section 3 presents the micro and the macro levels of a society of agents. The pitfalls of AOP are explained in Section 4. Two multi-agent systems platforms are presented in Section 5. In Section 6 two multi-agent applications are presented: a cognitive modeling of stock exchange and a military application of real-time tactical information management. Section 7 ends up the conclusions.

2. Programming with objects or agents: What are the main differences?

Procedural programs are typically intended to be executed discretely in a batch mode with a specific start and end (Huhns, 2004). However, the modular programming approach employs smaller units of code that could be reused under a variety of situations. The structured loops and subroutines are designed to have a high degree of local integrity (Odell, 1999). The concept of objects and agents are the key to understand the OOP and AOP, respectively.

2.1 Agents versus objects

In (Silva et al., 2003) an agent is defined as an extension of an object with additional features, because it extends the definition of state and behavior associated with objects. The mental states consist of its states and behaviors. The beliefs and goals, plans, actions are equivalent to the object's state and agent's behaviors, respectively. Moreover, the behavior of an agent extends the behavior of objects because the agents have freedom to control and change their behaviors. They also not require external stimuli to carry out their jobs. These make the agents active elements and objects passive ones.

Agents use some degree of unpredictable behavior. For example, the ants appear to be taking a random walk when they are trying to find food. Their behavior starts to become predictable when the pheromones or food are detected. Therefore, an agent can range from being totally predictable to completely unpredictable. On the other hand, the objects do not have to be completely predictable (Odell, 2002).

The agent has the ability to communicate with the environment and other entities. In MAS the agents are autonomous and has the characteristics to interact with the environments and other agents. However, the object messages have the most basic form of interaction. Also, it request via message only one operation formatted in a very exacting way. The oriented-object message broker has the job of matching each message to exactly one method invocation for exactly one object.

In the communication between agents in the MAS is allowed to use the method invocation of OOP. However, the demand of messages are greater than those used by objects technology. An agent message could consist of a character string whose form can vary, yet obeys a formal syntax, while the conventional object-oriented method must contain parameters whose number and sequence are fixed. Agents may engage in multiple transactions concurrently, through the use of multiple threads or similar mechanisms. Conventional OOP have difficulty to support such requirements (Odell, 2002). But it is possible for the agents to employ objects for situations that require a little autonomous or interactive ability. In the MAS environment, an Agent Communication Language (ACL) is necessary to send a message to any agent. KQML (Group et al., 1992) and FIPA ACL (Buckle & Hadingham, 2000) are examples of the ACLs.

2.2 Object-Oriented Programming

Rentsch (Rentsch, 1982) predicted in 1982 that the OOP would be in 1980s what the structured programming was in the 1970s. The earliest precursor of MAS was OOP. The subsequent OOP evolution provide methods and techniques to identify the objects and their attributes needed to implement the software, to describe the associations between the identified objects, to define the behavior of the objects by describing the function implementations of each object, to refine objects and organize classes by using inheritance to share common structure are the challenges of object-oriented analysis and design (Wahono, 2001). In OOP, an object is a single computational process maintaining its own data structures and procedures (Sawyer, 2003). Also, it maintains the segments of code (methods) and gain local control over the variables manipulated by its methods. In traditional OOP, the objects are passive because their methods are invoked only when some external entity sends them a message. The basic element used in the OOP is the class. A class definition specifies the class variables of an object and the methods that the object accepts. One class inherits from another class such that the new class is an extension of the existing class, instances of two classes collaborates with each other by exchanging messages (Lind, 2000).

In Wahono (2001), the object is defined as the principal building blocks of OOP. Each object is a programming unit consisting of attributes (instance variables) and behaviors (instance methods). An object is a software bundle of variables and related methods. It is easy to see many examples of real-world objects. Also, it is possible to represent the real-world objects using software objects. For example, bicycles have attributes (gear, pedal cadence, two wheels) and behaviors (braking, accelerating, slowing down). A software object that modeled our real-world bicycle would have variables that indicate the bicycle's current attribute: its speed is 10 mph, its pedal cadence is 90 rpm, and its current gear is the 5^{th} gear. These variables and methods are formally known as instance variables and instance methods to distinguish them from class variables and class methods.

2.3 Agent-Oriented Programming

The MAS is considered as an object-oriented system that is associated to an intelligent meta-system. By this way, an agent is viewed as an object that has a layer of intelligence, comprising a number of capabilities such as uniform communication protocol, perception, reaction and deliberation, all of them not inherent to objects. However, the AOP has code, states and agent invocations. The agents also have individual rules and goals to make them appear like active objects with initiative. In AOP the class is replaced by role, state variable with belief/knowledge and method with message. The role definitions describe the agent capability and the information needed to desired results.

In order to the agents act with intelligence in their environment, the idea is to develop the complex entities and provide the agents with the knowledge and beliefs to be able to achieve their desires.

Framework	OOP	AOP
Basic unit	object	agent
Parameters defining state of basic unit	unconstrained	beliefs, commitments, capabilities, choices, ...
Process of computation	message passing and response methods	message passing and response methods
Types of message	unconstrained	inform, request, offer, promise, decline, ...
Constraints on methods	none	honesty, consistency, ...

Table 1. Relation between OOP and AOP (Shoham, 1993).

2.4 Differences between Object-Oriented Programming and Agent-Oriented Programming

Table 1 summarizes the major features of the relation between OOP and AOP. In short, AOP is seen as an extension of OOP. On the other hand, OOP can be viewed as a successor of structured programming. Wagner (2003) defines two main characteristics about the AOP. First, while the state of an object in OOP has no generic structure, the state of an agent in AOP consists of mental components such as beliefs and commitments. Second, while messages in OOP are code in an application-specific ad-hoc manner, a message in AOP is coded as a speech act according to the standard Agent Communication Language that is application-independent.

The autonomy and interaction are the key areas to differentiate the AOP from OOP. The following list describes some underlying concepts that agent-based systems employ (Odell, 2002):

- Decentralization: the objects are centrally organized, because the objects methods are invoked under the control of other components in the system. On the other hand, the agent has a centralized and decentralized processing;
- Multiple and dynamic classification: in OOP, objects are created by a class and, once created, may never change their class or become instances of multiple classes (except by inheritance). However, the agents provide a more flexible approach;
- Small in impact: the objects and agents can be described as small grained or large grained. Also, in comparison with the whole system the agent or object can be small. In an agent-based supply chain, if a supplier or a buyer is lost, the collective dynamics can still dominate. If an object is lost in a system, an exception is raised;
- Emergence: the ant colonies have emergent qualities where groups of agents behave as a single entity. Each consists of individual agents acting according to their own rules and even cooperating to some extent. In MAS, simple rules produce emergence. Since traditional objects do not interact without a higher-level thread of control, emergence does not usually occur. As more agents become decentralized, their interaction is subject to emergence.

3. Agents and agency in the Agent-Oriented Programming

3.1 Micro level

The micro level refers to the agent itself. The agent definition, reactive and cognitive agents and their architectures are considered in this level.

3.1.1 The agent definition

The term agent is increasingly used in diverse ways in DAI. However, it has become meaningless without reference to a particular notion of agenthood. Some notions are primarily intuitive, others quite formal. It is also called a software agent or an intelligent agent. The words intelligent and agent describe some of its characteristic features. Intelligent is used because the software has certain types of behavior. The intelligent behavior is the selection of actions based on knowledge and the agent tells something about the purpose of the software. When the researchers in DAI think about the term, they refer to an entity that functions continuously and autonomously in an environment in which other processes take place and other agents exist. The sense of autonomy is not precise, but the term is taken to mean that the agents' activities do not require constant human guidance or intervention (Shoham, 1993).

In the literature we find many discussions about what constitutes an agent. In Shoham (1993), an agent is an entity whose state is viewed as consisting of mental components such as beliefs, capabilities, choices, and commitments. Also, there are several approaches utilized to define agents. (Wagner, 2003) uses the software engineering and the mentalist approaches. In the first approach, it emphasizes the significance of application-independent high-level agent-to-agent communication as a basis for general software interoperability. For example, an entity is a software agent if and only if it communicates correctly in an agent communication language. In the second, the approach is based on the knowledge representation paradigm of AI, points out that the state of an agent consists of mental components such as beliefs, perceptions, memory, commitments, expectations, goals and intentions. Its behavior is the result of the concurrent operation of its perception system, its knowledge system and its action system.

According to Wooldridge & Jennings (1995), it is difficult to define an universal accepted definition of the term agent. But the autonomy is the central idea of the concept of an agent. The authors also explained that the difficulty to find the definition is because of the term is widely used by many researchers working in closely related areas. Therefore, they define two general usages of the term: the weak and stronger notions of agency. The weak notion considers a set of properties that a software and hardware represent to be considered an agent. The following properties are defined:

- Autonomy: the agents are able to decide their actions without the direct intervention of humans and others;
- Social Ability: the agents communicate using some kind of agent-communication language with other agents (humans or computational) in order to solve a problem;
- Reactivity: the agents perceive their environments (which may be the physical world, a user via a graphical user interface, other agents, etc) and respond to changes that occurs in them;
- Pro-activeness: the agents has initiative, they do not act only in response to their environment.

In the strong notion the weak notions are preserved, and in addition other properties are considered. These properties are more applied to human's characteristics, such as knowledge, belief, intention, obligation, emotion, anthropomorphism and etc.

Since the beginning of the DAI area, at least a consensus can be perceived in the scientific community: the division of the agents in reactive and cognitive ones. This binary view allows to focus the analysis on the key points of each of these classes. In the following sections these points are highlighted.

3.1.2 Reactive agents

Reactive agents are usually modeled by following the metaphor of biological and ethological organizations such as: hives, anthill, populations of insects, bacteria, antibodies, etc. Such systems provide evidence of emergent intelligence. Following this metaphor, reactive agents tend to be structurally simpler since they do not have an explicit representation of its environment, they are also not capable to perform sophisticated logic reasoning. Their behaviors are based only on the stimulus-response.

In a society of reactive agents the communication takes place indirectly, through the external environment. Also, their decisions are concerned on the current situation since no history actions are stored. Generally, these agents do not plan their future actions and also they do not communicate with other agents. Normally, they know the other agents actions by the changing of the environment. Thus, their intelligent behaviors are obtained through the interaction with the environment and other agents.

In Gottifredi et al. (2010), the reactive architecture is proposed for the implementation of a robot soccer team. In the reactive layer the basic actions are implemented. It includes the basic hardware and software supports that are provided by the league. This involves physical support, such as infrared transmitters, video camera, communication network, and common software.

3.1.3 Cognitive agents

Cognitive agents are inspired by human social organization, groups and hierarchies. They have explicit models of the external world and memory structures that allow the agents to have the history of past actions used to solve the current problems (Bittencourt, 2006). Also, they communicate with each other directly using their perceptual systems (to sense the environment) and the communication system (exchange messages).

The main characteristics about the cognitive agents are: (a) the societies have a few agents; (b) the communications are made directly; (c) they have explicit models of the external world, structures of memory which keep the history of the past actions and have the ability to make predictions about the future; (d) they are able to decide about their intentions and knowledge, create and execute their actions plans; (e) They have a cooperation and coordination system.

The studies of cognitive science and DAI techniques allow the agents to add the ability of thinking and learning. In fact, cognitive agents have a certain computational complexity and are characterized by an intelligent behavior in the agency or separately. There are several theories utilized to model the cognitive agents such as decisions theory (Parmigiani et al., 1997) and intentional systems (Dennett, 1987). It is possible to achieve the solution of many problems using knowledge and thinking based on the information that represents aspects of their world.

3.1.3.1 Intentional Systems

It is vital the development and application of theoretical models to structure and help reasoning about the individual and social agents´ behaviors. Among such theoretical models we cite the theory of intentional systems proposed in (Dennett, 1987). The philosopher Daniel Dennett (Dennett, 1987) uses the term intentional systems to describe systems that can be described and/or predicted through mental attributes such as beliefs, preferences, desires, intentions, free will, goals, etc. These attributes are called by (Wooldridge & Jennings, 1995) as intentional notions.

Therefore, what characterizes the intentional system is the possibility to be interpreted as a system with its intentional notions. The desires specify the preferences regarding to the future states of the world. They can be inconsistent and not reachable. The goals are the desire that an agent consider achievable in a certain moment. The intention has an associated commitment that directs and controls the future activities of the agent, so that it achieves its goals. The beliefs are the expression of the states of the world seen by the agent. Moreover, they also make the vision of how the world will change if the agent performs some actions.
The cognitive agents can be modeled as intentional systems with mental attributes that influence their actions. The differences in the model depend on the specification of the mental states used to describe the behaviors of each agent.

3.1.3.2 BDI Architecture

The architecture specifies the structure and the behavior of an agent. The BDI (Belief, Desire and Intention) model is probably the most known cognitive agent architecture. It was proposed by (Bratman, 1987) as a theory of human practical reasoning. Thus, the BDI are the mentalists attributes to define the agent state. The agents beliefs and goals correspond to the information that the agent has about the world and the intuitively correspond to the tasks allocated, respectively. (Rao & Georgeff, 1995) have adapted the model of Bratman changing to the formal theory and in a software agent model based on beliefs, goals and plans. The model is a BDI interpreter used as an inspiration for the BDIs systems utilized until now (de Nunes, 2007). In order to design an agent based on the BDI model, the beliefs and desires need to be specified. The agent is allowed to choice the intentions attributes, based on a self-analysis of the states initially available.
The BDIs applications emerge as a solution to various problems. For instance, Unmanned Aerial Vehicles (UVA) (Reichel et al., 2008), Real-time Scheduling (Paulussen et al., 2004) and others. In Section 6.1.2 we propose an architecture based on the general BDI and dMars models presented in (Wooldridge, 1999) and (d'Inverno et al., 1998b), respectively. The BDI dMars architecture contains four keys data structures: beliefs, goals, intentions and a plan library. The agents beliefs and goals correspond to the information that the agent has about the world and the intuitively correspond to the tasks allocated to it, respectively. Agents must choose some subset of available desires (i.e. intentions) and commit resources to achieve them. These chosen desires are intentions. Each agent has a plan library, which is a set of plans, or recipes, specifying courses of action that may be undertaken by an agent in order to achieve its intentions (d'Inverno et al., 1998b). Additionally to the dMars architecture, the Wooldridge generic BDI architecture has useful attributes (Wooldridge, 1999). For example, the filter, generators and revisions functions that makes the deliberation information.

3.2 Macro level

At the macro level, we investigate the agent communication languages, protocols communication between agents, coordination mechanism and negotiation.

3.2.1 Agent communication languages

The communication between members in any society is very important. It is not different in the society of agents that communicate between them to achieve their goals. The communication is a natural way to have interaction, cooperation and negotiation between agents in MASs. It is important for the agent to have the ability to perceive (receive messages)

and to act (send messages) in an effectively and efficiently ways in their environment. For that purpose, they need a shared language and communication protocol.
Agents can process and refer to the same object differently. Therefore, the language structure is needed to allow the integration of these representations. The Speech Acts is a linguistic approach utilized to model the language adaptivity (Marietto, 2000). The philosophers John Austin (Austin, 1962) and John Searle (Searle, 1969) developed the Speech Acts theory, that views human natural language as actions such as requests, suggestions, commitments, and replies (Huhns & Stephens, 1999). These philosophers contribute to the studies related to the development of the agent languages and the communication protocols. John Austin noticed that certain expressions were like "physical actions" that seemed to modify the state of the world. The theory considers three aspects (Huhns & Stephens, 1999):

- Locution: the physical utterance by the speaker. For example: "Can you make a coffee?"
- Illocution: intended by the utterance. For example: "He asked me to make coffee."
- Perlocution: the action that results from the locution. For example: "He made me make coffee."

[Albuquerque e Tedesco, 2004] pointed out that a communication language must have a predictable perlocutionary act with a locutionary act to make it possible to know what is the issuance perlocutionary act. In other words, the language must provide a mechanism where an agent knows the possible responses that another agent will answer in relation to the sender's message, predicting the possible reactions of the receiver.
The illocutionary concept is utilized to define the type of message in the Speech Scts theory. The intentions of the sender communication act are clearly defined, and the receiver has no doubt about the type of the sent message (Huhns & Stephens, 1999). The speech acts are referred as "performatives" in the context of agent communication. It is utilized to identify the illocutionary force of the special class of utterance. The verbs promise, report, convince, insist, tell, request, and demand are examples of performatives.
Based on the Speech Acts theory, it is possible to represent the interaction between agents as an exchange of knowledge. In the communication between agents, the messages have an intention (illocutionary act) and content (locutionary act). For instance, the message TELL(A, B, P) means that the agent A is telling to the agent B that he believes in P. The intention is to inform, and its content is P (Coutinho et al., 2009).

3.2.2 Communication protocols between agents

Protocols are defined as a set of rules to support the network communication. The protocols control the format, the content and the meaning of the sent and received messages. However, they are not restricted only in the communication tasks, but also to assist the negotiation process. They also guarantee the interactions between other agents (Marietto, 2000). In other words, when the agents have conflicting goals or are simple self-interested, the objective of the protocols is to maximize the payoffs of the agents (Huhns & Stephens, 1999). Three protocols are described in (Marietto, 2000):

- Announcement Protocols: The agent informs other agents in the society the services that are able to offer. The agent who is offering the service sends an advertisement (performative) with the :CONTENT (parameter) specifying the actions that is possible to perform. It receives a positive response message if the :CONTENT is validated. Otherwise, it receives a message of rejection.

- Ask-about Protocols: This protocol defines the sequence of two messages, allowing an agent to ask another agent. For example, a general question about the knowledge base of another agent, a request to identify an agent or a request to evaluate an affirmation.
- Task/Action Agreement Protocols: This protocol defines a sequence of messages, allowing an agent to engage a task of another agent.

3.2.3 Coordination

In sections 3.2.2 and 3.2.1 we analyzed how the agent shared the same environment and communicate with each other. However, in order to work together in a harmonious manner, it is necessary more than the ability to communicate with each other. In DAI systems the agents cooperate to achieve their local goals and the goals of the society they belong . Also, they perform a range of useful activities, have their own aims, goals and communicate with other agents (Jennings, 1995). (Reis, 2003) defines coordination as "the act of working together harmoniously to achieve an agreement or common purpose".
The coordination should be able to deal with agent systems with opposite goals or commons goals acting collaboratively or cooperatively. A collaboration is established when an agent has autonomy to perform a task. In order to speed up the execution time, the agent is allowed to accept or ask other agents for help. Therefore, the cooperation occurs when an agent is not able to realize a certain task (Marietto, 2000). In Jennings (1995) and Nwana et al. (1997) a few reasons related to the coordination between multiple agents are described:

- Prevent the chaos among the agents: the coordination is necessary or desirable when the MASs are decentralized. Thus, the chaos is established quite easily;
- Distribute information, resources and expertise: through the coordinated distribution of information, resources and expertise, complex problems are solved more efficiently;
- Increase the efficiency of the system: through the exchange of information or shared of information, the coordination increases the efficiency of the system. Even the agents are able to perform the tasks, they can exchange their tasks if they are able to perform more efficiently;
- Manage interdependencies between the actions of the agents: interdependence occurs when the actions needed to achieve the goals of the individual agents are related;
- No individual agent has the ability, resources or enough information to solve the proposed problem in a separate way: most of the problems require different knowledge to be solved, which can only be achieved by different agents. Therefore, different knowledge of the agents is combined to find the desired result.

To find the best solution in a distributed intelligent system the coordination process manages the behavior of the agents because they have local incomplete information about the problem. The management aims to avoid conflicts, redundant efforts, deadlocks, etc. The higher incidence of such situations, less coordinated is the system.

3.2.3.1 Coordination Mechanism

Jennings (1995) and Bond & Gasser (1998) describe three most common coordination mechanisms.

- Organizational Structures: it is the formalization of the types of interaction between individuals. It offers a constraint environment and expectations about the behavior of the agents (i.e. through a set of rules) that guides the decision-making and the actions of the agents. Therefore, the relations specified by the organizational structures give long-term information about the agents and the society as a whole;
- Meta-levels Information Exchange: they are directly concerned to the concept of abstraction. According to Bond & Gasser (1998), the agents communicate with other agents semantically related to a common problem. Thus, it is possible to establish an abstract structure that formalizes these semantic relationships. The exchange information in a meta-level allows an agent to reason about the past, present and future activities of other agent;
- Multi-agent Planning: the shared tasks between the agents help the coordination mechanisms if the agent's behaviors are related to the overall goals of the system. It is possible to know how the actions performed by the agent affects the behavior of the system if we use the planning technique in the coordination process. In this case, the planning should take into the consideration issues such as plans control, inconsistencies, conflicts, etc.

3.2.4 Negotiation between agents

The negotiation represents an important technique related to cooperative activities and human societies, allowing people to solve conflicts that may interfere in the cooperative behavior. In the DAI, the negotiation between agents has the same purpose.
In general, the goals of a negotiation process are (E.H. Durfee, 1989):

1. Modify the local plans of an agent, the interaction does not occur successfully;
2. Identify situations where potential interactions are possible;
3. Reduce inconsistencies and uncertainties about different points of views or plans in common, through an organized exchange of information.

The negotiation mechanism provides efficiency to the system allowing the agents to redistribute the tasks between them in order to minimize an individual agent's effort. Thus, the negotiation will provide stability to the agents if their tasks are distributed correctly in the group (Faraco, 2001).

4. Pitfalls of Agent-Oriented Programming

In this section we identify some of the main pitfalls that any development project based on agent system can expect to find. Agent's technologies are not a universal solution even if they have been used in a wide range of applications. In many applications, using OOP is more appropriate. If a problem is solved with equal quality solutions using agent and non-agent approach, the non-agent approach is preferred since it is better understood by the software engineers involved in the system development. There is a misunderstanding that agents are the right solution to every problem. Therefore, they are often developed for inappropriate problems.
The interactions between agents yield unpredictable collective behavior in the general case. The patterns and the effects of their interactions are uncertain because of the agents' autonomy. The agents decide which of their goals require interaction in a given context. In

order to realize these goals, the agents decide which knowledge they will interact, and also when these interactions will occur (Jennings & Wooldridge, 2000).
In (Jennings & Wooldridge, 2000; Wooldridge & Jennings, 1998), some pitfalls are identified and happened in some of the agent development projects:

- You get religious or dogmatic about agents: even the agents have been used in a wide range of applications, their technology are not the only solution. For example, the OOP is more appropriate in many applications. However, there is a danger of believing that agents are the right solution to every problem. Therefore, sometimes they are developed for inappropriate problems;
- You don't know why you want agents: sometimes the concept of agent technology is not clear. But it is utilized to solve some problems only if someone read optimistic information about the potential for agent technology;
- You don't know what your agents are good for: it is important to understand how and where the agent technology is usefully applied.

5. Multi-Agent Systems platforms

Due to the fact that many of the MAS characteristics are independent of the application, frameworks started to be utilized to facilitate the development of such systems. These frameworks provide the basic functionality of MAS, which allows the developers to concentrate in the development of the agents. The main goal of the Foundation for Intelligent Physical Agents (FIPA) is to develop the standard implementation of the open, heterogeneous and interoperable agents. The foundation was created in 1996s and is also responsible to define some standard agents that many developers use to ensure the interoperability between MAS developed with generic frameworks.
Based on the vision of interoperability between systems with different manufacturers and operators, FIPA released as reference the FIPA standard pattern. This pattern has a primary focus on the external behavior of system components, leaving open the implementation details and the architectures of the agents. In addition to this feature, it sets the reference model for an agent's platform, as well as the set of services to be offered by the platform. Among these services there are: Directory Facilitator (DF), Agent Management System (AMS), the Message Transport Service (MTS), and the Agent Communication Language (FIPA-ACL). FIPA does not formally implement any agent architecture because its open standards allow various ways to implement it, simply by following the recommendations and abstract mechanisms defined within. Among the generic frameworks that use the FIPA pattern it is possible to cite: JADE (Bellifemine et al., 2007) and FIPA-OS (Poslad et al., 2000). These platforms are described in the subsections 5.1 and 5.2, respectively.

5.1 JADE platform

The Java Agent Development Framework (JADE) is an environment for developing applications according to the FIPA patterns. It is implemented in Java and was developed at the University of Parma, Italy (Bellifemine et al., 2007). Some characteristics of this platform are listed below:

- Distributed platform of agents - JADE can be divided into multiple hosts or machines. The agents are implemented in Java threads and inserted into repositories of agents called containers, which provide all the support for the implementation;

- Graphical user interface (GUI) - JADE has a GUI interface that assists in managing agents and agent containers;
- Running multiple, parallel and concurrent activities of agents - JADE provides these features through their pre-defined models of agent's behavior. The structure of the agents behaviors using the JADE platform takes place via a scheduler that automatically manages the scheduling of these behaviors.

5.1.1 Platform's architecture

The JADE architecture is based on the coexistence of several Java Virtual Machines (JVMs) that can be distributed over multiple computers, independently of the operating system.
Figure 1 shows the distribution vision of the JADE platform in many hosts. Each host run the JADE agents that forms a container. These containers are registered in the main container of the platform. In each host has a JVM, indicating platform independence, and in each JVM has a container of agents that provides a complete running environment for these agents, in addition to allowing multiple agents to run concurrently on the same processor/host. The execution of the JADE platform occurs at the main container of a platform. The other hosts who own the remaining containers should only have the files needed to run the platform.

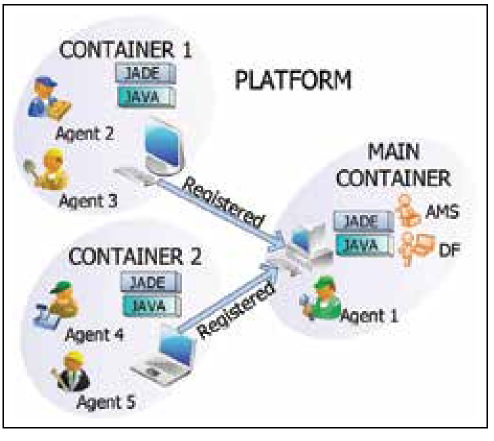


Fig. 1. Functional Architecture of the Jade Platform.

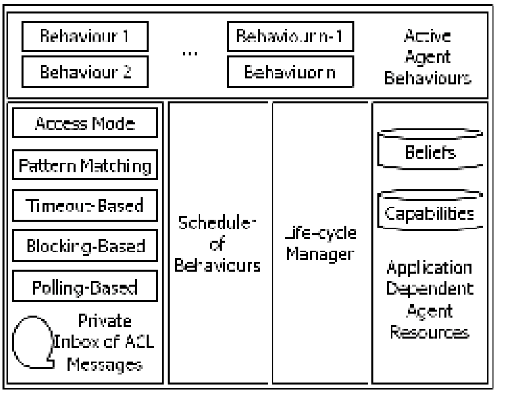


Fig. 2. Internal Architecture of an Agent in the Jade Platform (Bellifemine et al., 2007).

Figure 1 shows the main container where the AMS, the DF and the Remote Method Invocation (RMI) registry are located. The RMI is a name server used by Java to record and retrieve references of objects by name. With the RMI registry the JADE platform keeps references of other containers that connect to it.

5.1.2 Agents in the platform

In the JADE environment an agent is a process that is autonomous with an identity and requires communication with other agents to execute their functions. The JADE platform is neutral as regards of internal architecture of its agent. A JADE agent runs as a thread that employs multiple tasks or behaviors and simultaneous conversations.
Figure 2 shows the internal architecture of the JADE agent. At the top there are the active behaviors of the agent that represent actions that the agent can perform. The computational model of an agent in JADE is multitasking, where tasks (behaviors) are

performed concurrently. Each functionality provided by an agent must be implemented as one or more behaviors.
At the bottom of Figure 2 is possible to see a private messaging box of ACL. Every agent in JADE has this box, and can decide when to read incoming messages and which messages to read. In the center, there are the behavior scheduler and life cycle manager. The scheduler is responsible for scheduling the execution order of the behaviors. The life cycle manager is the controller of the current state of the agent. The agent is autonomous, it uses the life cycle manager to determine their current status (active, suspended, etc.). On the right side of Figure 2 there are the application-dependent capabilities of agents, where will be stored the beliefs and capacities that the agent acquires during the execution of the application.

5.2 FIPA-OS platform

The FIPA-OS platform was originally developed by Nortel Networks (Poslad et al., 2000) and currently is being developed by Emorphia Corporation. It is implemented in JAVA and its architecture has three types of components: mandatory, optional and switchable. The mandatory components are required for the execution of agents. The developers decide to use or not the optimal components. The switchable components have more than one implementation, which allows to choice the best implementation that adapt the necessity of the system. These components are illustrated in Figure 3 and will be discussed in the following sections.

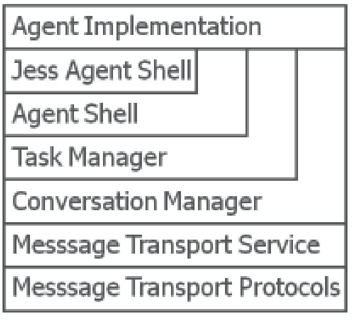


Fig. 3. Components of the FIPA-OS Platform (Poslad et al., 2000).

5.2.1 JESS Agent Shell

The Java Expert System Shell (JESS) is a tool for developing expert systems. The expert system is a set of rules drawn from a collection of facts, which are performed when a set of preconditions are satisfied. The structure of the JESS Agent provides an interface to the JESS, allowing agents to have a knowledge base and deliberative capacity (Buckle & Hadingham, 2000).

5.2.2 Agent Shell

The FIPA-OS platform provides a model called Agent Shell for the construction of agents. Agent Shell is a mandatory component that facilitates the implementation of FIPA-OS agents through an extended set of base classes. Through these base classes several features are provided such as transport, retrieval and buffering messages (Buckle & Hadingham, 2000). However, developers of agents does not necessarily need to use the Agent Shells, as FIPA-OS enables the independent development of agents.

5.2.3 Task Manager

Task Manager is a mandatory component that provides a way to split the functionality of an agent in small units of work, known as tasks. The purpose of this feature is to make the task as a piece of code that perform some functions and optionally returns some result (Buckle & Hadingham, 2000). Moreover, it must have the ability to send and receive messages, and should have a little or no dependence on the agent that is running it.

5.2.4 Conversation Manager

All messages exchanged between agents are part of a conversation. The Conversation Manager is a mandatory component that allows it to be kept a navigable history of conversations between agents (Buckle & Hadingham, 2000), as well as mechanisms for grouping messages of the same conversation. It is also the responsibility of this manager to ensure that both sides of a conversation use the same protocol.

5.2.5 Message Transport Service

Message Transport Service (MTS) is a mandatory component that provides capacity for sending and receiving messages between agents. The MTS of FIPA-OS platform is organized in a model of layer of services. Messages sent by an agent coming to MTS and are modified as they are moved from a higher to a lower layer, until they reach the Message Transport Protocol (see Subsection 5.2.6).
The protocol to be used to send the message depends on whether the message is for local or for a remote platform. If local, the message is sent to the destination agent passing through the layers of services in reverse order. If the message is intended for a remote platform, the service layers through which the message should go will depend on how the remote MTS is implemented.

5.2.6 Message Transport Protocol

Message Transport Protocol (MTP) is a switchable component that provides implementations of the protocols used by the MTS for sending and receiving messages (Buckle & Hadingham, 2000). The FIPA-OS platform divides these protocols into two types: internal and external. Internal protocols are responsible for carrying messages between agents in the same platform. For this task the system uses the FIPA-RMI. External protocols are responsible for carrying messages between agents in different platforms. For this task FIPA-OS uses the Internet Inter-ORB Protocol (IIOP). The IIOP is a universal standard for communication between distributed objects, having the advantage of being interoperable with any languages that support its specifications.

6. Multi-agent system applications

In this section we demonstrate in detail two multi-agent applications based on the JADE platform that were modeled and implemented by our Social and Artificial Intelligence research team at the Centre of Mathematics, Computation and Cognition (CMCC) of the Federal University of ABC (UFABC), Brazil. The two applications are: a cognitive modeling of stock exchange and a military application of real-time tactical information management.

6.1 Modeling cognitive Multi-Agent System for stock exchange

Stock exchanges play a major role in the global financial system. Among the most known stock exchanges we can mention: Dow Jones (New York, USA), Nikkei (Tokyo, Japan), and BM&FBOVESPA (São Paulo, Brazil). The behavior of the stock market is complex and dynamic, which characterizes this system as unpredictable. Even with such unpredictability the stock market possesses unique characteristics and some traits in common that when viewed in general, it is possible to detect certain stability (Azevedo et al., 2008; Rutterford & Davison, 1993).

Among the observed patterns in stock market, beyond the fluctuation of prices and indexes, it is also noted social phenomena that emerge between its investors. Aiming to better understand the formation of behavior patterns in a stock exchange, this paper presents a social strategy to determine the behavior of the investor based on the theory of imitation ((Wei et al., 2003)). This theory is based on the concept of herding behavior, which says that the behavior of an agent is a function of imitation of the others attitudes. Thus, in this simulation investor behavior is modeled for his/her decision is based on the behavior of other investors in the same stock market environment. With the development of this work it will be possible to study the social behavior, improve techniques in the field of AI, and to analyze an investment strategy in details.

The analysis of the social strategy of imitation in the stock market, the operation of a stock exchange was modeled encompassing elements such as investors, stock brokerage and financial market. As theoretical and technical basis will be used the concepts of the field of DAI, specifically MAS.

6.1.1 Theoretical basis of stock exchange

According to (Fama, 1965) stock exchanges are organized markets which negotiate equity securities denominated shares of publicly traded companies. Trading in the securities occurs when an investor passes his order to buy or sell shares to the market. If there is another transaction of equivalent value in the opposite direction (to sell or buy), then the deal is done. Thus, the share price is set by supply and demand of each share. To make deals at the stock market there are companies that perform the necessary procedures. They are called stock brokers and act as an interface between the investor and the stock exchange for buying and selling shares (Rutterford & Davison, 1993).

The market index is estimated from the shares prices with greater trading volume. This index serves as a mean indicator of market behavior, and is used as a benchmark by the investors to follow its behavior over the time. The first index calculated on the day is called open index and closing index is the last. The minimum and maximum index are, respectively, the lowest and highest value recorded on the day. When the closing index of the day is superior to the previous day's then it is said that the stock market is high. If it is lower, then it is said that the stock market is down. The market is considered stable when these values are equal (Rutterford & Davison, 1993).

6.1.2 Modeling a stock exchange with cognitive Multi-Agent System

The MAS modeled in this section simulates the cognitive environment of a stock exchange, which contains the main elements shown in Figure 4.

It consists of a MARKET agent and several STOCK BROKER MASs. The interaction between these entities occurs when the MARKET agent sends data (such as stock index and companies shares) to the STOCK BROKER MASs. The brokers communicate with each other to know what decision (buy, sell or stay with the shares) the other brokers took. In the process of decision making each STOCK BROKER MAS uses the theory of imitation to define what will be its behavior. With the decision, the purchase or sale of shares is authorized between the

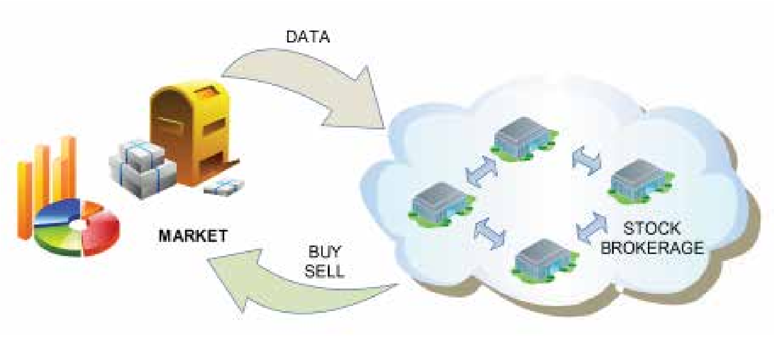


Fig. 4. The Main Elements of Stock Exchange MAS.

brokers. Each STOCK BROKER MAS is composed of five agents: COMMUNICATOR, MANAGER, DECISION MAKER, BUYER and SELLER (see Figure 5). This distribution aims to compose the STOCK BROKER with expert agents, each with a feature. This division facilitates future expansion of the simulation, for example, to adopt a different strategy for making decisions it is necessary just replace the agent responsible for this process.

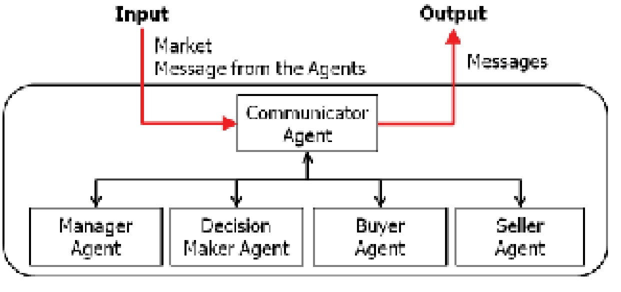


Fig. 5. STOCK BROKER MAS Architecture.

6.1.2.1 Agents of the STOCK BROKER MAS

The modeling of STOCK BROKER´s agents was based on BDI architecture of (d'Inverno et al., 1998a) and on general BDI architecture presented in (Wooldridge, 1999). These architectures are described in Section 3.1.3.2 and they are combined to join the functionality of sub-goals of the dMars architecture and the filters, generators and reviewers of Wooldrigde architecture. As can be seen in Figure 6, the proposed architecture performs the following cycle:

- INPUT MODULE is responsible for the agent perceives the world and receives messages from other agents. The information is sent to the INFORMATION MODULE for analysis by the INFORMATION FILTER;
- The INFORMATION FILTERS classify information according to two criteria: (i) Belief - information that the agent does not have absolute certainty, both its content or source are questionable, (ii) Knowledge: information that the agent believes to be true;
- GOALS MANAGER accesses the INFORMATION MODULE to define what goals are achievable by the agent in a given time. The goals selected by the GOALS FILTER are sent to the INTENTIONS MODULE for further analysis;

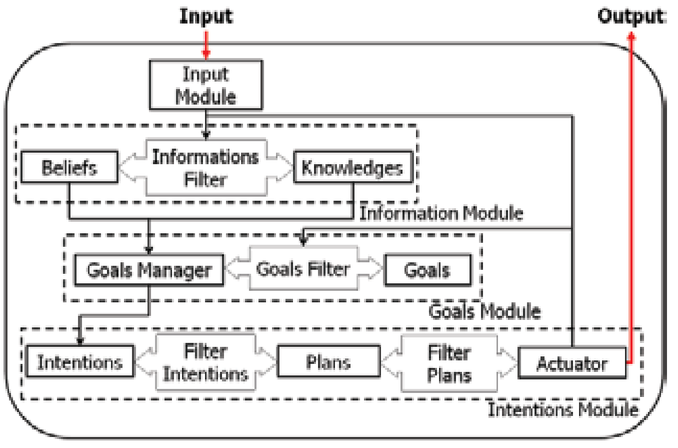


Fig. 6. Proposed BDI Architecture.

- The intentions are arranged in a row considering the priority order to be executed. Thus, the highest priority intention is the one the agent will try to achieve first;
- In the element PLANS there are the steps necessary to the agent perform and accomplish its intentions. These plans are defined in advance by the developer;
- ACTUATOR is responsible for the agent's action on the environment. It works as output to the world as it executes the plans for the intentions are met. During the execution of the plans, the ACTUATOR may: (i) insert new goals (subgoals), (ii) insert, remove and/or prioritize the intentions and/or information.

Now the features of each agents of the STOCK BROKER MAS are presented.

COMMUNICATOR Agent

The COMMUNICATOR agent is responsible for all the communications of STOCK BROKER MAS, acting as an interface with the outside world (other STOCK BROKER MAS and the MARKET agent) and its internal structure (formed by MANAGER, DECISION MAKER, BUYER and SELLER). Besides acting as an e-mail server, the COMMUNICATOR agent also assumes the functionality of general secretary because it knows who is responsible for each sector of the company.
This agent filters incoming messages and define them as part of their goals. Later, adopt a message as an intention and sends it to correct recipients. For that stores, a message queue that can be used for further processing, redirecting them or not when it seems necessary. Part of COMMUNICATOR agent's knowledge are (i) the messages, whether received or to be sent to COMMUNICATOR agents of other STOCK BROKER MAS, (ii) and MARKET agent.

MANAGEMENT Agent

The MANAGER agent stores global data relating to STOCK BROKER MAS. Information such as capital, stock portfolio, the final decision, how influenced the broker is, etc. It is also the responsibility of the MANAGER agent to send these information to other internal agents when required. The exchange of messages between the MANAGER and other agents is mediated by the broker's COMMUNICATOR agent. The MANAGER agent architecture is based on BDI architecture proposed in Figure 6. However, it lacks the GOALS MODULE because its functionality always provide data when requested.

DECISION MAKER Agent

In a stock brokerage firm, there are several types of analysts that define how to invest in the stock market: graphic analyzers, specialists in a niche market or specialists in high and moderate risk of investments. The DECISION MAKER agent implements the functionality analysis of the STOCK BROKER, deciding whether the broker will buy, sell or hold the shares it owns. In this work, the strategy adopted by the DECISION MAKER agent is imitation, so its behavior is based on the decision taken by the other STOCK BROKER MASs. The modeling of this strategy is based on the work of (Wei et al., 2003).

Investment preferences reflect the behavior of the investor's imitation, which is influenced by (i) macro factors (MF) represented by the stock index (ii) and a probability P to mimic the other investors, characterizing the investor's permittivity (degree of imitation).

P is a random number given to each agent when it is instantiated. The MF is a normalized value between 0 and 1 taking into the consideration the higher and lower stock index. The objective of MF is to quantify the economic situation of the market. With the MF above 0.5 agents tend to buy stocks, whereas the opposite trend is to sell. Given the state of the MF and the highest number of one type of behavior of other investors (buy, sell or hold), the agent in question defines its behavior.

Thus, the knowledge base of DECISION MAKER agent are composed by the coefficient of permissiveness of the investor, the stock exchange current index, the current MF and the most common decision taken by other STOCK BROKER MASs. To determine the most common decision taken, the DECISION MAKER agent requests to the COMMUNICATOR agent to send a broadcast message to other STOCK BROKER MASs. Until receiving all the answers from the brokers, the decision is suspended. After accounting the most frequent behavior, the DECISION MAKER agent decides its behavior.

With the goal chosen to be promoted to the intention (to buy, sell or hold shares), the DECISION MAKER agent performs a plan. If the intention is to purchase or sell the BUYER or SELLER agent, respectively, is triggered via the COMMUNICATOR agent to start in executing their tasks. If the decision is to keep the shares it owns, nothing is done.

BUYER Agent

In distribution of areas inside the corporation, there is a clear distinction between buying and selling areas. For STOCK BROKER MAS modeling this distinction was also used. The BUYER agent has the function of executing, purchase transactions of shares with the MARKET agent when the broker makes this decision. The purchase decision is made by the DECISION MAKER agent, which informs the BUYER agent. The MANAGER agent, through the COMMUNICATOR agent, informs the BUYER agent the current value of broker's capital.

The BUYER agent's knowledge consists of the shares it may acquire and their prices. After determining that it will buy shares and own the necessary capital for the purchase, it will make a purchase offer to MARKET agent. The proposed purchase may be accepted or not. If there is confirmation of the transaction, the capital employed will be charged to STOCK BROKER's capital and the shares purchased will be added to its portfolio.

SELLER Agent

As there is an agent to make the purchase, also there is an agent to conduct the sale of shares that the STOCK BROKER MAS has. The SELLER agent has the role of conduct sales transactions with the MARKET agent when the broker makes this decision. The actions executed for sale are similar to the purchase, but the SELLER agent, instead of capital, requests the MANAGER agent the shares portfolio of STOCK BROKER MAS.

The SELLER agent's knowledge consists of the shares and their respective prices. After informed that it must sell the shares, it will make the selling proposal to MARKET Agent. The proposed sale may be accepted or not. If there is confirmation of the transaction the capital from the sale will be credited to the capital of the STOCK BROKER, and the respective shares will be removed from its portfolio.

6.1.3 Case study

The STOCK EXCHANGE MAS was implemented using the JADE platform. The system was run using actual values of the BM&FBOVESPA index of 2004. In this simulation is established a communication between the agents that make up a STOCK BROKER, MARKET agent and other broker's COMMUNICATOR agents. At the start of the simulation the MARKET agent sends to all COMMUNICATOR agents (which are registered on the platform) a message informing the stock index value, characterizing the opening of trading. Thereafter every STOCK BROKER shall perform its flow to make decisions and make deals with the MARKET.

During the execution of the simulation it is possible to see that agents gradually converge to similar behaviors. In early trading, when the index is down, some agents hold the majority shares and start selling the shares it owns. In the middle of the year with the onset of high index, the agents start to buy shares in small numbers, increasing the number of agents with this behavior as the index increases at each interaction. It is perceived that agents with low permittivity are less influenced by others. The agents with high permittivity are more susceptible to the decision of others, thereby mimicking the behavior of majority of the agents assigning low relevance for the stock index.

6.1.4 Future works

Among the possible extensions of this work, there is the possibility of modeling the concept of reputation for each STOCK BROKER MAS. With this concept, it will be possible that each broker would have a perception about the "reliability" of the other brokers. This information can assist the process of decision making regarding to the investment, because the imitation strategy can now be adjusted to take into consideration the status of the reputation of each broker's information.

Another proposed improvement for the system is to adapt the DECISION MAKER agent to make more specific decisions, based on individual performance data of the companies participating in the stock market. For example, in the same round this agent may choose to sell a company's share and buy another if the first company is performing poorly while the second is on the rise. Or in a more complex reasoning, where two companies are on the rise, but one company is more promising than another. So the agent chooses to sell shares of a company to raise capital, with the intention to purchase shares of the company that has a higher profit outlook.

6.2 Multi-Agent Systems for real-time tactical information management

With technological advances implemented in today's battlefields, arises the need for the command of military operations enhances the ability of decision making by responding quickly and efficiently to the events in the field. Dealing with a growing number of variables and the continuous increase in the complexity of actions, there is a need for increased speed and quality of information processing. Within this context the armed forces fight today with a new perspective, using advanced technologies to improve their decision quality and,

consequently, the chance of victory (Cil & Mala, 2010), (Xuan et al., 2001),(Bonabeau, 2002), (Barbuceanu & Fox, 1995), (Wijngaards et al., 2002).
Following this new perspective, the current processes of decision making in combat use pre-processing technologies and information distribution that are based on symbols. These symbols represent important information for the participants of a specific mission. This simplified representation of the environmental elements and the dynamics of action allow a clearer interpretation of the direction of activities and improve the prospect for a global action in real time (Panzarasa et al., 2002).
To aid the process of decision making in areas of conflict, this work proposes a MAS where soldier agents interact with cyber agents. From this interaction will be possible to map and keep under surveillance a particular area. This work will also allow analysis of how this collaboration influences the quality of decision making, and what are the most significant characteristics that emerge from such human-machine interaction.

6.2.1 Multi-Agent System modeling tactical information in combat situations

To aid the process of decision making on the battlefield, this work proposes a MAS composed of cyber agents (radar and helmets) interacting with human soldiers and commanders. Figure 7 shows SOLDIER agents on patrol, whose mission is to map and keep under surveillance a particular area. SOLDIER agent carries RADAR agents, which are left over from the walk aiming to maintain surveillance of these areas after his/her passage. The CONTROL agent receives the information from all agents (SOLDIER, RADAR), processes and distributes to the soldiers a simplified representation of the information of the environment (projected on the visor of his/her helmet), also showing for each soldier the direction to follow.
The environment in which the action takes place consists of several obstacles: lakes, roads, forests, enemies, among others. The decision to fight or flee from obstacles is coordinated by the CONTROL agent who evaluates the possibilities in order to achieve tactical superiority in action.

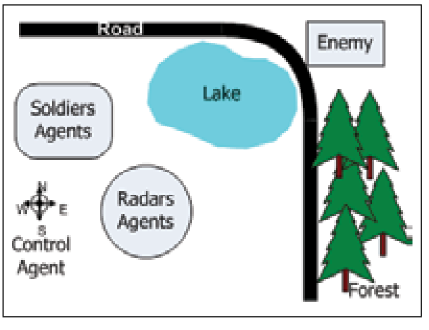


Fig. 7. Overview of a Combat Situation.

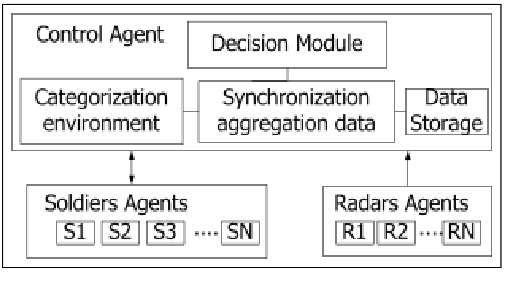


Fig. 8. General Architecture of MAS of Real-Time Tactical Information in Combat Situations.

The MAS proposed in this work to represent a war situation is illustrated in Fig 8. It consists of the following agents: RADAR, SOLDIER and CONTROL (presented in Subsection 6.2.1.1). This MAS is immersed in an environment that is described in Subsection 6.2.2.

6.2.1.1 Agents of MAS

The following are presented the features of the proposed MAS.

RADAR Agent

As the soldiers move on the ground they install radars, under the control and command of the CONTROL agent, aiming to maintain surveillance after leaving a particular area. After installed, the RADAR agent, continuously sends the CONTROL agent images that are in its scanning spectrum (within the region that is at your scope). In addition to send this data to the CONTROL agent, the RADAR agent also broadcast its data to the allied soldiers who are in the range.

SOLDIER Agent

Each human soldier has a helmet that receives and sends information in real time to the allied soldiers and their commanders. Both human soldier and helmet are represented in this work by the SOLDIER agent. Figure 9 represents the integrated architecture of the SOLDIER agent.

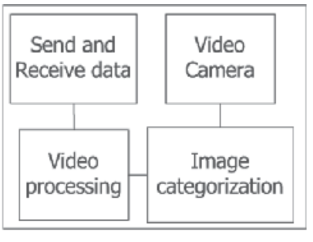


Fig. 9. Architecture of SOLDIER Agent.

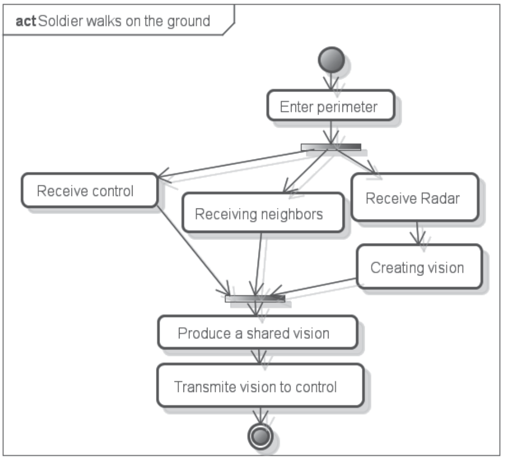


Fig. 10. Behavior of SOLDIER Agent.

The VIDEO CAMARA module of helmet captures images that are within eyesight of the soldier. The IMAGE CATEGORIZATION module recognizes environmental patterns warning the soldier about the position of weapons or recognizable obstacles, present in its active range. The SEND AND RECEIVE DATA module continuously sends the vision rendered in the display of the SOLDIER agent to the CONTROL agent, as well as for the other allied soldiers who are in their range. This module also receives the images of the CONTROL agent and displacement orders. Such data are analyzed by the SOLDIER agent, together with its own information.

In conflict region may be necessary to perform the mapping of obstacles and enemies, keeping the region under surveillance. For combat situations is defined one perimeter for mapping and surveillance. When the SOLDIER agent enters this perimeter, it starts the process of continuous exchange of messages between him, the CONTROL agent and other agents. Figure 10 illustrates this process. Messages from CONTROL agent are shown with priority on the display of the SOLDIER agent, which can orient his moves over the terrain to cover the region and scan for weapons and enemies.

CONTROL Agent

The CONTROL agent is responsible for receiving all messages from SOLDIER agent and RADAR agent, summarizing them in a format of matrix of symbols. Figure 11 describes the behavior of the CONTROL agent for two types of possible inputs: data transmitted by the SOLDIER agent, and by the RADAR agent.

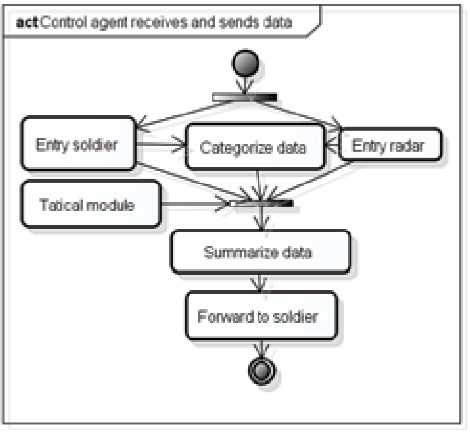


Fig. 11. Behavior of CONTROL Agent.

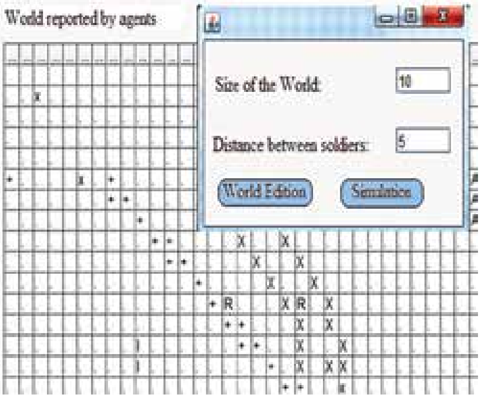


Fig. 12. Symbols Representation in the Matrix of the Environment.

The signal transmitted by the RADAR agent must pass through the IDENTIFICATION AND IMAGE CATEGORIZATION OF THE ENVIRONMENT module to transform the images in standard symbols. If the image comes from the SOLDIER agent, the categorization is not necessary because the images were already been processed in symbols by SOLDIER agent's equipment. The categorization is the identification of elements in the environment. In general the environment contains two types of obstacles: static (trees, rocks, rivers) and mobile (allied soldiers and enemies). When an element is identified by the categorization process, a specific symbol is used to represent this object, aiming to represent it with best level of information. Throughout this process the CONTROL agent builds up an overview of the battlefield and the dynamics of movement that drives the various components of the environment. This process is called aggregation.

In parallel, the TACTICAL module tries to recognize the direction of movement of allied soldiers and enemies, creating a tactical map with the different stored views. At the end of this process, the CONTROL agent builds an overview of the battlefield at every stage of combat. Based on this information, and with the objective to help the decision-making process of the soldiers, CONTROL agent executes two actions: (i) transmits to every soldier a partial view of the battlefield - this vision is related to the action and vision range of every soldier, and (ii) defines progress strategies for each soldier (tactical moves), and sends them individually.

6.2.2 The environment

The environment to be mapped and kept under surveillance can be quite complex in the real world, with natural obstacles such as rivers, lakes, depressions in the ground, buildings and vegetation. In this work, the environment where the combat situation occurs might contain the following elements: soldiers (allies and enemies), radar, trees, rocks and rivers. These elements will be displayed on the visor of each SOLDIER agent and on the interface of CONTROL agent, following the pattern presented in Figure 12.

The ground of the conflict area is represented as a matrix, and in each cell can be allocated an agent in every moment of combat. One of the advantages of using this representation is ease of manipulation for electronic systems. Since the main objective is to improve the speed of decision making, a lighter processing has many advantages over the use of complex imaging systems. In the Figure 12 the elements of the environment are represented with the following symbols:

- SOLDIER agent is represented by the character (S);
- Enemy soldiers are represented by the character (I) and may be of two types: those who walk and those that keep a fixed position;
- RADAR agents are represented by the character (R) and are inserted into the environment as the soldiers walk around. RADARS do not move;
- Trees are represented by the character (A);
- River banks are represented by the character (X);
- Stones are represented by the character (P) and can be isolated or grouped within the environment representing a rise or other natural obstacle;
- The symbol (+) represents the footprints (or the route) left by an enemy.

It can be seen in Figure 12 the initial screen of the simulation process where the size of the matrix and the distance between SOLDIER agents are chosen. The number of soldiers is calculated by the system as the size of the matrix divided by the distance between SOLDIER agents. As the simulation runs the partial visions of soldiers and radars are analyzed, compiled and aggregated by the CONTROL agent, who later returns for each SOLDIER agent only corresponding the part to his field of vision.

6.2.3 Future works

The application presented in this section describes a society of agents able to map and keep surveillance on any area or region. Therefore, in future works it is possible to extend this research to situations where the control flow of vehicles or persons should be evaluated for safety or improvement of the flow. A good example is the use of radars installed on highways as agents that send information about the average flow of vehicles, and the CONTROL agent with the ability to change the timings of traffic lights.

The implementation of the system using intelligent agents can be economically more attractive because it leverages the current infrastructure and has the characteristic of being resistant to failures. A radar system that does not send information for any reason would not invalidate the control operation of traffic signals, since for evaluating the mean flow of the traffic for decision making, is sufficient to collect the information from the other agents. Other advantages are the fact that the system can anticipate the arrival of a stream of cars and take a preventive action, and detect other abnormal behavior of the traffic.

7. Conclusions

We know that the hardware and software are constantly improving in performance and reliability. The programmers are also worry to improve their ability. For instance, many programmers try to minimize memory usage and to maximize throughput or processing speeds especially in the distributed applications. However, the low cost of the hardware make this problem almost disappear. The multi-agent technique tends to span many applications in different domains. The autonomy in the MAS allows the agents to learn the behaviors of the system to decide automatically, for example, the communication protocols and strategies, without the human intervention. The autonomy is constantly under improvement to make

the agents form and reform dynamic coalitions to pursue goals. In addition, the behaviors, the coordination between heterogeneous agents, faulty agents, languages, protocols may be improved.
The agent community has the interest to broaden the application of the agent technology and also the acceptance of the new paradigm among the researchers who are interesting to develop intelligent systems. The AOP has the potential to provide the best solution in different scenarios of abundance and opportunities, with the ability to create MAS that can work naturally on parallel and distributed environments. Agents that can collaborate and distribute tasks, and that can be built based on human characteristics like belief, desire and intention. The systems will be able to handle large numbers of agentes in the society. Also, the technology is easy to customize to provide new components and services. That is a technology to be followed closely.

8. References

Austin, J. L. (1962). *How to Do Things with Words*, Oxford University Press, London.

Azevedo, S., Teixeira, M., Costa, A., Borsato, B., Soares, F., Lucena, C. & de Janeiro-RJ-Brasil, R. (2008). Multi-agent system for stock exchange simulation-masses, *XXII Fourth Workshop on Software Engineering for Agent-oriented Systems (SEAS) of the Brazilian Symposium of Software Engineer (October 2008)*.

Barbuceanu, M. & Fox, M. (1995). COOL: A language for describing coordination in multi agent systems, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, Citeseer, pp. 17–24.

Bellifemine, F. L., Caire, G. & Greenwood, D. (2007). Developing multi-agent systems with jade (wiley series in agent technology), *John Wiley and Sons*.

Bittencourt, G. (2006). Distributed artificial intelligence.

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems, *Proceedings of the National Academy of Sciences* 99: 7280–7287.

Bond, A. H. & Gasser, L. (1998). Readings in distributed artificial intelligence, Morgan Kaufmann Publishers, pp. 3–35.

Bratman, M. E. (1987). *Intention, Plans, and Practical Reason*, Harvard University Press, Cambridge, Massachusetts and London.

Buckle, P. & Hadingham, R. (2000). Fipa and fipa-os overview, *Invited talk, Joint Holonic Manufacturing Systems and FIPA Workshop, London*.

Cil, I. & Mala, M. (2010). A multi-agent architecture for modelling and simulation of small military unit combat in asymmetric warfare, *Expert Systems with Applications* 37(2): 1331 – 1343.

Coutinho, L. R., Brandao, A. A. F., Sichman, J. S. & Hubner, J. F. (2009). A model-based architecture for organizational interoperability in open multiagent systems, *Proc. of the International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems*, pp. 128–140.

de Nunes, I. O. (2007). Implementation of the bdi waho architecture model.

Dennett, D. (1987). *The Intentional Stance*, MIT Press.

d'Inverno, M., Kinny, D., Luck, M. & Wooldridge, M. (1998a). A formal specification of dmars, *Intelligent Agent IV: 4th International Workshop on Agent Theories* . Architectures and Languages.

d'Inverno, M., Kinny, D., Luck, M. & Wooldridge, M. (1998b). A formal specification of dmars. intelligent agent, *4th Int. Workshop on Agent Theories*.

E.H. Durfee, V. L. (1989). Trends in cooperative distributed problem solving, 11(1): 63–83.

Fama, E. (1965). The behavior of stock-market prices, *Journal of business* 38(1).

Faraco, R. A. (2001). *Uma arquitetura de agentes para a negociação dentro do domínio do comércio eletrônico*, Master's thesis, Universidade Federal de Santa Catarina - UFSC, Brasil.

Fennel, R. D. & Lesser, V. (1977). Parallelism in artificial intelligence problem solving: A case study of gearsay ii, 26(2): 98–111.

Gottifredi, S., Tucat, M., Corbatta, D., Garcia, A. J. & Simari, G. R. (2010). A bdi architecture for high level robot deliberation.
URL: *http://dialnet.unirioja.es/servlet/oaiart?codigo=3214348*

Group, K. A., Finin, T., Mckay, D. & Fritzson, R. (1992). An overview of kqml: A knowledge query and manipulation language.

Hewitt, C. (1977). Viewing control structures as patterns of passing messages., 8: 323–364.

Huhns, M. N. (2004). Software development with objects, agents, and services, *Proc. of Third Int. Workshop on Agent-Oriented Methodologies, OOPSLA*.

Huhns, M. N. & Stephens, L. M. (1999). Multiagent systems and societies of agents, pp. 79–120.

Jennings, H. (1995). Controlling cooperative problem solving in industrial multi-agent systems using joint intention, 2(75): 195–240.

Jennings, N. R. & Wooldridge, M. (2000). Agent-oriented software engineering, *Artificial Intelligence* 117: 277–296.

Lind, J. (2000). Issues in agent-oriented software engineering, *Agent-Oriented Software Engineering III*, Springer Verlag, pp. 45–58.

Marietto, M. G. B. (2000). *Definição Dinâmica de Estratégias Instrucionais em Sistemas de Tutoria Inteligente: Uma Abordagem Multiagentes na WWW*, PhD thesis, Instituto Tecnológico da Aeronáutica - ITA.

Nwana, H., Lee, L. & Jennings, N. (1997). Coordination in software agent systems, p. http://www.labs.bt.com/projects/agents/publish/papers/coorination.html.

Odell, J. (1999). Objects and agents: How do they differ? (draft 2.2), *url: http://www.jamesodell.com*.

Odell, J. (2002). Objects and agents compared, *J. of Object Technology* 1(1): 41–53.

Panzarasa, P., Jennings, N. & Norman, T. (2002). Formalizing collaborative decision-making and practical reasoning in multi-agent systems, *Journal of logic and computation* 12(1): 55.

Parmigiani, G., Inoue, L. Y. T. & Lopes, H. (1997). *Decision Theory: Principles and Approaches*, John Wiley and Sons.

Paulussen, T. O., Zöller, A., Heinzl, A., Braubach, L., Pokahr, A. & Lamersdorf, W. (2004). Dynamic patient scheduling in hospitals. coordination and agent technology in value networks.

Poslad, S., Buckle, P. & Hadingham, R. (2000). The fipa-os agent platform: Open source for open standards, *Proceedings of the 5th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents*, Vol. 355, Citeseer, p. 368.

Rao, A. S. & Georgeff, M. P. (1995). Bdi agents: From theory to practice, *Proc. of the First Intl. Conf. Multiagent Systems*.

Reichel, K., Hochgeschwender, N. & Voos, H. (2008). Opcog: An industrial development approach for cognitive agent systems in military uav applications, *Proc. of the 7th int. Joint Conf. on Autonomous Agents and Multiagent Systems: Industrial Track*, pp. 97–100.

Reis, L. P. (2003). *Coordenação em Sistemas Multi-Agentes: Aplicações na Gestão Universitária e Futebol de Robôs*, PhD thesis, FEUP - Portugal.

Rentsch, T. (1982). Object oriented programming, *SIGPLAN Notices* 17(12): 51.

Rutterford, J. & Davison, M. (1993). *An Introduction to Stock Exchange Investment*, MacMillan.

Sawyer, R. K. (2003). Artificial societies: Multiagent systems and the micro-macro link in sociological theory, *Sociological Methods and Research* 31: 325–363.

Searle, J. R. (1969). *Speech Acts: An Essay in the Philosophy of Language*, Cambridge University Press.

Shoham, Y. (1993). Agent oriented programming, *Artif. Intell.* 60(1): 51–92.

Silva, V., Garcia, A., Brandão, A., Chavez, C., Lucena, C. & Alencar, P. (2003). Taming agents and objects in software engineering, *Software Engineering for Large-Scale Multi-Agent Systems: Research Issues and Practical Applications, volume LNCS 2603*, Springer-Verlag, pp. 1—25.

Wagner, G. (2003). The agent-object-relationship meta-model: Towards a unified view of state and behavior, *Information Systems* 5(28): 475–504.

Wahono, R. S. (2001). *Intelligent agents for object model creation process in object-oriented analysis and design*, Master's thesis, Department of Information and Computer Sciences, Graduate School of Science and Engineering, Saitama University, Saitama - Japan.

Wei, Y.-M., jun Ying, S., Fan, Y. & Wang, B.-H. (2003). The cellular automaton model of investment behavior in the stock market, *Physica A: Statistical Mechanics and its Applications* 325: 507–516.

Wijngaards, N., Overeinder, B., van Steen, M. & Brazier, F. (2002). Supporting Internet-scale multi-agent systems, *Data & Knowledge Engineering* 41(2-3): 229–245.

Wooldridge, J. M. (1999). Intelligent agents, *Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence* pp. 27–77.

Wooldridge, M. & Jennings, N. R. (1995). Intelligent agents: Theory and practice, *The Knowledge Engineering Review* 10(2): 51.

Wooldridge, M. & Jennings, N. R. (1998). Pitfalls of agent-oriented development, *Proc. of the Second Int. Conf. on Autonomous Agents (Agents 98*, ACM Press, pp. 385–391.

Xuan, P., Lesser, V. & Zilberstein, S. (2001). Communication decisions in multi-agent cooperation: Model and experiments, *Proceedings of the fifth international conference on Autonomous agents*, ACM, pp. 616–623.

Part 4

Multi-Agent Systems Simulations and Applications

Multimedia Authorship Tool for the Teaching of Foreign Languages and Distance Learning in a Multiagent Environment

Adroaldo Guimarães Rossetti[1], Almir dos Santos Albuquerque[2],
Vasco Pinto da Silva Filho[2] and Rogério Cid Bastos[3]
[1]Researcher at Embrapa Agroindústria Tropical,
[2]Student Doctorate in Engineering and Knowledge Management-EGC, Federal University of Santa Catarina-UFSC,
[3]Professor of Graduate Engineering and Knowledge Management-EGC, Federal University of Santa Catarina-UFSC
Brazil

1. Introduction

The agility of progress in the globalized world, in communications, in technology and in organizational activities demands more and more that people have formal knowledge of other languages. In this context, control of at least secondary level knowledge and other languages, such as the English language, predominant in practically all areas of knowledge and activities, is presented almost as survival need in people's lives. Although this reality is practically national conscience, official teaching, particularly in Brazil, doesn't seem to lead a way to suppress that need in a short term.

The Law of Basis and Guidelines of National Education (LDBEN) N° 9394, of December 20th, 1996 (BRAZIL, 1996), in Article 24, deals with basic education, in elementary and secondary levels. Topic IV of this article mentions that:

[…] classes can be organized, with students of distinct grades, with equivalent levels of knowledge in the subject, for the teaching of foreign languages […].

As far as is known, however, this still doesn't seem to be a reality, at least in official establishments, where education is offered by Brazilian state. It is understood that only what is settled on § 5º, of article 26 is followed, in which mentions that:

[…] in the diverse part of the curriculum, will be included, necessarily, as of fifth grade, the teaching of at least one modern foreign language, which will be chosen by the scholar community, within the institution's possibilities.

Despite not making explicit which "modern foreign language", English and French usually prevail in most regions of the country. This prevalence is a sharp expression of the recognition of the importance of the English language. It is convenient to register, however, according to Rossetti *et al.* (2009), that in regions with strong influence of immigrants (Germans, Italians, Japanese, etc), such as the South and the Southeast, this prevalence is less accentuated.

Even with the inclusion, "necessarily as of the fifth grade", independent of the language, this condition offers only elementary notions especially for two reasons. First, because the teaching of this language doesn't continue throughout the entire period, as the expression "as of" suggests. Second, the scantiness of the class schedule doesn't allow bigger advancements and stops there, not offering good learning conditions for foreign languages. Except for those who enroll in university, in a Letters course, where they can really specialize in any given language, there are few alternatives to deepen knowledge in this area. The only option left to acquire suitable knowledge are private foreign languages institutes or schools, which almost always have high costs and inflexible hours, which, in general, keep a considerable part of the population from inserting themselves in this process.

Some linguistic conceptions found in educational softwares hold two stages: one consists in the structural analysis of a sequel derived from the base; the other, of a structural change in that sequel. This premise supports the linguistic conception of this work, which originates from an analysis of the situation of foreign language teaching in Brazil, to then discuss the possible transformations caused by the suggested model. The methodological conception, on its turn, is centered in the study and analysis of the problems of teaching a foreign language in Brazil, identifying the causing factors which will be considered by the intervention of the proposed tool.

Both in secondary school, distance learning and undergraduate teaching and in traditional foreign language teaching, usually classes are ministered in days and times determined by the school, according to a predetermined calendar, almost always inflexible. This system, which does have its upsides, can hurt students that, for any given reason, are prevented from watching one or more classes successively. The same problem can be transferred to those who, for whatever circumstance, are eventually compelled to a temporary interruption of frequency in a determined module of the course, independent of the stage of said module. The occurrence of any of these cases may cost the student to lose track of taught subject, making it harder to progress with good improvement, once that, in cases of this nature, missed classes can hardly be recovered parallel to the normal course due to a number of factors. When a lot of classes are missed, the time needed solely make it not viable; when the missed classes are few and the time is available, the elevated costs usually imposed can be unviable to many students. On the other hand, students who learn any subject or language easily, with conditions to progress more rapidly are, sometimes, also penalized, once this structure usually doesn't support the structuring of classes that take this characteristic under consideration. And when that is possible, classes in general have few students, increasing costs, making it an obstacle for many. Under these circumstances, waiting for the start of a new class in the same level or module has been an alternative for those who wish to do it well, although that implies increasing the learning time of a given discipline or domain of the language of study.

Workers who wish to deepen their knowledge face not so different situations, sometimes, even for professional needs. In that sense, Pettenati *et al.* (2000) mentions that because of distance or difficulties to conciliate organizational appointments, a lot of people face difficulties to progress in the acquisition of knowledge and competence, for in the majority of times, these aspects depend on a teacher not always available in organizations.

Therefore, the lack of flexibility to start or restart a course, particularly a foreign language one in any period, time or place, is a problem to be dealt with, due to the importance of a teacher in this type of teaching and learning. In these courses, more than in any other, aspects such as pronunciation, hearing perception, repetition, reading and writing, for instance, have great importance and are object of orientation by the teacher through teaching material and exercises. Furthermore, these characteristics practically constitute the four abilities of language: understanding oral language, speaking, reading and writing, in which bases, according to Widdowson (1991), the courses of foreign languages are established. These abilities refer to the type of activities students must do when coursing an idiom, which a lot of teachers agree with. Therefore, those should be the bases for generating teaching material for these courses.

A flexible structure is needed, one that allows the interested person to initiate the course in any period, time and place, according to their possibilities, conveniences and to the level of knowledge they already have of the discipline or language which they wish to take. Or retake, from where eventually happened the interruption. Plus, it should allow those who learn more easily the possibility of completing their studies more rapidly, according to their abilities, necessities, availability or conveniences. It is essential to have, in this structure, teaching material, texts, reading and writing orientation; sounds and images that reflect clarity of pronunciation and develop hearing perception ability; exercises directed to improve pronunciation, the right spelling of words and the fluency, in case of idioms, besides exercises, referring to other available disciplines. It is also important to have a good dose of incentive and motivation to accelerate the learning process in the student.

Currently, however, according to Pettenati *et al.* (2000), these barriers are fully transposable with the help of Information and Communication Technology – ICT. The use of technology in this process is essential, for, according to Ramos (1996), the incorporation of technology to the learning process offers, concretely, the opportunity to implement a new model or teaching pattern. Furthermore, technology has the potential to produce new and rich learning situations. According to Coutaz (1990), the one who conceives an interactive system should offer a description, as accurate as possible, of the problems and cognitive processes of its user, in a way that it not only gives support, but also motivates, more and more, the learning process.

As stated by Kaskalis, Tzidamis & Margaritis (2007), with the explosion of multimedia computation, educators have been trying to achieve a total human-computer interaction, favoring the progress of teaching and learning. Consequently, a great number of multimedia tools have been created, always following the tendency of a simpler and easier development of these tools. Many actions have been taken to facilitate this intention: since the use of the multiple computer resources to the introduction of series of variables to base an appropriate evaluation. So far, according to these authors, there is no news of a tool that has completely fulfilled such task.

For Bailey & Konstan (2000), the lack of an authoring tool that favors the satisfactory interaction of the teaching and learning process happens because, although these tools are usually developed by great professionals in the computer area, it not always has the necessary cooperation of specialists in the subject to which the tool is destined. Besides, they are rarely validated with the participation of final user.

To Moran (2003), with the evolution of digital technology, online education has emerged as an excellent alternative for the advance of knowledge. To this author, online education involves a series of teaching and learning actions developed through telematic means, such as the Internet and the use of all its information and communication devices, useful to the improvement of inter-relational processes.

2. Technical revision

Technological advancements have favored the implementation of support tools to the regular teaching of foreign languages process in an attempt to improve the methods of teaching and learning, such as the ones by Courtillon & Raillard (1983) in the French Alliance, for instance. The execution of this practice consists in using the computer as a help instrument to teach the French language, using multimedia resources to listen, speak and write.

A number of initiatives have been taken with the use of technology, searching to make knowledge accessible to a greater deal of people. With that in mind, Pettenati *et al.* (2000) proposed a tool for distance learning, intending to make instruction available, without hurting the work, making it more accessible and efficient to whomever is interested, particularly employee and company. The proposed tool describes a *webbased*, author of a tutor learning system, conceived to suppress the difficulties of distance or organizational appointments. The environment is composed by several interconnected systems, including:

a. description of the course, with guidelines and agenda;
b. work plan and themes for revision of studied subjects; and
c. a structure that facilitates periodical self-evaluation.

It's a tool that, according to the authors, successfully complies with the goals it commits to.

Under the technological education in the teaching area view, Knowledge Engineering, through Artificial Intelligence techniques, may be the way to offer more options to who wishes to widen their knowledge, due to the resources it has. There are many available approaches to work knowledge, involving the acquisition, the storage and the use of knowledge, for example. In that way, AI can be classified under various categories. As for problem solving method: in Symbolic AI (SAI), Conexionist AI (CAI), Evolutionary AI (EAI) and Hybrid AI (HAI). As for location: in Monolithic AI (MAI) and Distributed AI (DAI). SAI uses logic as a basic tool to deal with knowledge, with its rules of inference inspired in syllogisms. CAI using Artificial Neural Networks applies to "not well defined" problems, but that are made known through examples. Among the fields of application of the Conexionist techniques, including most importantly artificial neural networks, are: knowledge of patterns; control of industrial processes; robotics and as option to the reasoning techniques based in cases, for the resolution of problems. In EAI, the mechanisms used are the same ones found in biological evolution. It is an example of "well defined" problem, of survival of a species in a variable environment. It can be looked at as an optimization process, although having variable restrictions, and, sometimes unknown. HAI gathers advantages of more than one method of approach to resolution of problems. MAI involves simple systems with no modularity, such as the specialist systems, for instance. The operation of DAI depends on a certain group of parts (or modules) to resolve a determined problem in a cooperative form. Its modularity to find solutions to the problems is directly connected to the concept of agents.

It was with AI that the first systems for teaching through computers came about, with Computer Based Training – CBT, mentioned by McArthur, Lewis & Bishay (2002), and Computer Assisted Instruction – CAI, referred by Beck, Stern & Haugsjaa (1996). According to these authors, the system mentioned by the first ones usually generated groups of problems projected to increase the student's performance in domains based in abilities, such as arithmetic and vocabulary regaining. Furthermore, the decisions on how the student should navigate through the material were based on decision trees and the questions and answer sequence were directed by the hits and errors of the student, their individual abilities not being taken under consideration. To Urban-Lurain (1998), the recommended proposals in theses systems limited themselves to present a problem to the student: register the answer and evaluate performance, but the instructions were not individualized to their needs.

Virvou & Moundridou (2009) used an Intelligent Tutoring System to develop an authorship tool for Web, destined to teachers and students in the domain of algebraic equations. When accessing the tool there is a "description" of a specific domain, given by a human teacher. The tool offers help to human teachers in exercise construction and, while the students resolve their exercises, "monitors" provide the suitable answers. It gathers intelligence in its component which executes diagnosis to the student's errors. It also controls the teaching material in a flexible and individualized way and executes intelligent analysis of the student's answers, providing interactive support to problem solving. Furthermore, the system adapts and takes note of the kind of connection the students make between their answers and the received support. The advantages of this system, according to Virvou & Moundridou (2000), is that human collaborating tutors can be in the same physical space or distant and the students can be in a classroom, in a specific location, or in a virtual class, which can be spread through many physical locations, promoting distance learning. Trying to answer the question: is it possible to achieve efficient learning environments from a generic tool, applicable to all knowledge contexts, throughout the entire period of learning? Ainsworth (2007) developed an authorship environment called REDEEM, which proposes to achieve such goal, adapting existing interactive learning environments of multimedia authorship software (or courseware). REDEEM was developed on the Click2Learn ToolBook instructor and it is executed on Windows platform. The software consists of two main parts (the authorship tool REDEEM and a shell), where the "authors" and the students can interact through. The existing courses on ToolBook are about, among others, genetic topics; information and communication systems; PCs and network management; html courses; statistics and photography basics, for example.

The first big task to be executed is describing the domain material, which the system will use to make decisions on how to present the material to students. The second biggest aspect of authorship domain is adding interactivity. The "authors" create questions (multiple choice, fill in the blank, true-false or combined questions) and provide the answers that explain to the student why an answer is right. Five different suggestions can be created for each question, which according to Ainsworth (2007) increases specificity. The "author" also describes a number of question characteristics which the shell uses to implement a new specific teaching strategy (difficulty, pre or post-test, for example). Reflection points can also be added (as means to make the student alert to make note) or about a non computer related task (which drives the student's attention to another activity). Ainsworth (2007) followed REDEEM for ten years, comparing it to other coursewares and describes it as one of the most usable and easiest to learn. He recognizes, however, that due to its generality, some of the benefits that come from the rich specific knowledge domain had to be sacrificed.

To Crispim, Abdalla Júnior & Molinaro (2002), a teaching cycle must meet the following requisites:

a. present information related to the teaching goal;
b. direct the students actions towards those goals;
c. value and evaluate the students actions;
d. provide feedback;

e. provide orientation strategies to the student;
f. conduct and motivate the entire teaching process.

With that in mind, they claim that most classical distance learning environments present limitations in the implementation of interaction between student and teaching environment. These authors developed an environment using artificial intelligence techniques, specifically Bayesians networks (which are based on the strength of their causal representation of human reasoning and the ability to represent and react to configuration changes); neural networks (that try to approximate the computer processing model to the one of the human brain, for possessing a similar interconnection level to the one of the human brain and, in a conventional modern computer, the information is transferred in specific times within a synchronized relationship); workflow (which makes easier the organization of knowledge that is to be transferred to the student through the development of a group of paths, focused in different levels of explanation and respective exercises for subject learning tests); and software agent or intelligent agents (due to its characteristics: ability to explore a significant sum of knowledge domain; tolerance to unexpected errors or resulting of wrong entries; ability in the use of symbols and abstractions; adaptation capacity and goal oriented behavior; ability to learn from the environment; real time operation capacity; and ability of communication in natural language). To Crispim, Abdalla Júnior & Molinaro (2002), the presented architecture corresponds to the inclusion of reasoning mechanisms in the learning environment model which will be responsible for the optimization of the construction process, knowledge transmission and intelligent learning.

An important step in that direction was taken by Barr, Cohen & Feigenbaum (1990) with the development of the Intelligent Tutorial Systems – ITS, which main goal, according to Clancey & Soloway (1990), was to reproduce the intelligent behavior (competent) of a human tutor, and can adapt its teaching methods to the students learning rhythm. In this context, according to Lévy (2000), the mail and the electronic conferences serve as intelligent tutors and may be placed at the cooperative learning device's service. The hypermedia supports (CD-ROM, online interactive multimedia database) allow fast and attractive intuitive access to great groups of information.

In this context, the multiagent systems have particular importance, for, according to Russel & Norvig (1995), an Agent is a system capable of perceiving through sensors the information of the environment in which it is inserted, acting and reacting through actuators. Ferber & Gasser (1991) reinforce this idea stating that an agent is a real or abstract entity, capable of acting on itself and on the environment, and has a representation of this environment, even if only partial. In a multiagent universe, it can communicate with other agents, which behavior derives from "its observations", from "its knowledge" and from the interaction with other agents. Bittencourt (2001) defends the idea that an "intelligent community" is the base of multiagent systems. This idea defines the system's intelligence as being the product of the "community's" social behavior which is made up of autonomous agents with cooperation capacity among themselves, with a common goal, which agents are inserted in the same environment. Gasser (1992) completes this reasoning claiming that an agent can be considered an entity capable of performing formal calculations and producing a certain number of actions, from knowledge and internal mechanism that are typical to them.

The multiagent system forms a sub-area of distributed AI and focus in the study of autonomous multiagents in a multiagent universe. To multiagent systems, the term autonomous means that the agents exist, independent of the existence of other agents. Usually, each agent has a group of behavioral capacities that define its competence, a group of goals and the necessary autonomy to use their behavioral capacities to reach their goals. To Álvares & Sichman (1997), an agent is a computational entity with autonomous behavior that allows it to make its own actions. The main idea in a multiagent system is that a global intelligent behavior can be reached from the individual behavior of the agents. In these systems, it is not necessary for each agent to be individually intelligent in order to reach a global intelligent behavior. The intelligence metaphor used by multiagent systems is the "multiagent community", that is, the social behavior, which is the base for the systems intelligence. The metaphor used by classic AI is basically of psychological origin, while the one used by DAI can be of sociological or ethological. A sociological/ethological approach is interesting when the goal is to resolve complex problems, that require knowledge from various domains and that can involve physically distributed data.

The multiagent systems are usually characterized didactically in two classes: *reactive* and *cognitive. Reactive* multiagent systems work with a great number of simple agents in the development of systems for the resolution of a determined problem. They are based in models of biological or ethological organization. Its functional model is formed by the binomial *stimulus-response (action-reaction)*. Cognitive multiagent systems work with few agents that perform much more complex tasks and are based in human social organizations such as groups, hierarchy and market, which characteristics are the base of this work.

This century has been branded by transition and by deep transformations, with extensive impacts in the educational process and pedagogical theories, so Azevedo (2000), Silva (2006) e Maia & Mattar (2008) say. The creation and dissemination of new technologies, with the multiplication of connected computers networks, advancement of interactive medias and other forms of content transmission, besides books, have taken the world to new experiences and interaction forms, specially in the teaching-learning field and professional formation. A new space of communication, organization and information socialization, knowledge and education is being inhabited. In this context, a new modality of teaching stands out – Distance Education, or Online Education, also called virtual formation environments. These resources allow people in different physical spaces to share, construct and trade information and knowledge, interactively, using various instruments or technological tools. This new modality of education allows flexibility and interactivity in the learning process, characteristics peculiar to the internet, instrument of construction of a communication culture, which methods demand new conducts.

These modern medias, in the case of Distance Education, allow both teachers and students, living anywhere in the world, to teach and learn, independent of where they live. There is flexibility to choose the best times and days for study, besides the student's possibility to even choose the rhythm that better adapts to their work, family or other situations. This form of education favors students that live in places distant from any educational institution, those who had to abandon studies by adversity or never even did study, to improve their knowledge. In a way, classroom instruction would not give them this opportunity, since it would require hundreds of schools and thousands of teachers, and additionally, a great volume of investment in infrastructure to achieve this task. Even so, the Brazilian legislation on Distance Education still imposes monitoring attendance that occurs when the teaching has the availability to go to communities where it is possible to gather a certain number of participant students.

This obstacle will be removed by a multimedia authoring tool, in multiagent systems that do not require human teachers, like the proposal of this work. The proposal of use of architectures in multiagent systems in Distance Education brings great advantages relating to traditional educational structures for presenting more flexibility in the treatment of elements that compose the system. The fact that Agents are used to model its components allows the grouping of the traditional structure or the expansion of each Module by the composing Agents.

Many applications have been developed, employing multiagent systems in the teaching and learning process, among which can be highlighted, for instance, the Multiagent System of Teaching and Learning on the Internet – SEMEAI. According to Geymer *et al*. (2003), this system's goal is to promote distance learning, using agent technology to adapt itself to each student's characteristics. That occurs by applying different forms of teaching that adapt to the psycho pedagogical aspect of the student, allowing effective retention in memory due to the diversity of forms and learning opportunities and its adaptation to the student. In this environment, the author directs the solution, highlighting three fundamental activities in a tutoring system:

a. adaptability to the student's profile, with the goal of providing more quality in the selection of available material;
b. automatic selection of adequate teaching strategies, with bases in good results attained by evaluations throughout the process. The environment can lead to the adoption of more suitable strategies to conduct the student's learning in a determined subject;
c. personalization of the teaching curriculum, through an specific agent, responsible for such activity. This personified character of the curriculum is defined as the selection of material according to established tactics in each teaching method that make up the selected strategy.

Fialho & Alves (2001), using Virtual Reality techniques associated to the Theory of Cognitive Agents, developed a teaching tool with the goal of helping child, teenager and adult to learn, interact and dive into the process of knowledge sharing. These authors encourage the development of softwares for Child Education, for example, with the intention of improving teaching quality. That allows, according to them, the software to be the support to knowledge construction on children, teenagers and adults, and not merely a synonymous of "automation" of education. This means the development of softwares must recover Brazilian values, in this case, aiming for the approximation of reality.

Santos & Osório (2003), trying to incentive the learning segment, presented an intelligent virtual three-dimensional environment adaptable to Distance Education. In this environment, used to make content available, the characteristic of adaptation is related to the possibility to reorganize them through process of insertion, removal or updating of information and personalization of the presentation of said content, according to the user's interests and preferences. For that, a content profile and a user profile are used in the adaptation process. Furthermore, the environment is inhabited by intelligent entities that act as the user's assistants during navigation and localization of relevant information, as well as helping with the organization of available content.

However, establishing these profiles is a complex task, which elements in general can't always be particularized, so they need to be considered under a group. For Morin (2004), complexity means a common texture that is presented as inseparably associated. This way, individual and environment, subject and object, order and chaos, professor and student, actions and life's "wefts" and other fabrics that conduct events need to be looked at from a general context. For this author, complex is what is weaved and must be considered as such. The separation in parts may cause the lost of context. This highlights the fact that knowledge is built by exchanges that nurture the relationship between subject and object, by dialogs, interactions, transformations and mutual enrichments, where nothing is linear or predetermined, but instead relational, undetermined, spontaneous, creative and new. Under that view, the technology that houses a teaching and learning system must hold these characteristics that imitate human behavior, gifted with these faculties.

According to Araújo (2007), the current teaching method does not support new tendencies on human subjectivity. In the author's opinion, there already is enough bases to a legitimate teaching transformation, knowing that one does not learn linearly, but instead moving the several dimensions of the human being. This moving will facilitate the reconfiguration of knowledge by being incorporated by students and will help find and attribute self sense to information native both from teachers and texts as well as from established dialogues in the teaching environment. It is from this process that emerges knowledge that allows the student to survive and transcend from a dialectical relationship between knowledge and action that happens through the joining of other human dimensions, including the sensorial, intuitive, emotional and rational which are known to be complementary.

In online environments based on the perspective of interactivity, the courses contents are built in a partnership process, where other authors create and socialize their knowledge in many ways: through softwares, interfaces, hypertexts and media. Araújo (2007) proposes the learning environment to be more interactive, with more student autonomy, with practices that contemplate different learning styles and others that stimulate the individual and collective knowledge building.

If, on the one hand, it has been difficult to produce an efficient authoring tool for multimedia or distance education, on the other hand, in the opinion of many researchers, including Kaskalis, Tzidamis & Margaritis (2007), finding a way or tool to evaluate the efficiency of the teaching process and learning of the proposed tools is still a question with no answer.

Oliver (2000) introduced a tool to evaluate the learning level obtained by experts and by groups, through shared procedure. The evaluation, according to this author, is a complex process and can be characterized as a resource through which people make judgments on value concepts. In the learning technology context, however, it becomes too simple, in his opinion. The tool proposed by Oliver (2000) offers an instrument to analyze this complexity, through the comparison of two cases (experts and by groups) afore mentioned, which concluded that there is a higher learning level on shared learning process. This model in the tool aims to give appropriate support to the management of an organization, in this case; promote professional training; and program time in distance learning processes, without hurting activities.

Kaskalis, Tzidamis & Margaritis (2007), aiming to find a tool that could serve a purely educational goal, evaluated 44 packages of authoring tools separately. They collected a reasonable amount of data which was treated as base in a five point scale, on which results the tools were evaluated. They reported the lack of data related to education, aiming to specify if it is possible for an authoring tool to be used by educators (with no particular ability) to facilitate a course in any field or subject. They could not find a software to clearly evaluate the educational part and finally concluded that, from the educational point of view, none of these tools had something special to offer to a learning environment.

To Collazos *et al.* (2007), many researches on collaborative work focus on the group the quality of the result as criteria for collaborative learning success. There are few researches on the collaborative process itself. Comprehending this process would not only help make it better, it would also help improve collaboration results. To these authors, understanding

and evaluating collaborative learning processes requires defined analysis of group interaction in goals and learning context. Taking relations between tasks, products and collaboration under consideration, Collazos *et al.* (2007) presented a group of measures projected to evaluate the collaborative learning process. These procedures emphasize: direction of the base system; measures based on data produced by the collaborative learning system during the collaboration process; and suggest that these measures can be widened, considering perceptions of participants in the process.
The model proposed by the authors is based on the size of the collaborative learning process. For that, they produced quality indicators for the student's work during the cooperative learning phases, from a model of high collaborative learning level.
In the context of these advances, multiagent systems, which components, according to Oliveira (1998), should reason about the actions and about the coordination process among itself, constitute more flexible architectures, in a system organization that is subjected to change, trying to adapt itself to environment and/or problem to be resolved variation. Under this view, the goal of this chapter is to present the proposal of an authoring tool, in a multiagent environment, as instrument for teaching and learning a second language and distance education, multimedia, of any discipline. A tool continually available on the Internet, so that whoever is interested can use it at any time and place, according to their potentialities, conveniences and possibilities, with no harm to other activities they may eventually have.

3. Material and methods

The tool to be implemented will be based on a multiagent system, according to the vision of Heilmann *et al.* (1995), which agents, with "cooperative spirit", working together, will execute mutually benefic tasks, independent of its complexities. These tasks are destined to serve students and teachers of foreign languages and distance education, of any discipline. A system of this nature is intended to act as representative of others, with the goal of performing actions that benefit the represented part.
The agents, as entities of software that form a group of operation in favor of the user of another program with some level of independence and autonomy, do that, according to O'Connor *et al.* (1996), employing some knowledge or representation of the users goals and desires. For these characteristics, they are, therefore, essential to a tool that incorporates this function of serving, of benefiting the user. Another aspect that justifies the option for a multiagent system to base the proposed tool are its priorities of intelligence (capacity to dynamically realize the environment, perform actions that affect the environment conditions, understanding perceptions, problem solving, reach conclusions and act); perception (ability to realize and execute a task with more efficiency than on previous executions); communicability (capacity to communicate with the user, with other software agents or other processes); cooperativity (capacity to work together in order to reach a general goal through mutual assistance); mobility (capacity to move by itself through the electronic net or moving to one location to another, preserving its internal state); veracity (high level of trust that the agent represents exactly the user); autonomy (capacity to operate by itself and execute actions with no user intervention); adaptability (capacity to process different task levels, accordingly to its competence); and many others, which advantages are widely discussed by many authors such as Gilbert & Manny (1996). Such proprieties allow implementation of flexible structures, which organization easily adapts to

environment or problem to be solved variation; this is the reason to its application in so many areas of knowledge and activity.

By allowing the representation of some reasoning abilities and specialist knowledge in the search for the solution of big problems in this area, these characteristics favor application to school activity and contribute with new teaching and learning methods. They also give the user the possibility to build new experiences and elaborate or create new mental models, as advised by Piaget (1983). In the specific case of teaching and learning foreign language binominal, the anthropomorphism of the agents has particular importance in representing human characteristics such as sounds, tone of voice, and others, so useful to the adequate expression of words, for instance.

The most relevant in the context of distance education of any discipline or foreign language is that multiagents systems constitute, one could say, a shared environment and, as such, are reflected on the growth and quality of knowledge, which growth happens as it is shared, as opposed to the capital of the industrial era, for example, that tended to grow as it was retained by its owner. So, there is the perspective that this character is maintained in the foreign language and other disciplines learning, with the use of this tool.

4. Proposal

The multimedia authoring tool for distance education, of any discipline or foreign language, based on multiagent systems, will give the student, independent of the knowledge level of the language discipline, more flexibility on the learning system, allowing total control over the process. The tool allows the development team of multimedia educational material to easily insert texts and objects such as figures, sounds and video, to create stories with dialogues, based on the didactic material used for the teaching of a foreign language or other discipline.

A multiagent architecture tries to assimilate the cooperation between people on human society to perform tasks within the tool and is transmitted by communication and message exchange between agents, aiming to execute determined activities. Based on the structure proposed in the tool, the teaching of foreign language will have its applicability contemplated in this methodology both for students and for the general work market. In the proposal presented here, based on a multiagent architecture, a specific function is attributed to each agent. To the performance of tasks associated to the role they have, the agents will implement specific cognitive abilities, with full communication and interaction between them, providing total functionality to the system. Unlike a centralized solution based solely in one base of knowledge, a structure distributed for knowledge is proposed here, as suggested by Rossetti *et al.* (2006). Thus, each agent will have its own base of knowledge, with the required representations for better treatment of the information it will use in the performance of its tasks and on the sharing and cooperation with other system agents, as seen on the conceptual framework of the diagram in Figure 1:

In this diagram, the rectangles represent the agents; in the circles are the Knowledge's Base – KB of each agent's domain; the arrows represent the relations (communication) between agents; the "stick dolls" are real actors, among which, teacher and students are the main ones. The tool will be available and accessed through the Internet, in the Web, with no direct human interaction between teacher and student. The interaction between them will happen indirectly, through the agents Author and Learner, which represent them in the teaching environment, and serve as mediators in the process of exposition and learning of the content. In this context, the author agent, which represents the teacher on the system, will have the knowledge over the domain of application, which will allow the elaboration of content to the ministered lessons, using multimedia resources. These lessons, elaborated with the support of base-texts, will contain information about the knowledge domain over the foreign language to be ministered in various formats such as text, sound and image. Learning situation will be created toward hearing perception (pronunciation) and for the correct spelling of words (writing), in the foreign language case. It will be, therefore, based on the listening of texts with visual tracking of scenes for contextualization. In this process, the student must write, in an indicated field, the text of language of study, relative to what was heard and the system will verify the level of success obtained (total, partial or unsuccessful). The system immediately interacts giving feedback until total success is reached.

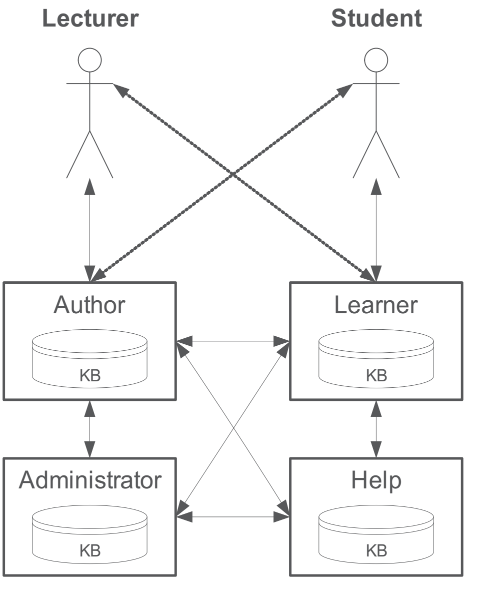


Fig. 1. Conceptual framework of multiagent environment for the teaching of foreign language or other disciplines

As an illustration of this interactive environment, consider, for example, that a dialogue in a foreign language has been created and recorded by the tool, which images are displayed to the student, offering conditions to contextualize the dialogue. The written text will not be presented. The student must listen to the dialogue and reach the conclusion that what was heard was understood, write what was understood in the referred field. The system will then verify if the student:

a. understood and was able to correctly reproduce the text, considering the exercise finished;
b. understood, but made small spelling errors when writing the text. In this case, the correct spelling is presented so that the student acknowledges it, considering the exercise finished;
c. did not understand. In these circumstances, the system offers the student two options: (i) suggest that he access “clues” in the system; (ii) follow a video displaying the

phrases being pronounced slowly to facilitate comprehension. Then, the exercise is repeated so that the student consolidates the understanding, after which it is considered finished.

Furthermore, the tool will allow the development of additional didactic material for the classes, allowing students to exercise and develop, on their own rhythm, pronunciation, hearing perception and orthographic abilities. Exercises relating to those classes will also be created from image association, recorded dialogues and its corresponding texts. To each exercise will be associated a group of "clues", that can be chosen in new texts, sounds, images or videos. The clues are intended to be associated with the lesson as a whole and/or to specific words within each dialogue, so that, if the student doesn't know how to spell one of the words for not understanding its pronunciation, there will be a clue relating to that word. For example, if in the basic module of an English course the heard text was "this is a pencil" and the student could not make out the last word and wrote "this is a pen", the system will present the image of a pencil, at the same time as it pronounces the word "pencil".

The author agent is responsible for the creation of situations involving stories with images and recorded dialogues, which the student will use in the course. Using multimedia resources (microphone, video camera, videocassette, etc.), the author will develop an attractive course, making the learners and consequently the students feel motivated.

The student is represented in the system by the learner agent. So, each student that uses the system will have a learner agent to represent them. The learner assimilates the student's learning capacities, building a profile. Knowledge of the student's abilities will help the author suggest different forms to minister the subject's content in order to increase its efficiency and effectiveness. On the other hand, it can also serve the teacher, as a way of testing new methods or exercises. Each learner agent will know the learning profile of the student it represents. Through it, the student will enter classes that are available, with access to images and sounds, but not to text dialogues. In fact, the student's main activity is to reproduce the hidden text after listening once or twice to the sound of recorded dialogue, using the clues or not. The tool will make available a dialogue box with several indications to answer presented situations which will allow full interaction between the student and the system. The information in the dialogue's sequence will be transmitted through a window. In this phase, an algorithm will be used as comparison criteria which will count the amount of written words. If that number is different from the goal number, an important mistake has occurred. If, on the other hand, the number of words coincides and the spelling errors correspond to similar phonemes, such as *ph* and *f*, only a "slight" mistake occurred. Otherwise, it will be important mistakes. The system will not present the correct answer while the student makes important mistakes. The options will be repeating the text, until hearing allows the identification of words and clues. This practice coincides with the one normally used by a human teacher on a classroom. When the student doesn't make any mistakes or only "slight" mistakes, then the system will consider the lesson's goals as achieved and will present the correct spelling of the words to the student.

In the case of distance education, consider, for illustration purposes, that in a Mathematics class, in which the approached subject is a first degree equation, for example, where the following equation is displayed to the student:

$$3x - 4 = 2x + 5$$

The student must solve the equation and after obtaining the answer, enter it on the indicated field in the text. Similar to the case of foreign language, the system will verify if student understands applying, in order, the three steps used for such, with the adaptation convenient to the characteristics of this discipline. It will also be used, in this case, as for any other discipline, as comparison criteria, an algorithm, not as a word counter, but as checker of solution, in the case of Mathematics, of local, in the case of Geography, of date, in the case of History, and so on.

The administrator agent manages the system and, therefore, must interact with all of the present agents. As an external observer of the author agent's activities and of the learner agent's evolution, the administrator can serve as a "critic" of the adopted didactic methodologies. This information, allied to the feedback given by the learners to the author, contributes to teaching methods and techniques more oriented to the goal of language teaching or other relevant disciplines. The knowledge associated to it is related to the systems administrative tasks such as configuration and performance statistics. It's through it and through their registration that the user has access to the tool. In this moment, besides welcoming, the administrator will ask the candidate some basic information, such as:

a. which foreign language course do you wish to attend or what language do you wish to learn? Or which discipline do you wish to attend?
b. what is you level of knowledge of the language (starter, medium, advanced, conversation, other)? Or of the discipline of interest. If the students answer "starter", it gives them an explanation of the course and how the system works; applies an exercise that allows access to material of classes in their level; and inserts them on the program. If the answer is not starter, they are invited to take a test to determine in which level they should start the intended course. Next, they get instructions about the course and how the system works; they are then invited to an exercise that allows access to class material, and then finally are informed that they are apt to use the system and are inserted in the program at the recommended level.

The agent will be available to the learner to help in the performance of lessons. Furthermore, it will identify the student's main difficulties by categorizing frequent doubts. This information can also contribute to the improvement of resource presentation under the author agent. On the other hand, the help agent itself can adapt, developing "tips" more oriented to the student's needs. The help agent will have knowledge associated to the more efficient form to help the system's agents, learners in particular.

The relationship of these four agent structures will allow knowledge exchange whenever one needs information from the other, will be object of definition during the software's elaboration and of the tool's implementation. When teaching a foreign language, each course (language) will be ministered in three modules: basic, intermediate and advanced, which will be detailed during the elaboration of the software and its respective implantation. Other disciplines will obey levels established by the official teaching system and respective Brazilian legislation, likewise, established during the confection of the software and implantation of the tool.

The tool will have a periodic evaluation system to test its efficiency, through user satisfaction, according to the users learning in the respective discipline. This evaluation, in the case of language teaching, will contemplate the pedagogical resources text, sound, image etc. and the use of the system. In any case, it will be based on indicators related to collaborative learning process, according to Collazos *et al.* (2007), properly adapted to such

reality. For this effect, students will answer questions, which answers and suggestions, after being compiled, will serve as grounds to implement improvements in the system. This implies the use of a well defined and efficient communication protocol, with clarification of transmitted information resources, where ontologies, which have the role of facilitating the construction of a domain model in the Knowledge Engineering process, will be used. Besides, ontology will provide a vocabulary of terms and relations to which a domain can be modeled, therefore, will be used to guide the extraction and integration of relevant information from various structured sources. This will be used to fill the void between information conceptualization, both from the user's point of view, as from the ones offered by different information sources. This multimedia character of representation of information meets the need of application of different teaching methods that can vary according to the subject to the language of study or to the student's learning abilities.
The tool will be implemented by a team constituted of professionals of the information and communication technology area, professionals from the knowledge engineering and management area, informatics and language teachers, initially English and French, the first ones to be implemented. When it comes to implementing any other discipline as a distance education instrument, for example, besides these professionals, teachers of the respective disciplines to be included in the system will also participate. The participation of teachers of each discipline to be linked to the tool is indispensable to the efficiency and effectiveness in the teaching and learning process.

5. Final considerations

Immediately after the availability of the tool on the Web, there will be, with any language or discipline, a period of "validation" by the user of about six to twelve months, which will be monitored by the implementation team. During this period, in the moment of "registration", the student will be informed of this situation and invited to collaborate in the process. For that, as the tool is being used, a form will be filled, which will be available on the system, where the student is to write in any criticism, observations or suggestions to improve the system. This material is automatically sent to the team, which will analyze it and implement in the tool necessary adjustments, looking to optimize it.
The tool will offer the advantages of a foreign language teaching system, initially, English and French, permanently available on the Internet and of other general knowledge subjects. This will create the opportunity for a great number of people to study a foreign language or other disciplines, according to their conveniences, time and available hours, anywhere there is a computer connected to the Internet. This advantage, besides providing the opportunity of study to a great number of people, will favor a possible social policy of acquisition of community computers available for people of low income, especially, constituting a qualitative social cultural leap on the subject of general teaching and learning of a foreign language in particular, specifically in Brazil.

6. Conclusions

Multiagent systems offer consistent conditions to serve as base to construction and implementation of the proposed tool.
A tool to be implemented on a multiagents systems base will constitute a great advance on the teaching and learning of foreign languages or other disciplines binominal, in the

Distance Education context, once communication and cooperation operated by agents are similar to the ones that occur between people in human society to perform certain activities.
This authoring tool, continually available on the Internet, will present the flexibility that traditional courses usually don't have, particularly the foreign language ones. Besides the advantage of starting a course in whatever period, place and time, it will offer the possibility to restart it in any period in case an interruption occurs.
The tool will make possible the access to foreign language or other discipline courses, in the case of Distance Education, to anyone, at any time or place that has a computer connected to the Internet.
People who learn easily and have the time to progress faster on the learning of foreign languages or to improve their knowledge via Distance Education, will benefit, since they will learn the language or discipline of interest in shorter periods.
The tool to be implemented can be, in the future, a product used and disseminated by public schools, universities and companies.
The validation done by the system's users and the periodical evaluation to each it will be submitted will allow adjustments which will conceive its constant efficiency.

7. References

Ainsworth, S. (2007). Using a single authoring environment across the lifespan of learning, *Educational Technology & Society*, Vol. 10 (N°. 3): 22-31.

Álvares, L. O. & Sichman, J. (1997). Introdução aos Sistemas Multiagentes, *Anais da 16° Jornada de Atualização em Informática*, Brasília-DF, pp. 1-38.

Araújo, M. M. S. de. (2007). O pensamento complexo: desafios emergentes para a educação on-line. *Revista Brasileira de Educação*, Vol. 12 (N°. 36): 515-551.

Azevedo, W. (2000). Muito além do jardim da infância: o desafio do preparo de alunos e professores on-line. Conect@ N° 2. Disponível em: <http://www.revistaconecta.com/conectados/wilson_muito_alem.htm>. Acesso em: 30 jul. 2010.

Bailey, B. P. & Konstan, J. A. (2000). Authoring interactive media. In: Webster, J. G. (Ed.), *Encyclopedia of electrical and electronics engineering*, John Wiley & Sons, New York.

Barr, A.; Cohen, P. R. & Feigenbaum, E. A. (1990). *The handbook of artificial intelligence*, Vol. 1, Addison-Wesley, Los Altos-California.

Beck, J.; Stern, M. & Haugsjaa, E. P. (1996). Applications of AI in education. The ACM Digital Library, Vol. 3 (N°.1): 11-15. Disponível em: <http://www.acm.org/crossroads/xrds3-1/aied.html>. Acesso em: 3 set. 2007.

Bittencourt, G. (2001). Inteligência Artificial Ferramentas e Teorias. Editora da UFSC, Florianópolis.

Brasil. (1996). Lei n. 9.394, de 20 de dezembro de 1996. Estabelece as Diretrizes e Bases da educação nacional. Legislação, Brasília, DF. Disponível em: <http://www.planalto.gov.br/ccivil_03/LEIS/l9394.htm>. Acesso em: 18 nov. 2003.

Bulterman, D. C. A. & Hardman, L. (2005). Structured multimedia authoring, ACM Transactions on Multimedia Computing, *Communications and Applications*, Vol. 1 (N°. 1): 89-109.

Clancey, W. J. & Soloway, E. (1990). Artificial intelligence and learning environments, *Artificial Intelligence*, Vol. 1 (N°. 42): 1-6.

Collazos, C. A. et al. (2007). Evaluating collaborative learning processes using system-based measurement, *Educational Technology & Society*, Vol. 10 (N°. 3): 257-274.

Courtillon, J. & Raillard, S. A. (1983). Français langue étrangere: cours credif, Didier, Paris.

Coutaz, J. (1990). Interfaces homme-ordinateur: conception et réalisation, Bordas, Paris.

Crispim, H. A. F.; Abdalla JR, H. & Molinaro, L. F. R. (2002). Proposta para um ambiente inteligente de ensino a distância, *Anais do 9° Congresso internacional de educação a distância, Associação Brasileira de Educação a Distância*, Vila Mariana-São Paulo, pp. 36-53. Disponível em: <http://www.abed.org.br/congresso2002/trabalhos/texto48.htm>. Acesso em 24 out. 2007.

Ferber, J. & Gasser, L. (1991). Intelligence artificielle distribuée, *Proceedings of 10th International Workshop on Expert Systems & Their Applications*, Cours N°. 9, Avignon-France, [s.n].

Fialho, F. A. P. & Alves A. G. (2001). Agentes cognitivos na educação, *Anais do 1° Congresso Brasileiro de Computação*, Vol. 1, Sociedade Brasileira de Computação, Itajaí-SC, pp. 173-183.

Gasser, L. (1992). Boundaries, identity and aggregation: plurality issues in multiagent systems, Werner.

Geymer, C. F. R. et al. (2003). SistEma Multiagente de Ensino e Aprendizagem na Internet (SEMEAI). Disponível em: <http://www.inf.ufes.br/~sbie2001/figuras/artigos/a068/a068.htm>. Acesso em: 2 set. 2007.

Gilbert, D. A. & Manny, A. (1996). Intelligent Agent Strategic. Disponível em: <http://activist.gpl.ibm.com:81/WhitePaper/ptc2.html>. Acesso em: 4 set. 2007.

Heilmann, K. et al. (1995). Intelligent agents: a technology and business application analysis. Disponível em: <http://haas.berkeley.edu/~heilmann/agents/index.tml>. Acesso em: 13 set. 2007.

Kaskalis, T. H.; Tzidamis, T. D. & Margaritis, K. (2007). Multimedia authoring tools: the quest for an educational package, *Educational Technology & Society*, Vol. 10 (N°. 3): 135-162.

Lévy, P. Cibercultura. (2000). Editora 34, São Paulo.

Maia, C. & Mattar, J. (2008). ABC da EaD – A educação a distância hoje, Pearson, São Paulo.

McArthur, D. J.; Lewis, M. & Bishay, M. (2002). The roles of artificial intelligence in education: current progress and future prospects, Rand Corporation, Santa Monica.

Moran, J. M. (2003). Contribuições para uma pedagogia da educação on-line. In: Silva, M. (Org.). *Educação on-line*, Loyola, São Paulo.

Morin, E. (2004). Os sete saberes necessários à educação do futuro, Cortez, São Paulo.

O'connor, D. O. et al. (1996). Intelligent agents strategy: white paper, IBM Corporation, Research Triangle Park, North Caroline. Disponível em: <http://activist.gpl.ibm.com/WhitePaper/ptc2.html>. Acesso em: 6 set. 2007.

Oliveira, F. M. de. (1998). Advances in artificial intelligence. In: *Proceedings of 14° Brazilian Symposium on Artificial Intelligence, Sociedade Brasileira de Inteligência Artificial*, Porto Alegre. Porto Alegre, pp. 54-73.

Oliver, M. (2000). An introduction to the evaluation of learning technology. *Educational Technology & Society*, Vol. 4 (N°. 3): 20-30.

Pettenati, M. C. et al. (2000). The learning tutor: a web based authoring system to support distance tutoring. *Educational Technology & Society*, Vol. 3 (N°. 4): 151-164.

Piaget, J. (1983). Psicologia da inteligência, Zahar, Rio de Janeiro.

Ramos, E. M. F. (1996). Educação e informática: reflexões básicas. Disponível em: <http://www.edit.inf.ufsc.br:1998/album99/referencias/Q1.htm>. Acesso em: 9 set. 2007.

Rossetti, A. G. et al. (2009). Ferramenta de autoria multimídia para ensino de língua estrangeira em ambiente multiagente. *Educação e Pesquisa, São Paulo*, vol. 35 (N°.1): 81-97.

Rossetti, A. G. et al. (2006). Proposta de um ambiente multiagente para ensino de língua estrangeira multimídia, *Anais da 3ª Conferência Sulamericana em Ciência e Tecnologia Aplicada ao Governo Eletrônico*, Digital Ijuris, Curitiba, p. 47-54, DVD.

Russell, S. & Norvig, P. (1995). Artificial Intelligence: a Modern Approach, Prentice Hall, Series in Artificial Intelligence, New Jersey.

Santos, C. T. dos & Osório, F. S. (2003). Um ambiente virtual inteligente e adaptativo para a educação a distância, *Anais do 14° Simpósio Brasileiro de Informática na Educação*, NCE - IM/UFRJ, Rio de Janeiro.

Silva, M. (2006). Educação online, Edições Loyola, São Paulo.

Urban-Lurain, M. (1998). Intelligent tutoring systems: an historic review in the context of the development of artificial intelligence and educational psychology. Disponível em: <http://www.cse.msu.edu/~urban/ITS.htm>. Acesso em: 13 set. 2007.

Virvou, M. & Moundridou, M. (2000). A Web-based authoring too for algebra-related intelligent tutoring systems. *Educational Technology & Society*, Vol. 3 (Nº. 2): 61-70.

Widdowson, H. G. (1991). O ensino de línguas para comunicação, Pontes, Campinas.

18

Image Processing on MR Brain Images Using Social Spiders

Moussa Richard, Beurton-Aimar Marie and Desbarats Pascal
LABRI - University of Bordeaux
France

1. Introduction

Image segmentation is one of the most difficult tasks for digital image processing. This is an important step for quantitative analysis of images as brain images and so for studying many brain disorders. Indeed, structural changes in the brain can be the result of brain disorders. The quantification of these changes, by measuring area of regions of interest, can be used to characterize disease severity or evolution. The classical way of manual marking out of cerebral structures in MRI[1] images by an expert is obviously a time consuming process. Moreover, these manual segmentations are prone to large intra/inter-observer variability.
A lot of researchers have spent many years trying to solve this problem and proposed lots of methods. Unfortunately, these methods are dedicated to particular solutions. There is no generic method for solving the image segmentation problem. One main difficulty comes from that two types of noise are presented in medical images: physical noise due to the acquisition system, for example, Optical, X-rays and MRI, and physiological noise due to the patient status. Thus, the segmentation remains a challenging task. Image segmentation algorithms subdivide images into their constituent regions, with the level of subdivision depending on the problem to be solved. Robust, automatic image segmentation requires the incorporation and efficient utilization of global contextual knowledge. However, the variability of the background, the versatile properties of the target partitions that characterize themselves and the presence of noise make it difficult to accomplish this task. Considering the complexity, it is often required different methods in the segmentation process according to the nature of the images. In this work, we propose to use a MAS[2] to realize image segmentation and particularly, a MAS based on social agents.
A MAS is composed of heterogeneous unembodied agents carrying out explicitly assigned tasks, and communicating via symbols. On the contrary, many extremely competent natural collective systems of multiple agents (e.g. social spiders and social ants) are not knowledge based, and are predominantly homogeneous and embodied; agents have no explicit task assignment, and do not communicate symbolically. A common method of control used in such collective systems is stigmergy, the production of a certain behaviour in agents

[1] Magnetic Resonance Imaging.
[2] Multi-Agent System.

as a consequence of the effects produced in the local environment by previous behaviour (Wooldridge, 2002).
Ramos and Almeida have explored the idea of using a digital image as an environment for artificial ant colonies (Ramos & Almeida, 2000). They observed that artificial ant colonies could react and adapt appropriately their behaviour to any type of digital habitat. Ramos et al. investigated ant colonies based data clustering and developed an ant colony clustering algorithm which he applied to a digital image retrieval problem. By doing so, they were able to perform retrieval and classification successfully on images of marble samples (Ramos et al., 2002). Liu and Tang have conducted similar works and have presented an algorithm for grayscale image segmentation using behaviour-based agents that self reproduce in areas of interest (Liu & Tang, 1999). Hemernehl et al. have shown how an intelligent corpus callosum agent, which takes the form of a worm, can deal with noise, incomplete edges, enormous anatomical variation, and occlusion in order to segment and label the corpus callosum in 2D mid-sagittal images slices in the brain (Hemernehl et al., 2001). Bourjot et al. have explored the idea of using social spiders as a behaviour to detect the regions of the image. The principle is to weave a web over the image by fixing silks between voxels (Bourjot et al., 2003) .
In the goal to provide a more generic method not dedicated to a specific noise, we have adapted a Multi-Agent model based on Bourjot's idea for implementing a new way of performing an image segmentation.

2. Context

The MAS is a distributed system composed of a group of agents, which interact between them through an environment. Agents are classified in two categories: cognitive and reactive. Cognitive agents have a global view of the environment, they know the task for which they work. Reactive agents only know a restricted part of their environment, they react to environmental stimuli and can modify this environment by adding or removing informations. But reactive agents do not communicate directly together and do not know the complex task for which they work: they have a restricted set of simple features and they only apply them (Chevrier, 2002).
In biology, many natural systems composed of autonomous individuals exhibit abilities to perform skilled complex tasks without overall control. They can be adapted to their environment to survive or to improve the collective functioning. This is the case of social insects colonies (Camazine et al., 2001) such as termites, ants (G. Theraulaz & Spitz, 1997) or spiders (Gleizes & Marcenac, 1999) which are actually an evidence of remarkable abilities to perform collaborative tasks such as: construction of complex nests, bridge construction, efficiently resources research and capturing prey. The study of collective movements of migratory flocks of birds or fish stocks also shows that the collective task is the result of interactions between autonomous individuals (Theraulaz & Spitz, 1997a). The immune system is also a representation of a complex system operation composed of a set of autonomous agents (Ballet et al., 2004).
Insects colonies like spiders ones are groups of reactive agents: each one knows locally what it has to do, but no one knows the more complex task for which they work. Such insects are called social agents. Two behaviours can be found in social agents system: cooperation and competition. Agents can cooperate to perform their task through the environment. They can also be in competition with other agents to be the first to perform their own task.

Social spiders belong to natural spider species whose individuals form relatively long-lasting aggregations. Whereas most spiders are solitary and even aggressive toward con-specifics, hundreds of species show a tendency to live in groups and to develop collaborations between each other, often referred to as colonies. For example, as shown in figure 1, such spiders of 5mm in body length are capable to fix silks up to a volume of $100m^3$ (Jackson, 2007). This technique is used to trap big size preys.

Fig. 1. Social spiders web silking.

Social spiders have been defined by the biologists to present stigmergic process (Theraulaz & Spitz, 1997b). The characteristics of these societies and the importance of the silk in the various behaviour have created a different model from the social insects one. During their artificial cycle, social spiders have the abilities to perform several actions as: `Fix A Silk`, to `Move Forward` and to `Move Backward`. This model have characteristics which sufficiently distinguishes the levels of the realized spots, the society organization and the communication supports. Indeed, social spiders correspond to an interesting model for three reasons (Chevrier, 2002):

1. Social spiders do not present any specialization in morphology and ethology;
2. An isolated social spider presents behavioural characteristics very close to lonely species;
3. Social spiders show spectacular organization and cooperation forms, in particular, the web construction and the prey capture or its transportation phenomenon.

These are the reasons why this model can be easily implemented in computer programs and used to achieve complex tasks. Before presenting the social spiders segmentation method, we will explore the commonly used approaches to segment images.

3. Image segmentation

On the first hand, we will present the different categories of image segmentation methods and particularly the two ones used for the evaluation tests: Region Growing (Shapiro & Stockman, 2001) which is a region-based method and Otsu (Chen et al., 2001) which is a voxel-based method in order to compare them with our method. Then, on the second hand, we will discuss the noise problem when attempting to segment an image.

3.1 Classical approaches

Image segmentation consists on partitioning an image into a set of regions that covers it. After this process, each voxel is affected to a region and each region corresponds to a part of the image. The discontinuity between the regions constructs the contour of the object. The segmentation approaches can be divided into three major classes (Pham et al., 2000). The first one corresponds to voxel-based methods which only use the gray values of the individual voxels. The second one is the edge-based methods detect edges, for example, this can be done by computing a luminancy function. The last one, the region-based methods which analyze the gray values in larger areas for detecting regions having homogeneous characteristics, criteria or similitude. Finally, the common limitation of all these approaches is that they are based only on local information. Sometimes, a part of the information is necessary. Voxel-based techniques do not consider the local neighbourhood. Edge-based techniques look only for discontinuities, while region-based techniques only analyze homogeneous regions.

3.1.1 Region Growing method

The *Region Growing* method consists on building a region from one chosen voxel and then adding recursively neighbours whose grayscale difference with the original voxel is below a threshold (Shapiro & Stockman, 2001).
This method tries to grow an initial region by adding to this region the connected voxels that do not belong to any region. These voxels are the neighbourhood voxels already in the region and whose grayscale is sufficiently close to the area. When it is not possible to add voxels, we create a new region with a voxel that has not been selected yet, then we grow the region.
The method ends when all the voxels were chosen by a region.

3.1.2 Otsu method

Otsu has developed a multi-level thresholding method (Otsu, 1979) . Its aim is to determine, for a given number of regions, the optimum values of different thresholds based on the variance of subdivisions created.
The basic method consists on separating the foreground from the background. In this case, we search the optimal threshold to split the voxels in two regions. For a threshold t, it is possible to compute *the between-class variance* σ^2(t). This measure is derived from the average intensity μ_1, μ_2 and μ of Regions [0; t], [t + 1; L] and [0; L] where L is the maximum intensity.
The Equation 1 introduce the computation of σ^2, where w_1 and w_2 represent the proportion of voxels in the class [0; t] and [t + 1; L] compared to the total number of voxels.

$$\sigma^2(t) = w_1(t)(\mu_1(t) - \mu)^2 + w_2(t)(\mu_2(t) - \mu)^2 \tag{1}$$

The Otsu method shows that the optimal threshold t^* is obtained for a between-class variance. The method consists on computing the variance for all possible thresholds (t ∈ {1; . . .; L - 1}) and determining its maximal value.
This method could be extended easily to the computation of M Regions with M - 1 thresholds $\{t_1; t_2; \ldots; T_{M-1} - 1\}$ $(t_1 < t_2 < \ldots < t_{M-1})$. The between-class variance is defined then as follows:

$$\sigma^2(t_1, \ldots, t_{M-1}) = \sum_{M-1}^{M} w_k(\mu_k - \mu)^2 \tag{2}$$

where w_k represent the proportion of voxels in the class $[t_{k-1}; t_k]$[3], μ_k the intensity average of this same Region and μ the intensity average of the class [0; L].
For each M-1-uplet, we compute thresholds of the between-class variance. The optimal thresholds, $(t^*, \ldots, t^*_{M-1})$, correspond to the maximum value of the between-class variance.
Chen et al. propose an algorithm that minimizes the number of necessary computation to obtain a faster algorithm (Chen et al., 2001). This method had been implemented for our evaluations tests.

3.2 Noise problem

Noise in biological images is still a major issue for the segmentation process. Two major contributions exist to explain medical imaging noise: the acquisition noise and the physiological noise. The acquisition noise can be identified as and sometimes reduced to physical noise meaning that it depends on the physical method of acquisition (Hendee & Ritenour, 2002; Vasilescu, 2005), for example, MRI noise caused by MRI machines. For the physiological noise, there are three main contributions : the respiratory cycle, the cardiac cycle and the scattered noise mainly due to physiological liquid action. Some solutions have been proposed intrinsically combined with the acquisition process. Another way is to use external monitoring to keep track of the cycles themselves and to reconstruct the cycle mainly using low order polynomial function. This function could be used to aggregate liquid noise regions into one in the post-treatment process (Bankman, 2000). Medical images exhibit all these types of noise characteristics. These characteristics need to be taken into account for all image processing. One important image processing operation is image segmentation. Image segmentation is often the starting point for other processes, including registration, shape analysis, motion detection, visualization, quantitative estimations of linear distances, areas and volumes. In these cases, segmentation involves categorizing voxels into object regions based on their local intensity, spatial location, neighbourhood, or shape characteristics. But, methods are often optimized to deal with specific medical imaging modalities such as MRI, or modelled to segment specific anatomic structures such as the brain, the liver and the vascular system. It is clear that a single segmentation technique is not able of yielding acceptable results for all different types of medical images. Still, the simplest way to obtain good segmentation results is segmentation by hand. This segmentation requires expertise in the field, in our case, a doctor is the person concerned. However due to the huge number of images to treat and the complexity of 3D images, it is not acceptable to consider that segmentation by manual contour tracing is an efficient solution. So, automatic image segmentation appears to be the right way to perform this task.

4. Social spiders model

The multi-agent system is composed of an environment and a set of agents. For segmentation purpose, the environment is created from a given grayscale image: it is a matrix of gray voxels. System and agents have a life cycle. Figure 2 shows that a cycle of the system consists in executing the three behaviours of each agent. This life cycle is transposed to a step. The number of steps to be executed is given by the user. Algorithm 1 presents a description of figure 2 where for a given number of time steps, each spider computes its life cycle. Its

[3] $t_0 = 0$ and $t_M = L$.

complexity is about $O(Steps.N)$, where $Steps$ is the number of time steps and N the number of spiders.

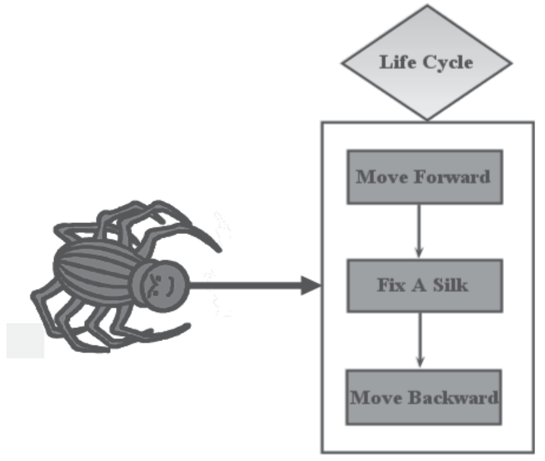


Fig. 2. System overview.

Algorithm 1 Segmentation method

Require: Voxels: Matrix of Voxels $\in \mathbb{N}^3$, Steps $\in \mathbb{N}$ and Conf: Configuration parameters
Create colonies and spiders from the Conf.
while Steps - - > 0 **do**
  for Each spider S **do**
    MoveForward(S).
    FixASilk(S, Voxels).
    MoveBackward(S).
  end for
end while

4.1 Environnement

The environment is composed of gray voxels. Each voxel is a position for spiders and allows them to access to other voxels. For a spider, the neighborhood of a voxel V which is coloured in blue in figure 3 is defined in two ways:

- The voxels around V, which are named *Local Neighbourhood* coloured in gray except those in green;
- And all voxels linked to V by a silk, which are named *Silked Neighbourhood* coloured in green.

Set of all reachable voxels are named *Access*.

4.1.1 Spider agent

As previously mentioned, spiders are reactive agents. They are defined by an internal state composed of a set of parameters values, a current position and the last voxel where a spider has silked. These spiders have three behaviour abilities:

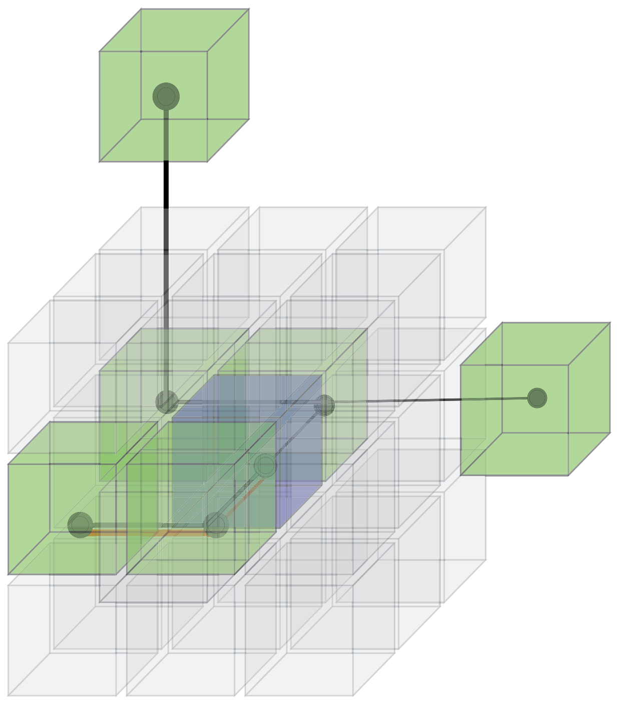


Fig. 3. Neighbourhood of a voxel.

1. Move Forward;
2. Fix a Silk[4];
3. Move Backward[5].

Spiders are grouped in a set called a colony. Spiders of a same colony share the same set of parameters values. Their aim is to detect the same region. For doing that, they communicate locally via silks.
An agent life cycle, as shown in figure 2, consists in firing each behavioural item according to a probability which depends of parameters values and environmental characteristics at the spider position.

4.1.1.1 Move forward

Moving Forward is computed according the way where there are several colonies trying to detect multiple regions. Each spider reaches one neighbour and reacts on it as described in figure 3. A weight function is defined to compute the probabilities of the *Access* voxels in order to select the best voxel to move to. So, for each voxel V in the neighborhood, the probability to move to V is as below:

$$P(Move(V)) = \frac{W(V)}{\sum_{a \in Access} w(a)} \tag{3}$$

[4] Weave a dragline between two voxels.
[5] Return to the last fixed voxel.

$w(V)$ is constant if V is in the *Local Neighbourhood*, else if V is in the *Silked Neighbourhood*, we distinguish the drag-lines woven by its colony and those woven by other colonies. Two simulation parameters, *attractself* and *attractother*, are used to compute this probability. These parameters are respectively the attraction for drag-lines woven by its colony and drag-lines woven by other colonies. A function F is also used to express drag-lines-counts influence on weight. This function express the saturation of the voxel bounded by a parameter called $saturationvalue$. So, for a voxel V in the *Silked Neighbourhood*, we have the following weight :

$$\begin{aligned} self &= attractself \; . \; F(draglines_{self}) \\ other &= attractother \; . \; F(draglines_{other}) \\ W(V) &= self \; + \; other \end{aligned} \tag{4}$$

Algorithm 2 describes the movement function by affecting to the weights of the neighbourhood voxels whether the weight of the colony of the spider whether its weight function and finally move the spider to a voxel from the neighbourhood according to its weight value. Its complexity is $O(Nei)$ where Nei is the maximum of neighbourhood of a voxel.

Algorithm 2 Move Forward

Require: S: Spider, Voxels: Matrix of Voxels $\in \mathbb{N}^3$.

```
weights[Size(Access(position(S)))]: weights of the neighbourhood voxels.
for i ∈ {0, . . ., Size(Access(position(s)) - 1)} do
    weights[i] ← WeightFunction(S).
end for
Choose a voxel from the neighbourhood according to its maximal weight and move the
spider to the voxel chosen.
```

4.1.1.2 Fix A Silk

Here, two other parameters are used, *reflevel* and *selectivity* computed from the histogram of the image. The first one corresponds to the graylevel of the region to detect, and the second one defines the tolerance to fix a drag-line with a voxel whose graylevel is not exactly reflevel. Probability to fix a drag-line with current voxel (here, the blue voxel in figure 4 follows a Gaussian distribution whose mean is *reflevel* and standard deviation is *selectivity*). Algorithm 3 consists on choosing a random number and evaluating it with a Gaussian function to add or not a drag-line. Its complexity is constant.

Algorithm 3 Fix A Silk

Require: S: Spider, Voxels: Matrix of voxels $\in \mathbb{N}^3$.

```
p ← random(0, 1).
if p < Gauss(level(position(S))) then
    Add a drag-line between postion(S) and lastfixed(S).
end if
```

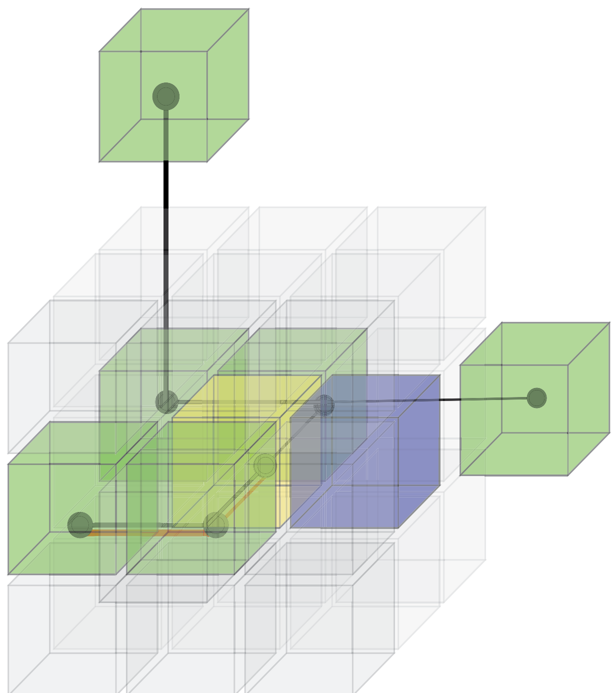


Fig. 4. Decision to fix a dragline with current voxel.

4.1.1.3 Move backward

This behavioural item allows spiders to come back to the last silked voxel (here, the yellow voxel in figure 4). This action is fired depending on the probability value defined as *backprobability*.

The aim of this action is to detect connected region of voxels: spiders can not go far of voxels they have silked. As it is shown by (Bourjot et al., 2003), disable this item leads to create drag-lines between two unconnected groups of voxels. Algorithm 4 test the possibility to return to the last fixed drag-line if the condition is satisfied in ligne 2. Its complexity is also constant.

Algorithm 4 Move Backward

Require: S : Spider
1: p ← random(0, 1).
2: **if** p < backprobability(colony(S)) **then**
3: position(S) ← lastfixed(S).
4: **end if**

4.2 Simulation process

Simulation process is based on stigmergy: each spider lets informations on the environment, those are used by other agents or by itself in a next cycle. Image segmentation emerges from the global task achieved by all the spiders: after a certain number of system time steps, drag-lines are created. Degree of a voxel defines number of drag-lines coming in or out of this voxel. In our case where there is a multiple regions detection, degree is given according to a colony. Global degree is the sum of degree of each colony. Region detected by a colony is composed of voxels having higher degree for this colony.

4.3 Problems

The social spiders method raises two problems:

1. There is a few number of parameters to fix;
2. Computing the number of time steps required or defining a stop condition.

Fixing the parameters can be problematic if it is done empirically. Indeed, to compare the results of social spiders segmentation among several images, we must be sure that the computation of the parameters will be in an equivalent manner in all cases.
Similarly, it is important that the stop condition meets the same criteria between different acts of segmentation. Otherwise, the results could be evaluated: a number of time steps not big enough leads to a poor qualitative analysis, and a number of time steps too high could increases the simulation time of the method without improving the quality of the results.

4.4 Solutions

First of all, we define the set of parameters to be fixed and its usage in the program. Then, we propose some ideas to reduce their number. Finally, we propose solutions to the two problems evaluated above.

4.4.1 Auto-detection of parameters

The method of social spiders has several parameters that will define the regions to be detected. These parameters are presented in Table 1.

Parameter	Description
reflevel	Grayscale of the colony
selectivity	Acts on the probability to weave a dragline from the current position of the insect
attractself	Attracts toward the draglines of the same colony
attractother	Attracts toward the draglines of the other colony
backprobability	Probability to turn back
$w(v)$	Weight of a voxel with a direct neighbourhood
saturationvalue	Upper bound of the weight computed inside a voxel

Table 1. Colony parameters.

First, it must know the number of colonies (the number of type of regions) and each of the properties, in particularly the grayscale reference and the standard deviation parameter. We have seen that the computation of these empirical parameters could be an obstacle for comparing the results produced by the social spiders method: if the parameters are not computed in the same way in each case, the comparison became unreliable because they

depends on the computation method. We will see in this part a method for determining the simulation parameters. This method will be used thereafter in all segmentation processes.
Each colony has a grayscale (or intensity) of reference which will serve the spiders of the colony to determine if they should fix a silk. There is two possibilities:

1. Determining "manually" the intensities, which is unattractive except where the intensities are known by the user;
2. Using a method to determine automatically the intensities.

The automatic detection will allow us to obtain optimal parameters without the user knowledge about the image. Thereafter, we will appoint an intensity level of voxels of this intensity.
We will use the histogram of the image to determine the intensity of interest to be detected. Indeed, a region inside an image usually implies a peak more or less important in the histogram of the image, but one can note that this histogram is particularly sensitive to noise. Therefore, we will introduce a method to reduce the noise effects on the histogram by smoothing it and allowing to obtain a maximum number of representative of the number of Regions intensities of the image.

Histogram smoothing

Histogram smoothing will reduce the peaks caused by noise. The method proposed here is done by time step. At each time step, the degree of intensity n becomes the average degrees of intensity n - 1, n, n + 1. We propose a method that detect automatically the optimal settings for the segmentation of the image. The method determines the number of settlements and the parameters *reflevel*, *selectivity* and the number of spiders in each colony.
First, we will determine the maxima in the histogram as described above. Then, the histogram will be smoothed until fewer maxima are significant. Indeed, the maxima caused by noise are on the peaks of the histogram whose slopes are small and will be erased by smoothing them in few time steps. Finally, the maxima representing a region is located on a peak having an important slope which requires a large number of smoothing in order to be eliminated. However, the risk to clear up an interesting maxima is not zero.
In order no to fall into aberrations, we add the condition that the number of maxima, and therefore the number of regions to detect, must be fixed by the user which is often a well known parameter. For example, The number of regions to detect in the brain is 10.
We get a series of maxima $M_1, M_2, \ldots, M_k$. The number of maxima, k, determines the number of spiders colonies that will be used. Each parameter is the maximum *reflevel* of the colony. Then, we will partition the intensities in as many Regions as maxima detected. To do this, simply find the minimum level of intensity between two maxima. This yields a series of minima $m_1, m_2, \ldots, m_{k-1}$ so that $0 < M_1 < m_1 < \ldots < m_{k-1} < M_k <$ L where L is the maximum intensity. The k Regions intensities obtained are $[0; m_1],]m_1; m_2], \ldots,]m_{k-1}; L]$.
The variance of each class provides the parameter *selectivity* of each column. The number of spiders per colony is the same for all colonies.
We can now determine the main parameters of the social spiders method. It remains to determine the numbers of spiders per region and the attraction of the silks on spiders.

4.4.2 Stop condition

We have seen that the social spiders method has a life cycle which is repeated a number of times until the image is segmented. The number of time steps will influence two important points on the result of segmentation:

1. The quality of segmentation;
2. The execution time required to achieve this result.

At each time steps, the spiders will weave between the voxels that will be used to determine to which region belongs the voxel. If the number of time steps is not enough, the number of voxels that do not belong to any region will be important and the result will be of poor quality. On the contrary, if the number of time steps is too large, the spiders will only increase after a certain time the silks already existing without providing any new information. This last point will have as effect a longer execution time for an approximatively identical quality result.
As for the parameters, the computation of the number of time steps is an important point to obtain comparative results that are credible. Rather than fixing a number of time steps, it is possible to determine a stop condition to be verified before every time steps.

Definition Let β be the number of silks fixed between voxels whose degree is zero during a time steps.

The result of socials spiders segmentation depends on the silk which will be fixed between the voxels. During a time steps, when β tends towards zero, we can consider that the system stabilizes and the spiders only reinforce the existing silks.
It is possible to detect when β remains at zero. This moment determines to stop the simulation. We can improve this condition by adding two parameters:

- A threshold that determines when β may be considered invalid;
- The number of authorized β zero before stopping the simulation.

The threshold may be determined by the number of spiders. Indeed, at each time steps, each spider has the possibility to fix a silk. The number of silks fixed during a time steps is bounded by the number of spiders.

5. BrainWeb application

In this section, we present the model generated by Brainweb[6] and its various components. We introduce the validation steps and the comparison technique used for the evaluation. Finally, the results are exposed and discussed.

5.1 BrainWeb

Brainweb is a Simulated Brain Database which contains a set of realistic MRI data volumes produced by an MRI simulator. These data can be used by the neuroimaging community to evaluate the performance of various image analysis methods in a setting where the truth is known. BrainWeb offer two anatomical models: normal and multiple sclerosis. For both of these, full 3D data volumes have been simulated using three sequences (T1-, T2-, PD- weighted) and a variety of slice thicknesses, noise levels, and levels of intensity

[6] http://www.bic.mni.mcgill.ca/brainweb/.

non-uniformity. As we are interested by the noise levels, The noise added by BrainWeb has Rayleigh statistics[7] in the background and Rician statistics[8] in the signal regions. The "percent noise" number represents the percent ratio of the standard deviation of the white Gaussian noise versus the signal for a reference tissue.

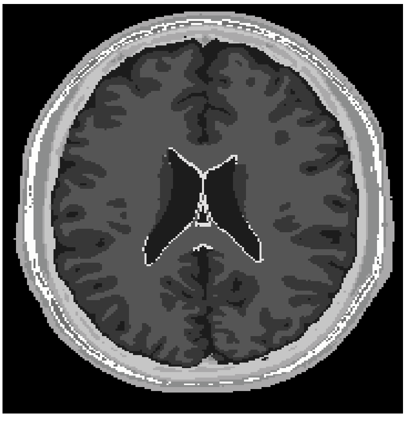


Fig. 5. Slice of the 3D model.

The parameters of the model to be used in this study are:

- Modality: T1
- Slice thickness: 1mm
- Noise: X%
- Intensity non-uniformity: 20%
- Dimensions: 181 x 217 x 181 voxels

Where X is the variability of the noise in the image. It will goes from 0% till 9%.
We can distinguish ten items in figure 5 are listed in Table 2.

5.2 Experimentation

In this part, we compare the social spiders method with other segmentation methods. These comparisons allow us to determine whether the social spiders method leads to a good segmentation compared to traditional segmentation methods:

- A classification method by thresholding: the Otsu method;
- A region-based method: the Region Growing method.

To compare these methods, we need to establish criteria to be used on all test images. We compare the results on several points:

1. The number of regions;
2. The pourcentage of voxels with correct labels before and after post-processing CL_b and CL_a between the truth model and the segmentation result.

[7] http://en.wikipedia.org/wiki/Rayleigh_distribution.
[8] http://en.wikipedia.org/wiki/Rice_distribution.

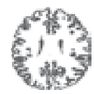
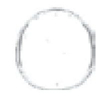
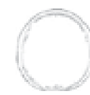
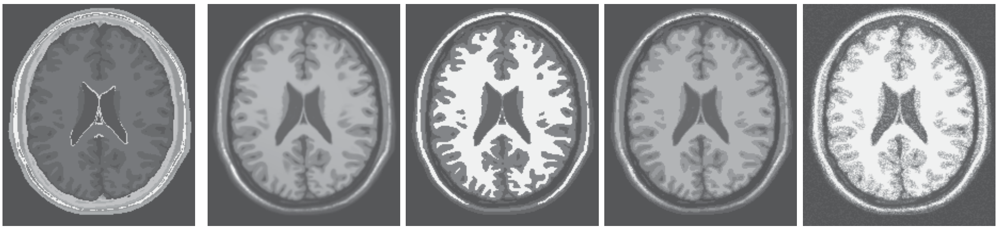
Element	Pourcentage	Form
background	42.2%	
csf	5.2%	
grey matter	12.7%	
white matter	9.5%	
fat	2.1%	
muscle skin	8.7%	
skin	10.2%	
skull	5.1%	
glial matter	0.1%	
connective	4.2%	

Table 2. BrainWeb Model: the various components.

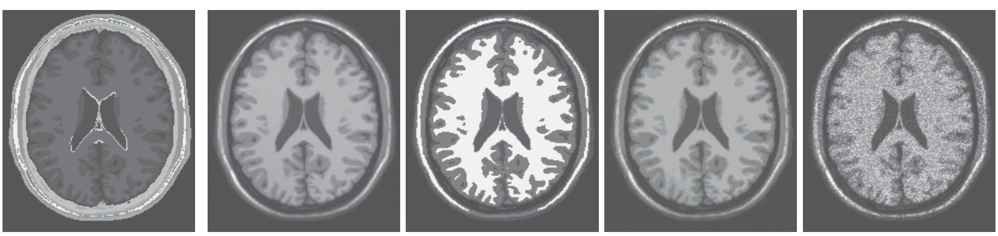
The number of regions allows us to determine whether the method considered detects a number of regions close to reality. It is possible that classical methods detect regions with insignificant size. That is why we add to the total number of regions, the number of regions having insignificant size. For the classical methods of segmentation, we consider a region as insignificant if its size is less than 10 voxels.
The computation of the number of regions is done on the segmentation method result on which a labelling is added to the connected components to consider the regions connected.
The execution time to be given comes from the simulation of the methods on a machine equipped with an Intel Quad Q9550 (4 cores having 2.83GHz) and 4GB of RAM. The operating system of this machine is a Linux kernel 2.6.21 x86_64. The brain images are composed of 10 regions which correspond to the structure of the brain. The size of the test volumes is 181x217x181 voxels. The slice presented in the screen capture here is the 94th slice of the 3D image. Furthermore, for the three methods compared here, the parameters which were empirically set are those which have produced the best results.
Figures 6, 7, 8, 9, 10 and 10 present the results of the different image segmentation techniques applied on a brain image with different levels. Their informations are explored in table 3.
As we can see from the results in table 3 that the Region Growing method (threshold = 406) has the lowest execution time, the lowest CL_a and the biggest number of regions. the Otsu method (thresholds = 406, 1175, and 1990) has produced a result similar to Region Growing with less number of regions. The social spiders method (time steps = 1000, backprobability = 0.1, $attract_{self}$ = 0.4, $attract_{other}$ = 0.6 and saturation = 50) has an important execution time with an additional region that corresponds to the extra voxels that were not detected by any spider. This region is not connected, the voxels that compose it are scattered throughout the image. It is therefore possible to perform a post-processing that would link these voxels to the colony that have a strong presence in their neighborhoods.



(a) (b) (c) (d) (e)

Fig. 6. 3D segmentation of Brain image without noise: a) Ground Truth Slice, b) Original Slice, c) Otsu thresholding, d) Region Growing, e) Social spiders.

Despite the fact that the difference between region-based segmentation methods is small, CL_a of the result of the spiders is the best. However, as the spiders methods is a stochastic method, we do not expect to get maximum CL_a. Let us test that this CL_a will remain stable when adding noise.
For that, we added noise to the original images. This noise added by brainweb has Rayleigh statistics[9] in the background and Rician statistics[10] in the signal regions. The "percent noise" number represents the percent ratio of the standard deviation of the white Gaussian noise versus the signal for a reference tissue. The results and the statistics of the different image segmentation techniques applied on the brain noisy image are presented below with different noise levels.

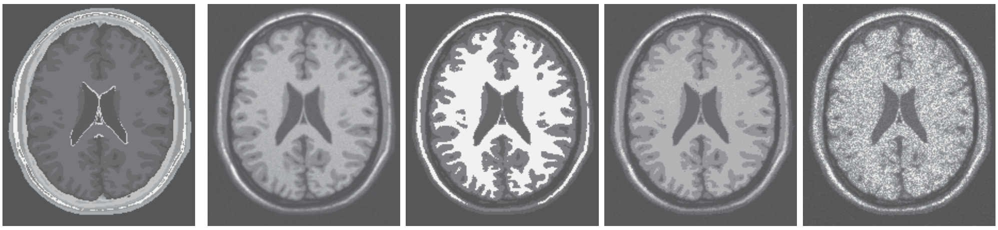


(a) (b) (c) (d) (e)

Fig. 7. 3D segmentation of Brain image with 1 % of noise: a) Ground Truth Slice, b) Original Slice, c) Otsu thresholding, d) Region Growing, e) Social spiders.

For 1% of noise, the Region Growing (threshold = 410) method has produced an oversegmentation with an important decrease of the CL_a and the best time to process. The Otsu (thresholds = 410, 1168 and 1966) method has lightly oversegmentated the image despite the fact that its CL_a has not decrease a lot when adding the 1% of noise. The CL_a of the social spiders method (time steps = 1000, backprobability = 0.1, $attract_{self}$ = 0.4, $attract_{other}$ = 0.6 and saturation = 50) has increased a little bit and remained stable as the best one in term of performance. This correlation is the result of a big computation time.
For 3% of noise, the Region Growing (threshold = 420) method continues to make an oversegmentation with an improvement of the CL_a and a stability of the time to process. This improvement is the result of a good post-processing. The Otsu (thresholds = 420, 1123 and

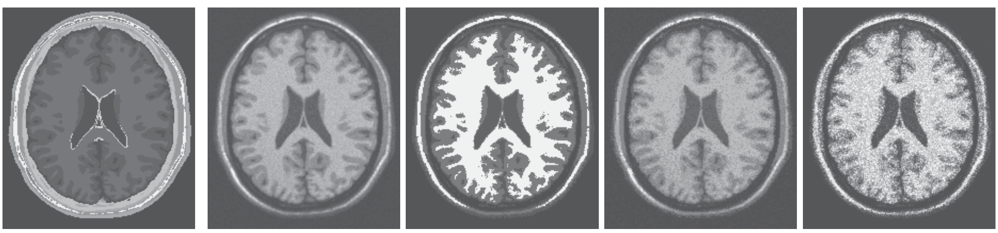
[9] http://en.wikipedia.org/wiki/Rayleigh_distribution.
[10] http://en.wikipedia.org/wiki/Rice_distribution.



(a) (b) (c) (d) (e)

Fig. 8. 3D segmentation of Brain image with 3 % of noise: a) Ground Truth Slice, b) Original Slice, c) Otsu thresholding, d) Region Growing, e) Social spiders.

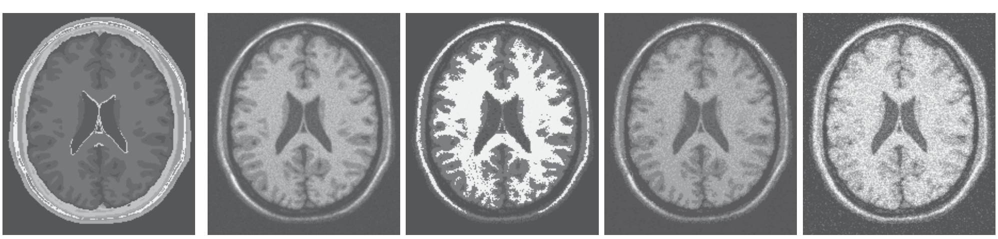
1889) method begin to be in competition with the social spiders method (time steps = 1000, backprobability = 0.1, $attract_{self}$ = 0.4, $attract_{other}$ = 0.6 and saturation = 50) by being very close in term of CL_a. The time to process of these two methods have made a small variation with the best time for the Otsu method.



(a) (b) (c) (d) (e)

Fig. 9. 3D segmentation of Brain image with 5 % of noise: a) Ground Truth Slice, b) Original Slice, c) Otsu thresholding, d) Region Growing, e) Social spiders.

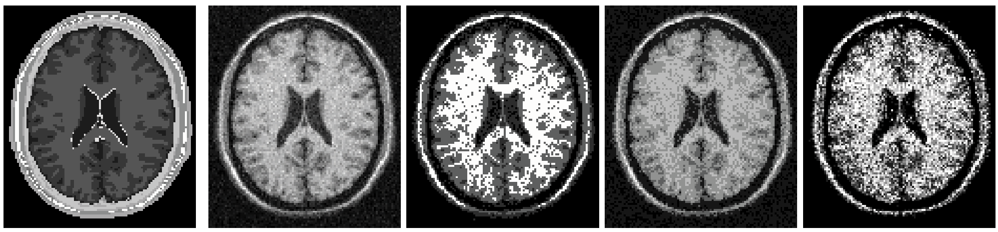
For 5% of noise, the Region Growing (threshold = 442) method has decreased the less between the three methods having always done an oversegmentation and remaining the worst. The Otsu (thresholds = 442, 1099 and 1823) method remains in challenge with the social spiders method (time steps = 1000, backprobability = 0.1, $attract_{self}$ = 0.4, $attract_{other}$ = 0.6 and saturation = 50) by continuing to be very close in term of CL_a.



(a) (b) (c) (d) (e)

Fig. 10. 3D segmentation of Brain image with 7 % of noise: a) Ground Truth Slice, b) Original Slice, c) Otsu thresholding, d) Region Growing, e) Social spiders.

For 7% of noise, the Region Growing (threshold = 500) has decreased the lot with the oversegmentation remained due to the complexity of the image and the additional noise effect. The Otsu (thresholds = 500, 1155 and 1880) took the lead from the social spiders method (time steps = 1000, backprobability = 0.1, $attract_{self}$ = 0.4, $attract_{other}$ = 0.6 and saturation = 50) with an effect on the post-processing which is done badly. This is mainly due to the non-selected voxels presented on the contour of the image.



(a) (b) (c) (d) (e)

Fig. 11. 3D segmentation of Brain image with 9 % of noise: a) Ground Truth Slice, b) Original Slice, c) Otsu thresholding, d) Region Growing, e) Social spiders.

% of noise		**Otsu**	**Region Growing**	**Social spiders**
0 %	**Regions**	917	70231	11
	Regions > 10 voxels	203	1006	11
	CL_b	67.91 %	68.02 %	70.51 %
	CL_a	67.90 %	67.70 %	69.13 %
1 %	**Regions**	1150	135093	11
	Region > 10 voxels	205	1609	11
	CL_b	67.90 %	59.90 %	71.23 %
	CL_a	67.89 %	60.03 %	69.51 %
3 %	**Regions**	2503	224653	11
	Region > 10 voxels	198	4890	11
	CL_b	67.17 %	57.63 %	67.96 %
	CL_a	67.16 %	64.51 %	67.62 %
5 %	**Regions**	9669	241482	11
	Regions > 10 voxels	209	3483	11
	CL_b	66.11 %	57.23 %	67.85 %
	CL_a	66.14 %	64.28 %	66.84 %
7 %	**Regions**	23138	249995	11
	Regions > 10voxels	317	4673	11
	CL_b	63.83 %	49.73 %	63.26 %
	CL_a	64.21 %	54.54 %	62.92 %
9 %	**Regions**	42392	186376	11
	Regions > 10voxels	330	2935	11
	CL_b	60.69 %	47.13 %	53.21 %
	CL_a	62.06 %	47.41%	55.02 %

Table 3. 3D results: Brain image with different levels of noise.

For 9% of noise, the Region Growing (threshold = 539) method continues to decrease the lot with also a decrease of the number of regions and the best time to process. The Otsu (thresholds = 539, 1172 and 1866) method has remained the best in term of CL_a while the social spiders method (time steps = 1000, backprobability = 0.1, $attract_{self}$ = 0.4, $attract_{other}$ = 0.6 and saturation = 50) decreases a lot due to the important number of non-silked voxels. Therefore, as shown in figure 12, it appears that the social spiders segmentation is robust to noise effect from 1% to 5% to and could be better for the rest. It also shown that social spiders method is better the region growing method (p-value = 0.08983) and have conducted similar results to that done by the Otsu method. This robustness has however led to an additional region in the image with a big influence on time to process.

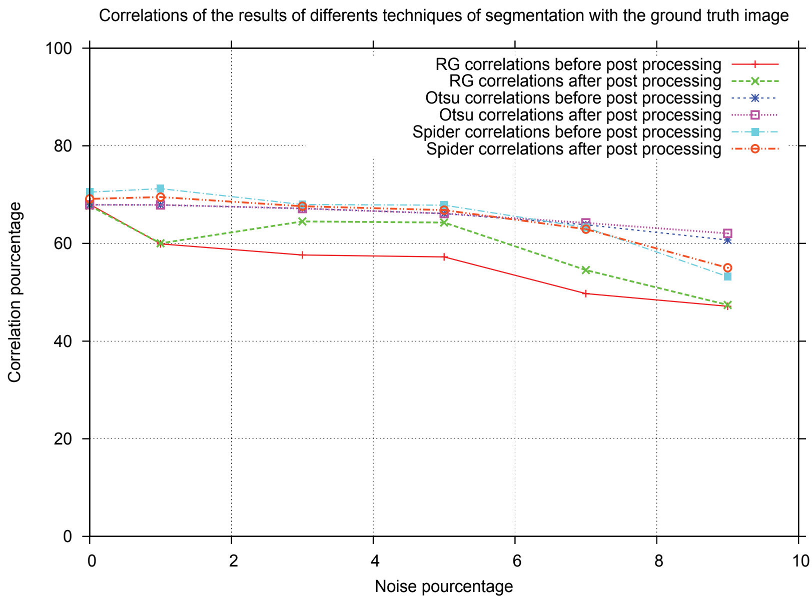


Fig. 12. Correlation of different segmentation techniques.

6. Conclusion

In this chapter, we have presented a new region-based method, the social spiders method. This method has been compared with the Region Growing Method and the the Otsu method. These comparisons focused on the nombre of voxels well labeled and the number of regions produced. The spiders method has produced a non neligeable time processing in the case of non-noisy image with a result better than the others. When noise is added, the processing time has increased a little bit but with a better result than the other region-based method and a similar result with the voxel-based method. Note that the results of social spiders method are influenced by the number of parameters to fix and particularly the repartition of the agents on the grid and the number of step to do. But automatic methods has been described to fix these parameters. Through these comparisons, we have put forward some drawbacks on the social spiders method. Particularly, we have seen that this method produced an additional region composed of the non selected voxels and that the execution time was particularly

long as discussed above. Some solutions have been considered to improve the method. The spiders seem sensitive to the topology of the image, so it is possible to guide the movement of spiders with a gradient or a laplacian. Indeed, these measures will provide informations on the possible presence of contours. It would be then possible to use other species like the social ants to guide the social spiders in order to cover all non-selected voxels.
However, the social spiders method is based on an architecture that is ideal to be parallel. This method is composed of a group of agents that can be spread over several processors. As GPU[11] allows us to share memory space betwenn processes. So, it is easy to use the social spiders program by putting the image to segment in the global memory of the GPU. First tests have shown that the processing time only depends from the number of steps fixed by the user. They also give back the simulation time to the same level of the other methods.

7. References

Ballet, P., Zemirline, A. & Marcé, L. (2004). The biodyn language and simulator. application to an immune response and e.coli and phage interaction, *JOURNAL of Biological Physics and Chemistry* 4: 93 – 101.

Bankman, I. (2000). *Handbook of Medical Imaging: Processing and Analysis*, Academic Press.

Bourjot, C., Chevrier, V. & Thomas, V. (2003). A new swarm mechanism based on social spiders colonies: from web weaving to region detection, *Web intelligence and Agent Systenms: An International JOURNAL - WIAS* 1(1): 47–64.

Camazine, S., Deneubourg, J.-L., Franks, N., Sneyd, J., Theraulaz, G. & Bonabeau, E. (2001). *Self-Organization in Biological Systems*, Princeton University Press, Princeton, NJ, USA.

Chen, T.-S., Liao, P.-S. & P.-C.Chung (2001). A fast algorithm for multilevel thresholding, *JOURNAL of Information Science and Engineering* 17: 713–727.

Chevrier, V. (2002). Contributions au domaine des systemes multi-agents, *Technical report*, Université Henry-Poincarré - Nancy 1.
URL: *http://www.loria.fr/ chevrier/Publi/HDR.zip*

G. Theraulaz, G. & Spitz, F. (eds) (1997). *Auto-organisation et comportements collectifs : la modélisation des sociétés d'insectes*, Editions Hermes, pp. 91 – 140.

Gleizes, M.-P. & Marcenac, P. (eds) (1999). *Coordination par le biais de l'environnement : une approche biologique*, Hermes, pp. 237–250.
URL: *http://hal.inria.fr/inria-00107820/en/*

Hemernehl, G., McInerney, T. & Terzopoulos, D. (2001). Deformable organisms for automatic medical image analysis, *Medical Image Analysis* pp. 66–76.

Hendee, W. & Ritenour, E. (2002). *Medical Imaging Physics*, Wiley and Sons.

Jackson, E. (2007). Social spiders, *Current Biology* 17(16): R650 – R652.

Liu, J. & Tang, Y. (1999). Adaptive image segmentation with distributed behavior-based agents, *IEEE Transactions Pattern Analysis and Machine Intelligence* 21(6): 544–551.

Otsu, N. (1979). A threshold selection method from gray level histograms, *IEEE Trans. Systems, Man and Cybernetics* 9: 62 – 66.

Pham, D., Xu, C. & Prince, J. (2000). A survey of current methods in medical image segmentation, *Annual Review of Biomedical Engineering*, Vol. 2, Annual Reviews, pp. 315 – 338.

[11] Graphical Processing Unit.

Ramos, V. & Almeida, F. (2000). Artificial ant colonies in digital image habitats - a mass behaviour effect study on pattern recognition, *Proceedings of ANTS'2000 - 2nd International Workshop on Ant Algorithms (From Ant Colonies to Artificial Ants), in Marco Dorigo, Martin Middendorf and Thomas Stuzle (Eds.)* pp. 113–116.

Ramos, V., Muge, F. & Pina, P. (2002). Self-organized data and image retrieval as a consequence of inter-dynamic synergistic relationships in artificial ant colonies, *Frontiers in Artificial Intelligence and Applications, Soft Computing Systems - Design, Management and Applications, 2nd International Conference on Hybrid Intelligent Systems, IOS Press, in Javier Ruiz-del-Solar, Ajith Abraham and Mario Köppen (Eds.)* 87: 500–509.

Shapiro, L. & Stockman, G. (2001). *Computer Vision*, Prentice-Hall.

Theraulaz, G. & Spitz, F. (1997a). *Auto-organisation et comportement*, Hermes, Paris.

Theraulaz, G. & Spitz, F. (eds) (1997b). *What are the mechanisms involved in the emergence of cooperativity ? The spider model*, Hermes, Paris, pp. 85–90.

Vasilescu, G. (2005). *Physical Noise Sources*, Springer Verlag, Heidelberg, pp. 45–68.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*, Wiley and Sons.

19

Towards Implementing an Intelligent System for Securing and Monitoring using Agents

Faisal Alkhateeb[1], Zain Al-Abdeen Al-Fakhry[2], Eslam Al Maghayreh[3], Ahmad T. Al-Taani[4] and Shadi Aljawarneh[5]
[1,2,3,4]*Yarmouk University, Irbid*
[5]*Al-Isra University, Amman*
Jordan

1. Introduction

Most of organizations such as universities have critical positions and rooms and therefore require to be secured. We mean by "securing" here is detecting and isolation of any problem: temperature, motion, stole attempts, etc. Detection is the phase of announcing a problem while isolation is the phase of determining the type and the location of the problem. Securing a university campus could be achieved based upon a number of security guards making security checking rounds (on foot or by vehicles) inside the university campus and in particular around the critical rooms. Other way is to use security camera-based systems. Nonetheless, such systems need to be monitored on the fly by a person. Additionally, they are inapplicable in some cases such as detecting unseen phenomena (e.g. temperature problems). Finally, cameras should be spread out over critical positions in order to monitor every corner. In this chapter, we present an effective and low security system that operate over a wireless network and based on multi-agents to secure the buildings of a given university.

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations, and are now used in many industrial and civilian application areas, including industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, and traffic control.

The remainder of this chapter is organized as follows: Section 2 provides an overview of some related systems. We introduce in Section 3 the ACCESS architecture and location based services and their applications. Then we introduce our system in Section 4. We present in Section 5 a simulator as well as an implementation of the system.

2. Related systems

In this section, we divided into two main categories: Multi-agent/sensor-based systems and Vision-based systems.

2.1 Multi-agent/sensor-based systems

A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents. Multi-agent systems can be used to solve problems which are difficult or impossible for an individual agent or monolithic system to solve. Multi-agent are open and extensible systems that allow for the deployment of autonomous and proactive software components. For instance, they have been applied to several application areas such as context-aware and infomobility services. The term context refers to any information that can be used to characterize the situation of an entity, where an entity is a person, place, route, train, or any relevant object. So, context-aware services could be defined as those services that deliver "up to the minute" information about a given entity. Such services include location-based services, travel assistance and planning, impaired people mobility assistance. Agent-based approaches are considered as an appropriate solution to address the issue of the overwhelming data traffic in Wireless Sensor Networks (WSNs).

Bus catcher 2 is an agent-based architecture for wireless bus travels assistants (O'Hare et al., 2000; Strahan et al., 2003). The system delivers up to the minute bus network information that is relevant to the transportation and movement of the user.

Another example is EasiShop (Keegan & O'Hare, 2002), which is an agent-based, location-aware, automated ubiquitous commerce (uCommerce) system using mobile agent interaction facilitated by the Bluetooth wireless radio transmission medium.

Gulliver's Genie (O'Grady, n.d.) is an archetypical ubiquitous computing application. Gulliver's Genie uses three basic concepts in its construction, these are: agency, mobility and adaptively. The architecture adopts a Multi-Agent Systems (MAS), where agents manage and maintain a context within which mobile users exist and based upon this context seeks to adapt and personalize content based upon perceived individual user needs.

Another usage of sensors networks and environmental monitoring (Mainwaring et al., 2002). Deploying numerous networked micro-sensors in natural spaces can enable long-term data collection at scales and resolutions that are difficult, if not impossible, to obtain otherwise. Wireless sensor networks have been also used for commercial lighting control (Sandhu et al., 2004), which provides a practical application that can benefit directly from artificial intelligence techniques. This application requires decision making in the face of uncertainty, with needs for system self-configuration and learning. Such a system is particularly well-suited to the evaluation of multi-agent techniques involving distributed learning.

Senor-based systems could be also used to automatically notify emergency centers of a given problem. They are commonly equipped with distributed sensors and used to collect incident severity information. The system then communicates with an emergency dispatcher to assist in determining the appropriate emergency personnel. As an example of such systems is OnStar (Carrigan et al., 2005). It has been designed in 1996 to deliver safety, security and information services using wireless technology and the Global Positioning System (GPS) satellite network. OnStar services include but are not limited to automatic notification of air bag deployment, stolen vehicle location assistance, emergency services, roadside assistance with location, and remote door unlock. It also allows drivers to make and receive voice-activated wireless calls and access a wide range of other information services through a nationwide cellular network. However, such systems have limited usage in the case of University campus since problems that can be occurred in the campus should be treated locally by a particular person. For instance, a temperature problem occurred in a servers' room should be treated by the person from the University responsible for solving that problem. Moreover, such systems cannot be directly applied everywhere because it is required to identify each country' section in order to use the Global Positioning System (GPS) satellite

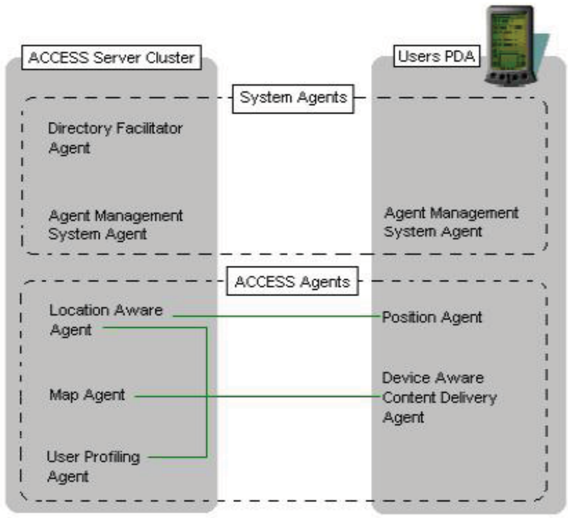


Fig. 1. ACCESS architecture.

network, where a section can be a street, subway, university campus, etc. Additionally, the installation and the maintenance of such systems are too expensive.

2.2 Vision-based systems

The second category of the systems that are closely related to the system presented in this chapter is the vision-based systems. An example of such systems is the vision-based traffic are vision-based traffic accident detection systems (such as traffic Accident Recording and Reporting Systems (ARRS) (Ki, 2007; Ki et al., 2006; Shehata et al., 2008; Wells & Toffin, 2005)). Such systems are either image-based or video-based and used for detecting, recording, and reporting an incident. They consist mainly of a charge coupled device (CCD) camera (located at several places) for monitoring purposes and/or a digital video recorder (DVR) that has recorded all the situations at a given place, and finally an image processing unit that detects images which could be related to an incident. Nonetheless, the usages of such systems are limited since cameras (CCD) and digital video recorders (DVD) cannot be distributed everywhere. Additionally, it is required to have at least one person to monitor the system and alert the security guards in case of a security attack.

3. Research background

In this section, we present the ACCESS architecture since our system is built on top of it as well as Location-Based Services (LBS) that could be used in our system to locate security guards.

3.1 ACCESS architecture

The Agents Channeling (or Conveying) ContExt Sensitive Services (ACCESS) architecture (*cf.* (Strahan et al., 2003)), as shown in Figure 1, consists of two agent categories: system agents and ACCESS agents.

The System Agents manage the platform. The Agent Management System manages the creation and deletion of agents and provides a white pages directory service for agents that reside on that platform. The Directory Facilitator Agent provides a yellow pages service for agents, it has information about agents and the services they provide.

The ACCESS Agents consists of generic agents responsible for channeling context sensitive services. The Device Aware Content Delivery Agent manages the local cache and the user interface to ensure real time content delivery to the user. The Position Agent determines the users physical position and informing specified agents of the user's location and direction. The Location Aware Agent receives location information from the position agent. It also acts as a hotspot manager, where a hotspot is an area of physical space limited by specific boundaries. It can inform specified agents of when the user enters or leaves a hotspot and provides functionality to allow agents register and delete hotspots. The User Profiling Agent tries to determine what type of user it believes is using the system. It monitors the user's location preferences, available hardware and services used. The Map Agent is responsible for dynamically generating maps merged with service specific overlays.

3.2 Overview of location based services

As a result of the huge numbers of mobile phone users, it becomes possible to use the mobile in specifying the location of the mobile users. One method that can be used to accomplish this task is the use of location based services (LBSs). A location-based service (LBS) is an information service that can be accessed using mobile devices through the mobile network and utilizes the ability to make use of the geographical position of the mobile device (Adams et al., 2003; Steiniger et al., 2006; Vijayalakshmi & Kannan, 2009).

There are a number of services included in LBSs such as specifing other people positions, other resources, and the position of the user itself, etc. The primary service is obtaining the position of the user itself in order to use a given service such as finding the nearest restaurant.

There are different types of location based services (Bellavista et al., 2008; Chen et al., 2002; Rao & Minakakis, 2003; Schiller & Voisard, 2004):

Pull services: this type of service is initiated by the client himself by requesting the service from LBS and giving the permission to the LBS to know his/her location. One example of a Pull service is traffic information requested by sending an SMS to a given number such as 1000 that is specialized in this service. The service provider needs to know the location of the client to provide him with information that fulfills the request of the client.

Push services: this type of service is initiated by the service provider as a result of previously getting the authority from the client to receive the requested information from the service provider. For example, a client can register in a traffic service. Every morning when he/she is going to work at 7:30AM, the service provider supplies her/him with information regarding traffic movement at that time depending at her/his current location. Consequently, if there are a huge number of vehicles at a given street it will provide her/him with possible alternatives that she/he can go on so that she/he can arrive at her/his work on time.

Tracking services: this type of services allows someone to request the location of another one. In such type of services, the person whose location is required has to permit the first one to follow him. A client can press a number and send an SMS message to specify a given person location, if the person replay, the client will receive his location via SMS.

Emergency Services: such kind of services provide an automatic or manual call to civil defense in case of an accident or risk. This service should be provided to all mobile clients registered in this mobile company. In USA when you dial 911 from your mobile this service will allow the emergency personnel to specify the location of the caller directly.

4. Campus-agents security system

In this section, we present the system requirements, its architecture and a typical scenario of the Campus-agents security system.

4.1 System requirements

The main components of the system are:

- Wireless sensors and receivers: the wireless sensors will be distributed in critical rooms to detect and to alert the system in case of a security attack occurred.
- Database: the database contains information about sensor flags and locations, security guards, critical campus rooms in which there are sensors.
- PDAs and/or mobiles: security guards will be provided with PDAs or mobiles[1].

4.2 Campus-agents service architecture

Due to the fact that the ACCESS architecture is generic, application specific Service Agents, which use the ACCESS architecture could be added to the system. Therefore to fit the requirements of our system, we extend the architecture as shown Figure 2, which we have proposed in (Alkhateeb et al., 2010). The Service Agents for this application are:

- Alarm Management Agent: its function is to receive the signal from the wireless sensor and determine the code of that sensor and get the building code from the sensor table and send it to the Building Position Agent, it will also get the floor number, the room number and security degree and send them to the Security Guard Call Agent.
- Building Position Agent: its function is to receive the building code from the Alarm Management Agent and will get the building name and position from the building table and send them to the Security Guard Call Agent.
- Security Guard Call Agent: it receives the information from the Alarm Management Agent and the Building Position Agent and will find the shortest distance between the building in which the alarm activated and the nearest free guard (by checking the security guard position table), then it will send the building position to the Map Agent to highlight the building on the map displayed in that guard's PDA and send a text message to the same PDA telling the guard to go to that building and to the room where the sensor is located.

4.3 Typical scenario of the system

The principal use of the Campus-agents will be detecting security attacks. Basically, the system will work as follows:

If one of the wireless sensors detects an attack occurring at some room in one of the university buildings, then it sends a signal to the system. The Alarm Management Agent receives the signal and determines the room code by searching the sensor table of the database based on the identifier of the sensor sending the signal.

The Building Position Agent receives the code from the Alarm management Agent and will get the building name and position from the building table and send them to the Security Guard Call agent.

Then the Security Guard Call agent receives the information from the Alarm Management Agent and the Building Position Agent and will find the shortest distance between the building in which the alarm activated and the location of security guards. Also it identifies the nearest free guard by checking the security guard position table, pass the building position to the Map Agent to highlight the building on the map displayed in that guard's PDA, and send a text message to the same PDA telling the guard to go to that building and to the room where the sensor is located. One usage of the Position agents and Location agents is to check whether a given security guard is out the zone of the University campus or not. This could

[1] Note that it is assumed that everyone posses a mobile and hence this requirement could be ignored.

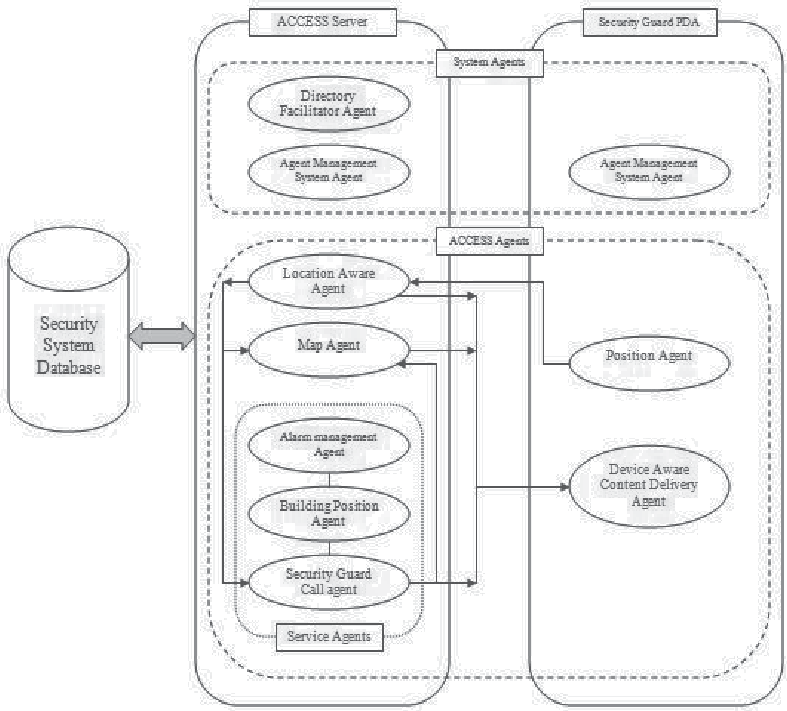


Fig. 2. Campus Service Architecture.

be achieved by using location-based services discussed earlier. Based on that information, the system will allocate other security guards to fill his position.

5. System simulation and implementation

This section introduces the components that we have used in our implementation. For more details, the reader could refer to [16].

5.1 Agent factory agent programming language 2 (AFAPL2)

ACCESS was originally built using Agent Factory Micro Edition (AFME) (C. Muldoon, 2006), which is a minimized footprint agent platform developed to enable the creation of intentional agents for mobile devices. Due to the fact that the NetBeans IDE does not compile AFME, we used the AFAPL2, which is an extended version of the original Agent Factory Agent Programming Language (AFAPL) (cf. (Collier, 2007)).

5.2 Databases

The database we constructed contains the following tables:

- Attack table: this table is used to make reports on the attacks detected by the sensors and dealt with by the guards during a specified time period.
- Building table: this table is used to store the buildings' name and co-ordinations.
- Securityguard table: this table is used to store the security guards information.
- Sensors table: it is used to store the sensors information.

Fig. 3. System simulation using laptops.

- Sgposition table: this table is used to store the security guards positions.

5.3 System simulation

Without loss of generality, we have used laptops to simulate the system. This does not provide any restriction to the usage of our system. That is, the wireless sensors could be integrated to the system by first adding their codes and/or their IP addresses to the sensor table mentioned above. Then, they will be connected to the system using a number of access points deployed in the system. Types of sensors include but not limited to glass break detectors, door and window open detectors, motion detectors.

In the simulation, we used two laptops: one to represent the wireless sensor and the other is the server both equipped with built-in wireless network card (WiFi). We used a Wireless router as an access point. We gave them fixed IP addresses.

The classes "java.net.Socket" and "java.net.ServerSocket" are used to implement sensor communications which use the TCP protocol (see Figure 3).

5.4 System work flow

At the beginning of the system execution the start window will be displayed and the user has to click the start button to display the rest of the windows.

From the sensors window the user will select a sensor (which is a simulation of the sensors control panel) and click it to set the alarm of that sensor on.

The alarm management agent, which its task to monitor the wireless sensors signals, will detect the new alarm signal and then it will signal the building position agent to act. After receiving the signal from the Alarm Management Agent, the building position agent will call some classes to link to the sensors table to get the building name, floor number and room number where the sensor is deployed. These will also create the calling message that will be displayed on the security guard PDA. The building position agent will also call another class to inquire the building table to get the co-ordinations of the building, and then it will send them to the security guard call agent.

The security guard call agent contains several classes that are used to inquiry the Sgposition table and calculate the distance between each free security guard and the building to select

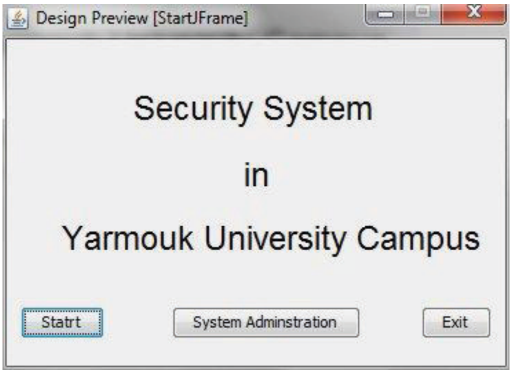


Fig. 4. System start menu.

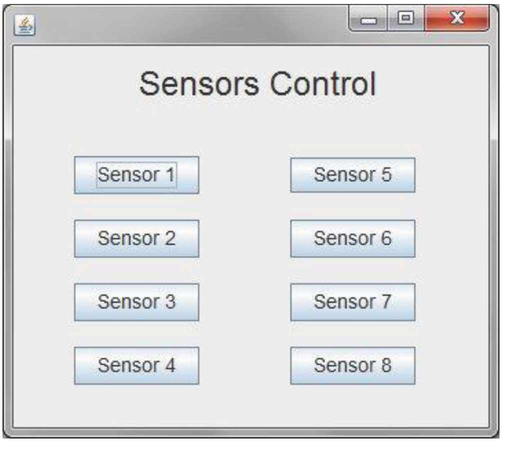


Fig. 5. Sensors control menu.

the nearest available security guard to the building. Then it will send two signals, one to the map agent and the other one to the Device aware content delivery agent.
The map agent in tern will select the map that will be displayed in the security guard PDA, and then it will send a signal to the Device aware content delivery agent.
The Device aware content delivery agent will receive the map address from the map agent and display the map and a message on the security guard PDA as shown below.

6. Conclusion and future work

We have proposed in this chapter a novel system that can be employed to build an effective and low cost security system made by wireless sensor network and multi agents. The intelligent agents (with the aid of sensors) are responsible for detecting security attacks and alerting security guards. Consequently, there is no need for a human to continuously monitor the rooms and the buildings of the University campus. Additionally, the system could be used to secure any organization containing critical locations.
Although we have used laptops in the implementation to simulate the functions sensors, real sensors could be integrated seamlessly to the system.
As a future work, we plan to investigate the usage of cameras in conjunction of this system and therefore use the image processing techniques to extract security attacks features. In

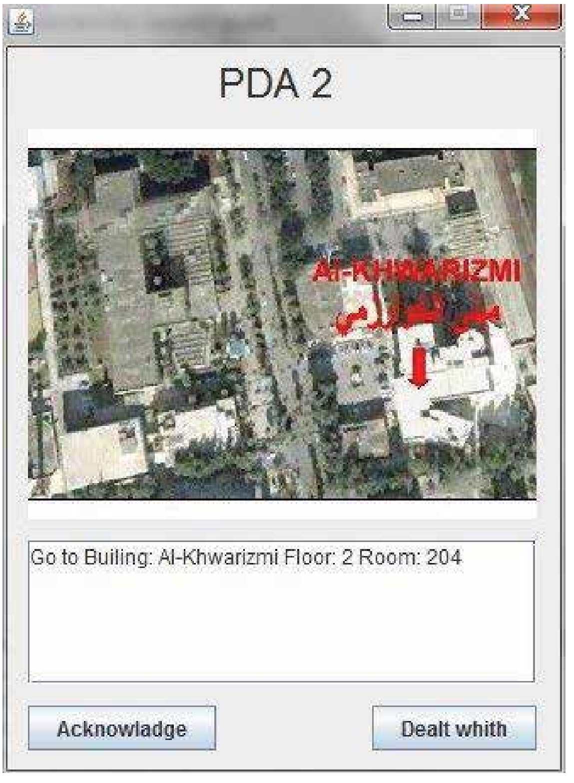


Fig. 6. A message and a map displayed in the PDA.

this direction, in case of personal attacks, location-based services could be used to track the attacker as discussed in (Al-Fakhry, 2010).

7. References

Adams, P. M., Ashwell, G. W. B. & Baxter, R. (2003). Location-based services — an overview of the standards, *BT Technology Journal* 21(1): 34–43.

Al-Fakhry, Z. (2010). Towards implementing intelligent security system.

Alkhateeb, F., Al Maghayreh, E. & Aljawarneh, S. (2010). A multi agent-based system for securing university campus: Design and architecture, *Proceedings of the International Conference on Intelligent Systems, Modelling and Simulation (ISMS'10)*, IEEE, Liverpool, England, pp. 75–79. URL: *http://doi.ieeecomputersociety.org/10.1109/ISMS.2010.70*.

Bellavista, P., Küpper, A. & Helal, S. (2008). Location-based services: Back to the future, *IEEE Pervasive Computing* 7(2): 85–89.

C. Muldoon, G. M. P. O'hare, R. C. M. J. O. (2006). Agent factory micro edition: A framework for ambient applications, *Proceedings of Intelligent Agents in Computing Systems Workshop (held in Conjunction with International Conference on Computational Science (ICCS)) Reading, UK. Lecture Notes in Computer Science (LNCS)*, Springer-Verlag, pp. 727–734.

Carrigan, R., Milton, R. & Morrow, D. (2005). ADVANCED AUTOMATIC CRASH NOTIFICATION (AACN), *Computerworld honors case study*. URL: *www.onstar.com/* URL: *www.cwhonors.org/archives/case_studies/onstar.pdf*.

Chen, Y., Chen, X., Ding, X., Rao, F. & Liu, D. (2002). Bluelocator: Enabling enterprise location-based services, *Mobile Data Management, IEEE International Conference on* 0: 167.

Collier, R. (2007). Debugging agents in agent factory. programming multi-agent systems, *Lecture Notes in Computer Science (LNCS)*, Springer- Berlin/Heidelberg, Springer Berlin/Heidelberg, pp. 229–248.

Keegan, S. & O'Hare, G. (2002). Easishop: Context sensitive shopping for the mobile user through mobile agent technology, *Proceedings of PIMRC 2002 13th IEEE International Symposium on Personal Indoor and Mobile Radio Communications, IEEE Press*.

Ki, Y.-K. (2007). Accident Detection System using Image Processing and MDR, *International Journal of Computer Science and Network Security IJCSNS* 7(3): 35–39.
URL: *http://paper.ijcsns.org/07_book/200703/20070306.pdf*.

Ki, Y.-K., Kim, J.-W. & Baik, D.-K. (2006). A Traffic Accident Detection Model using Metadata Registry, *Software Engineering Research, Management and Applications, ACIS International Conference on Software Engineering Research, Management and Applications* pp. 255–259.

Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D. & Anderson, J. (2002). Wireless sensor networks for habitat monitoring, *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, SESSION: Applications and OS*, Atlanta, Georgia, USA, pp. 88–97.

O'Grady, M.J., O. G. (n.d.). Gulliver's genie: Agency, mobility & adaptivity, *Computers & Graphics, Special Issue on Pervasive Computing and Ambient Intelligence - Mobility, Ubiquity and Wearables Get Together* 28(4, year = 2004,

O'Hare, G., O'Hare, P. & Lowen, T. (2000). Far and a way: Context sensitive service delivery through mobile lightweight PDA hosted agents, *Proceedings of 15th International Florida Artificial Intelligence (FLAIRS) Conference*, AAAI Press, Pensacola, Florida.

Rao, B. & Minakakis, L. (2003). Evolution of mobile location-based services, *Commun. ACM* 46(12): 61–65.

Sandhu, J. S., Agogino, A. M. & Agogino, A. K. (2004). Wireless sensor networks for commercial lighting control: Decision making with multi-agent systems, *Berkeley, ca 94720, moffett field, ca 94035*.
URL: *best.berkeley.edu/ aagogino/papers/AAAI_04_SN_J_Sandhu_ final.pdf*.

Schiller, J. H. & Voisard, A. (eds) (2004). *Location-Based Services*, Morgan Kaufmann.

Shehata, M. S., Cai, J., Badawy, W. M., Burr, T. W., Pervez, M. S., Johannesson, R. & Radmanesh, A. (2008). Video-based automatic incident detection for smart roads: The outdoor environmental challenges regarding false alarms, *IEEE Transactions on Intelligent Transportation Systems* 9(2): 349–360.

Steiniger, S., Neun, M. & Edwardes, A. (2006). Foundations of location based services, *Cartouche - lecture notes on lbs, v. 1.0*. URL: *http://www.geo.unizh.ch/publications/cartouche/ lbs_lecturenotes_steinigeretal2006.pdf*.

Strahan, R., Muldoon, C., O'Hare, G., O'hare, G. M. P., Bertolotto, M. & Collier, R. W. (2003). An agent-based architecture for wireless bus travel assistants, *Proceedings The Second International Workshop on Wireless Information Systems (WIS 2003)*.

Vijayalakshmi, M. & Kannan, A. (2009). Proactive location-based context aware services using agents, *International Journal of Mobile Communications* 7(2): 232–252.

Wells, T. & Toffin, E. (2005). Video-based automatic incident detection on san-mateo bridge in the san francisco bay area, *Proceedings of 12th World Congress on ITS*, San Francisco.

20

Pro-Active Multi-Agent System in Virtual Education

Victoriya Repka, Vyacheslav Grebenyuk and Katheryna Kliushnyk
Peoples-Uni,
Kharkiv National University of Radioelectronics
Australia,
Ukraine

1. Introduction

As virtual education becomes more and more widespread, its' application provides a unique opportunity for us to develop new applications in adaptive or intelligent agent technology. Adaptive or intelligent agent technologies allow education methods to be identified on a case-by-case basis, and undertaken regardless of location, time, age and life lifelong. There are many different distant education (DE) models developed to take into account modern tendencies of distributed system ideas and intelligent agent technology. The last strengthens the ontological features of DE system and move up the users from passive knowledge recipient role into active actors in the educational process. Therefore the main priority for the development of a modern virtual education system is to provide each student with an individual program, and to allow them to choose courses to fit their level of knowledge as well as make information searchable in accordance to the query criteria and within existing user skill sets. New methods and methodologies are being developed to solve this problem, as well as many others associated with virtual education, distant education, and e-learning.

An adaptation to a learner's personal interests, characteristics and goals is a key challenge in e-learning. In this chapter we discuss the architecture of web-based learning systems that addresses the learners' need for activities and content based on their preferences and equally considers the designer's and tutor's needs for the efficiency. The system aims to develop new methods and services for pro-active and adaptive e-learning. Proactive means the system involves acting in advance of a future situation, rather than just reacting. It means taking control and making things happen rather than just adjusting to a situation or waiting for something to happen. Adaptive means the learners are provided with a learning design that is adapted to their personal characteristics, interests and goals as well as the current context.

Currently, two approaches to adaptation are common within e-learning. In the first, dominated by a strong tradition in instructional design, a team produces a detailed design of content, interaction and presentation. Within the design different options may be worked out for different learners based on user data, e.g. level, interest or learning style. The options for adaptation are prepared at design time and require limited, if any, interaction of tutors at runtime. The second approach is based on the assumption that author and tutor is the same

person. The author designs the material. Next at runtime, the author, now as a tutor, adapts the course based on a direct interpretation of usage data, i.e. how well the learners succeed and what questions arise. However, both approaches tend to be (too) expensive because of high development costs or high delivery costs through extensive support.
Using of agent technologies in virtual education became very popular because of their properties like: reactivity – agents respond on environmental changes in the real time based on the rules of "WHEN event IF condition THEN action"; goal driven – agents have an ability to solve tasks and attain objectives, an ability to exist in a constantly active state having their own control flow; flexibility – agents' actions are not mounted rigidly; intellectuality – agents have an ability to learn, to find new solution, to change their behaviour with their own experience and experience of other agents as well. Due to these properties we can say the flexibility of action for such system becomes sufficiently notable and more over it has even some sort of "consciousness". We can individualize the virtual education process and build up individual educational trajectories for a student based on those agents' properties.

2. Related publications

Nowadays the agent technology has been applied in a various types of applications for education: to the design of peer-help environments, for information retrieval, for student information processing, distribution and feedback collection, pedagogical agents, teaching agents, tutoring agents, agents for assignment checking and agents for student group online support. Recently, agent e-learning systems are also used to perform mechanisms of interaction and collaboration in a network community environment.
In (Fuhua, 2005) it was suggested to call the adaptive learning environment for distributed learning as "a distributed adaptive learning environment." DALE (Distributed Adaptive Learning Environment) integrates agents and learning objects in the context of a distributed system. The main features of such environments are adaptability and distribution. They use intelligent software agents to provide the key intelligence to dynamically adapt the contents to the needs of particular learners. An empirical study aimed at the evaluation of the intelligent agents effectiveness in online instruction shown that agents can improve completion rates, learner satisfaction and motivation.
Samuel Pierre in (Samuel, 2007) operates an e-learning networked environment. This virtual environment can be defined as a software system within which multiple users, possibly located worldwide, interact with one another in real time (Singhal and Zyda, 1999).
Pashkin in (Burkhard, 2007) proposed agent-based architecture of intelligent distance learning system. This system contains agents divided into three groups: decision making agents that analyze data and require performance of actions from DLS side, Pedagogical Strategy Agent – interprets raw data received from Collector Agent and makes decision which rule should be executed, and Data Mining Agent -analyses of data to define association rules between student's behaviour and cluster students; interface functions agent – responsible for learning content presentation and human agents that model and represent of the human and perform tasks on behalf of them. The main goal is to model active interface for presentation of learning content depending on student's ability.
Interface or communication agents are available almost in all Multi-Agent Systems (MAS). For example in (Fuhua, 2005) they provide user support by interacting with the user. They help the user in completion of some task in an application. Interface agents carry out the user's instructions. In situations with no exact instructions to follow, agents act according to

the knowledge established through feedback from the user. Agents can also request help from other agents. Interface agents provide useful assistance to human beings. For example, e-mail filters learn to pick out junk e-mails by hints from the user and experience. According to (Fuhua, 2005) the MAS-based distributed-learning environment (DLEs), provide intelligent decision-making support as well as multitude of learning objects for students to choose and experience under various navigation environments. This environment infrastructure consists of the following agents: Information management agent; Intelligent decision-making support agent; MGIS support agent; Collaboration agent; and Interface agent.

The learning environments can support the interactions between agents that might be geographically dispersed on the Internet. DLEs provide an online training environment that enables students to acquire knowledge and improve safe navigation skills at any time and from anywhere, simply by using a Java-enabled browser.

Another similar environment, Intelligent Virtual Environments for Training (IVET) was proposed by Angelica de Antonio, Jaime Ramirez et al. in (Pechoucek et al., 2005). It is based on a collection of cooperative software agents. Among other things it includes agents able to simulate the behaviour of students and tutors, as well as agents able to plan the procedures to be taught prior to the tutoring process. IVET allows students to navigate through and interact with a virtual representation of a real environment in which they have to learn to carry out a certain task.

Adrianna Kozierkiewicz-Hetmanska established in (Nguyen et al., 2009) the concept for modification of learning scenarios in an Intelligent e-learning system. It is based on the ontology of knowledge structure of lessons. A set of linear relations between lessons takes into account. Such relation defines the order in which the lessons should be presented to a student, because some lessons should be learned before others. The system stores all student's preferences and changes in the user profile. Modification of a learning scenario is conducted in three steps. If student has a problem with passing a test for the first time he is offered repetition of the same lesson but in a different version. If the student fails again the system changes the lessons order, based on data of students who belongs to the same class. In another scenario if a student provides false information about himself then he may be classified incorrectly. If this possibility is taken into account, in the last instance the student is offered a modified lesson order based on all collected data.

All above works and also (Woda and Michalec, 2005), (Lin, 2007), (Shih and Hung, 2007), (Uhrmacher and Weyns, 2009), (Bellifemine et al., 2007) prompted our research.

Overall the intelligent learning systems are able to: conduct learner analysis based on initial interaction with the learner; adapt the instruction to meet the student learning style; monitor the learner progress, providing declarative knowledge when required; decide on the best way to present the next problem or instructional sequence; diagnose problems and provide corrective feedback; oversee the successful completion of the learning process.

In this Chapter the architecture and appropriate diagrams of the Pro-Active Multi-Agent System and the result of research the interaction of Jade agents on Jadex platform will be described within the virtual education purposes.

3. Pro-active approach to virtual education

Modern virtual education allows us to withdraw from strict academic program planning and lesson development in a virtual classroom. In order to improve virtual education and learning processes, the agent system, reacts to the users' actions and supports students in choosing courses or modules of study, Software agents respond to user actions and build a personal trajectory of study modules. There must be a personal agent which controls all user's actions and registers them in the system.

The finite set of agents and message channel preside of student's flow through the learning module. The student has an access to the one learning object agent at the moment and follows to the instructions generated by agents according to beliefs, goals and plans. Under certain circumstances, an agent sends messages to another agent with different aims, e.g. to request information or to redirect the student to the next part of the educational program. The transitions and links between agents form the student's preferred educational trajectory. That helps the student to avoid a static modules sequence precompiled by a teacher. The coordinator agent controls the succession order of transitions. The main goal of the agent is to achieve the best subject coverage in a logical sequence and direct the student to meet course outcomes . The agent of a learning object communicates with the student and passes the contact with the student to other agent(s) to achieve appropriate learning objectives based on the student's knowledge level. In such way one can make nonlinear structure of learning content performance which is adapted to needs and achievements of the student. The basic classes of different e-learning content agents are considered in this Chapter. The basic structure for agent-to-agent and agent-to-student communications during the learning processes is researched as well.

4. Research of agent technology

As far back as in the nineties Pospelov published two fundamental works on the foundations of agents theory (Pospelov, 1997), (Pospelov, 1998). There he examined the historic transition from modeling of collective behavior to agents theory and introduced agents classification with the help of three criteria "type of environment - the level of 'free will' - the degree of social relations development," identified and analyzed the different types of agents operation environment, examined the key intentional agent characteristics. According to Pospelov prerequisites for the realization of a certain behavior for the artificial agent are special devices directly interpreting effect of the environment (receptors) and executive bodies that affect the environment (effectors), as well as processor (information processing unit) and memory. By memory we mean the agent ability to store information about its condition and state of the environment. Thus, the original idea of the simplest agent turns into the well-known "organism-environment" model (Pospelov, 1997).

Intelligent agent, generally, consists of the following components:

- interface - is responsible for agent communication with the environment and consists of sensors and effectors;
- agent database of knowledge - stores all the agent knowledge. It includes knowledge about the interfaces and other agents functionality, as well as current task data;
- scheduler - is responsible for planning further agent activities based on the existing knowledge about the task and environment.

The interface includes various functional units responsible for the interaction of agents and environment. Sensors are responsible for receiving messages by the agent from the environment and other agents (which in the case of the agent autonomy is equivalent to the environment by the manner of interaction).

Effectors, on the contrary, serve as the sender of agent messages to the environment and other agents. All interaction with the network neighborhood and other implementation details are usually not included in the interface, as low-level tools (e.g. agent framework where the system is implemented) provide it.

The knowledge database serves as an agent for storage of all knowledge without exception, gained during the life cycle of the agent. This includes agent modeling database, the knowledge database about the tasks and knowledge base of their own "experience".

The knowledge database of the tasks contains statement of a problem, as well as the knowledge gained in the process of solution. It stores intermediate results of subtasks solutions. In addition, the database stores the knowledge about the task solutions and methods of selection of these solutions.

The knowledge database of the agent's own "experience" contains knowledge about the system, which can not be attributed to the foregoing categories. This knowledge database contains the solutions of previous problems and the various secondary (though, perhaps, useful) knowledge.

The scheduler is responsible for planning agent actions to solve the problem. As noted above, the scheduler must balance the agent activity between the plans construction for solving the problem in a changing environment and immediate implementation of plans (Pospelov, 1998).

All agents can be divided into five groups, according to the type of processing of perceived information:

1. agents with simple behavior - act only on the basis of current knowledge;
2. agents with the model-based behavior - can interact with the partially observable environment;
3. task-oriented agents - similar to the previous type, but they, among other things, store information about the desirable situations;
4. practical agents - recognize only the states when the goal was achieved, and when not achieved;
5. autonomous intelligent agents.

4.1 Agent platforms

An important concept of the multi-agent systems (MAS) implementation is the agent platform (AP) - system that allows you to create and delete, interpret, initiate and transfer agents (Weiss, 2000). Server environment contains agents in certain contexts (rights). For messaging intermediaries are used (to hide the code, have an opportunity to create their own copy or move to a different context).

Communication infrastructure (e.g., RPC) does for agent platform and its context for agent communication. The highest level is the region - set of AP.

Thus, the AP functions are the following: agent’s communication, transfer of messages between agents of different platforms, support of ontologies, agent’s management, search for agents and information about them within the system, management of agent life cycle, and security.

Platforms vary in range of application, technology (language, FIPA organization standards), design community, extensibility (API and plug-ins), integration with corporate systems, documentation, licenses, relations with business corporations and samples of projects.

The preferred platforms for development, standardized by FIPA are: JADE (most popular), Coguaar (no documentation), Aglobe (poorly supports FIPA), Jason (has own language AgentSpeak to describe the agents), Jack (commercial license).

For agent communication modeling MASON, RePast, Ascape, NetLogo are used. In our country, especially for academic and educational purposes, platform of multi-agent programming JADE (Java Agent Development Framework), fully implemented in Java using Java RMI, Java CORBA IDL, Java Serialization and Java Reflection API, is widely used.

JADE provides the agent system programmer-designer with the following tools:

1. FIPA-compliant agent platform - agent platform based on FIPA recommendations, including mandatory types of system agents: agent platform management (AMS - Agent Management Service), a channel of communication (ACC – Agent Communication Channel) and directory services (DF - Directory Facilitator). These three agent types are automatically activated when you start the platform;
2. distributed agent platform - distributed agent platform, which can use several computers (nodes), though each node runs only one virtual machine (Java Virtual Machine); agents are performed as a Java-threads. For agent messaging, depending on their location, the appropriate transport mechanism is used - Multiple Domains support - a number based on FIPA specifications DF agents that can be united to federation, implementing multiple domain agent environment;
3. multithreaded execution environment with two-level scheduling. Each JADE agent has its own thread of control, but also it is able to work in multithread mode;
4. library of interaction protocols - a library of interaction protocols using the standard interactive FIPA-request and FIPA-contract-net protocols. To create an agent that can operate according to the protocols the designer needs to implement only the specific domain activities; all the protocol logic, independent from the application, will be implemented by the JADE system;
5. GUI administrator - an administrative graphical user interfaces (GUI) provides simple platform management and shows the active agents and agent containers. Using the GUI, platform administrators can create, delete, interrupt, and resume agent activities, create domain hierarchy and multi agent DF federations.

JADE simplifies the process of multi-agent system development using FIPA specifications and a number of tools, which support debugging and system deployment. This agent platform can be distributed among computers with different operating systems and can be configured through the remote GUI. The platform configuration process is flexible enough: it can be changed even during the program execution; to do it, you need just to transfer agents from one machine to another. The only system requirement is installation of Java Run Time 1.2 on this machine. The communication architecture provides flexible and efficient messaging, while JADE creates the queue and manages the ACL-messages threads, which are private for each agent. Agents are able to access the queue using a combination of certain modes of their work: blocking, voting, shutdown, and pattern matching (concerning search methods).

Now Java RMI, event notification and IIOP are used in the system, but you can easily add other protocols. The ability to integrate SMTP, HTTP, and WAP is also provided for. Most of the communication protocols that have already been identified by international community of agent environments designers are available and can be illustrated by specific examples, after determining of the system behavior and its major states. SL (semantic language) and agent management ontology are also implemented as well as the support of user-defined content languages and ontologies that can be incorporated and registered by agents, and used by the system. For the purpose to substantially increase JADE efficiency there is a possibility of JESS and CLIPS Java-shell integration.

4.2 Jadex agent development technology

While carrying out the certification paper a number of software tools for the agent realization have been reviewed. Jadex represents a conservative approach to the agents. (Braubach et al., 2009) One of its advantages is that no one new programming language was used. Instead, Jadex agents can be programmed as object-oriented by means of such technologies as IDEs-Eclipse or Intellij IDEA, i.e. in Java. Jadex provides the basic structure and set of programming tools that facilitate agent creation and testing. In order to facilitate a smooth transition from traditional distributed systems to the development of multi-agent systems, there should be used, as far as possible, proven object-oriented concept and technology.

Research project Jadex is implemented by Distributed Systems and Information Systems Group at the University of Hamburg. (Pokahr, 2010) The developed program structure is currently under test. Primary function set is already capable to support the construction process of intelligent agents on FIPA-compliant JADE platform, and the system has already been put into use in some applications. Releases are available under GNUs LGPL license.

Let us present some of the advanced Jadex features. Therefore, Jadex is Java-based FIPA-compliant agent environment, which allows, according to the "Belief-Desire-Intention" (BDI) model (it is shown on Fig 1) the development of target agents. Jadex provides the basic structure and set of development tools to facilitate agent creation and testing. Jadex project aims to maximally simple development of agent systems, which means its Java base. Using Jadex agent system can be created without a preliminary study of a new programming language. Jadex designed to facilitate the implementation of agents in widespread programming language Java, using a huge number of existing tools and libraries.

The aim of the Jadex research project is to create rational layer over the FIPA-compliant infrastructure that provides the possibility of constructing of agents technically based on audio software. To create an intelligent agent one needs to collect some components. Therefore, it is necessary to provide optimal agent architecture, which would take into account intra-agent concepts, agent society, and artificial intelligence. Specific character of this trend progress is that all the most interesting research results are in different fields that are isolated, independent and not joined into a well-organized architecture. Thus, until now, standards have not allowed us to create intelligent agents, considering all aspects. Jadex project provides open scientific information map that, firstly, outlines the fields of interest and, secondly, presents actual groundwork along with studies in progress. Due to the project openness anyone can contribute their ideas and practical improvements.

4.3 BDI architecture

Architecture agents require mental subsystems that help agents to communicate. Based on review the deliberative architecture was chosen, which had requirement functionality. One of deliberative architecture is BDI architecture.

In the development of models and agent architectures BDI architecture dominates (Pollack and Ringuette, 1990); the agent is considered there as a social creature, that communicates with other agents using a certain language, and plays a role in society, depending on beliefs, desires and intentions. It is considered that these three components fully define the state of social agent "intelligence". In terms of programming, BDI-agent beliefs are agent knowledge (information) about the state of the environment, which are updated after each action. Desires designate the goals of the agent, including their priorities. Intentions designate the actions to achieve the goal (behavior samples). Interaction protocols allow agents to reduce the search area of possible solutions and identify a limited range of responses possible for this situation.

In hand-on programming agent is a shell structured computer system, situated in an environment and designed for flexible, autonomous actions in this environment in order to achieve desired goals. Agents alter from tradition software by complexity of interaction and communication scenarios. Ontologies are evident formal term specifications of application domain and the relations between them. This is term of social sciences, used in the theory of agents; it is almost equivalent to the programming concept of language semantics in the application domain.

Example of BDI architecture is presented in Fig.1

Fig. 1. BDI model

The most attractive BDI features are:

1. philosophical components that based on the theory of human rational action;
2. software architecture that was implemented and successfully used in several complex real applications. Examples: Intelligent Resource-bounded Machine Architecture (IRMA), Procedural Reasoning System (PRS);
3. logical components - the model is strictly formalized in the family of BDI logic (Wooldridge and Jennings, 1995). "Base BDI logic", introduced in Rao & Georgeff research (Rao and Georgeff, 1995) is a quantified expansion of expressive logic with time branch.

Intensions roles and features: intentions control mechanisms-outcomes analysis; intentions limit (invalid) reasoning / conclusion; intentions hold to the required direction; intentions affect the presentations, on which the prospective practical conclusion is based.

The agent accepts the obligation to some alternative, if the alternative overcomes the filter successfully and is selected by an agent as an intention. Obligations entail temporal stability of intentions. If the intent was accepted, it should not be immediately rejected. To what extent the agent is obliged to follow its intentions? "Blind commitment," "Single minded commitment," "Open minded commitment." Agent obligations are relates both to outcomes, and to the mechanisms.

4.4 BDI usage in Jadex

"Belief-Desire-Intention" (BDI) - the means of thinking for intelligent agents. The term "intellectual tool" means that it can be used with various kinds of firmware that provide essential services to the agent of communication infrastructure type and management tools. The concepts of BDI-model, originally proposed by Bratmen as a philosophical model for the description of rational agents (Bratman, 1987) , have been adapted by Rao and Georgeff to some more models. They transformed it into a formal theory and model for software agent implementation, based on the concept of beliefs, goals and plans, but it is more suitable for multi-agent systems in software and architectural terms.

The notion of "agent", that is considered as a powerful example of software development, is very useful with regard to the complexity of modern software systems, including Jadex. It allows considering systems as independent interacting objects (entities, modules) that have their own objectives and act reasonably. Consequently, the internal state and the decision making process of agents are modeled intuitively, following the principle of intellectual relation. Focus means that (instead of direct call for agents for the implementation of any action) the developer can define goals that are more abstract for the agents and thus provide certain degree of flexibility to achieve them. BDI model, based on the intellectual relationship, originally was presented as a philosophical model for the creation of reasonable (human) agents. However, later it was adapted and transformed into a model for software agents, based on the ideology of the beliefs, goals and plans. Jadex collates this model into JADE agents by providing first-class objects with "beliefs", "goals" and "plans" that can be created inside the agent and which can be manipulated. (Pokahr et al., 2006)

Jadex agents have possibilities; they can be any type of Java object and be stored in the knowledge base. Goals are explicit or implied descriptions of states that are to be achieved. To achieve its goals, the agent fulfills the plans, which are Java coded procedural means.

Jadex structure consists of API, executed model, and predefined repeatedly used common functionality. API provides access to the Jadex concepts during programming plans. Plans are clear Java classes that expand special abstract class, which, in turn, provides useful methods of sending messages, organization of secondary objectives or events expectation. Plans are able to read and change the agent views, using the API belief base. In addition to Java coded plans the developer provides XML based agent definition file (ADF), which sets out the initial beliefs, goals and plans of the agent. Jadex execute mechanism reads this file, initiates the agent, and tracks its goals during the continuous selection and launch of plan based on internal events and messages from other agents. Jadex is equipped with some predefined functionalities - such as access to the directory facilitator service. The functionality, coded in separate plans, is grouped as reusable agent modules called abilities. Ability, described in a format similar to ADF, can be easily embedded into existing agents.

According to (Pokahr et al., 2006) BDI agent model in Jadex can be represented like this (see Fig. 2). These elements can be specified inside the ADF.

An important qualitative aspect of any development environment is the availability of tools support. For example, Jadex is JADE add-in, and thus a huge number of ready, available tools can certainly be used with Jadex. Nevertheless, this is true not only for tools that are available in JADE (such as Sniffer or Dummy Agent), but also for the third person instruments (such as bean generator plug-in for Protege). On the other hand, new notions

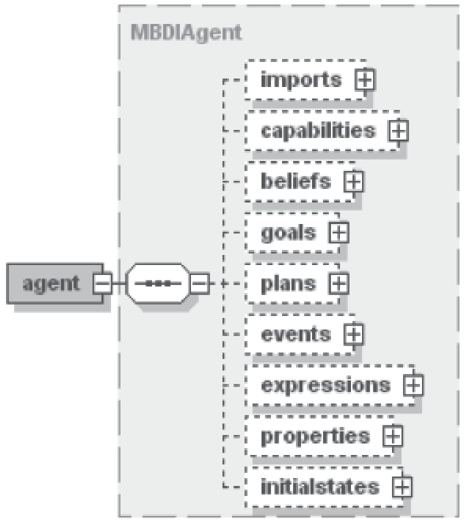


Fig. 2. Jadex agent XML schema

presented by Jadex, must also be supported. Considering the above reasons, the tools were implemented to assist working with these issues. BDI Viewer tool provides an opportunity to consider the internal state of Jadex agent that is its beliefs, goals and plans. The Jadex Introspector is similar to Jade Introspector, which allows to monitor and influence the agent behavior by means of observation and effect the way of managing the incoming events. Introspector is also capable to transfer agent to the step-by-step mode to eliminate errors. In addition to tools, Jadex has Logger Agent, which, if necessary, provides registered JADE and Jadex agents messages collection and demonstration, following the Java Logging API.

5. Multi-agent system for pro-active learning

5.1 Interaction schema

The Pro-active e-Learning Content Model focused on implementation of agent-oriented approach to developing computer-based learning systems. First of all, all learning object connect to appropriate agents. Complete set of the agents compounds learning content (LC) and with agents' messages forms network for a student to glide through it in studying of a subject.

At every instance the student has an access to only one agent of LOs (Learning Objects) and follows instructions generated by the agent due to its own beliefs, goals and plans. When some conditions would be reached the agent sends the messages to another agent(s) with different purposes, e.g. a query for help or to redirect the student to the next part of the learning content. Gliding from one agent to another, forms the students preferred learning trajectory. This enables the student to move away get from a static sequence pre-complied by an instructor. The sequence is controlled by LC Manager Agent. The main goal of the agent is to reach the best coverage of the LC and to assist the student in determining the optimal direction. If the student does take an unexpected direction, the LO's agents jumps in with help of a User Preferences Agent (UP agent). Therefore, in general the goals of LCM and UP agents can be mutually contradictory.

The general ideas of the Flow Control and Reconfigure processes are beyond of the scope of this paper. However, the role of them is very important for developing flexible and user friendly LMS.

Type of interaction between the user and LO agent depends from agents type. A skeleton of the interaction includes: request of the LO content; present the content under the specified conditions; student's feedback; assessment of the students' progress; determining of the next action. Some of these steps may be omitted or altered with concern to the specific of LO agent.

LO agents communicate with each other to send the information about current student's progress and in order to get an additional information and help. Most popular messages are on completion of the LO or reactivation of the some LO agent activity.

Every time when collisions happen in conversation between of LO agents, the LC Manager agent must resolve the problem and re-establish smooth flow of the educational process. So, its goal is not only to meet the educational requirements but also to maintain the order and consistency.

5.2 Types of Learning Object agents

By definition we know that there are many types of Learning Objects. Therefore, we concentrate our efforts on developing Agents only for theory, workshops, assignments, labs, and tests. Learning object may have in its structure some instructional content, glossary of terms, general metadata, quizzes, and assessments by the definition. So, all of this part can trigger different kind of user's action. However, not all of the LOs may include all of the listed elements. For example, simple theoretical notes may include only short portions of the instructional content and quiz with one or two questions, at the same time laboratory works can gather inside some theoretical text, instructions how to perform correct actions, initial data, pre-test, questionnaire, calculations, simulations, interactive models, and so on.

LO Agent for theoretical components of the learning content can be considered as a simple model-based reflex agent. That means it has very simple goals and reactive model of functioning based on condition-action rules. A user may perform some actions which generate conditions for agent's rules activation. Most complex cases of such agent functioning is when a cycle of delivering, showing, explaining, assessing repeats until the agent reaches a satisfactory user knowledge level. The agent can assess student's knowledge of their? content via simple single/multi choice questions.

All lecture notes, vocabulary articles, definitions, glossary terms, comments, online resources, learning films fall under this class of LO.

In addition to studying theoretical issues, discussions between students and teachers play very important role in every learning process. In discussion student may found an answer on questions or develop their own ideas. This process is different form interaction within theoretical components. Thus, another type of agent is needed to assist to that kind of learning activities. There are several triggered actions which can describe the work of users and agents on forums and workshops: Question, Answer, Opinion, Mark, Success, Fault, and Redirect. A sequence of these actions cannot be predefined. Therefore, there is no predefined way to fulfill the goal of such LO. The agent has to make plan and continuously to maintain it with re-planning. Such type of agents, let's call it Workshops LO Agent, could be implemented as utility-based planning agent.

Assignments are a type of the learning activities which stipulate that the student has to do some homework and represent his/her results on a teacher assessment. Generally, a teacher may assign a task which is based on some theoretical LO or is an auxiliary to another LO. The teacher may also propose some script for fulfilling the assignment. The script states the dates, sample sequence, and list of links on readings. The student using all of the presented information completes the assignment and sends it for examination. Assignment LO Agent performs its action in some prescript sequence and may help student to find needed information, teacher to check posted assignments with known digital sources on coincidence, and both to be on the schedule. Thus, based on the analysis of agent's activities we can conclude it might be implemented as goal-based planning agent with library of preprogrammed plans.

During laboratory lessons students gain some practical skills and strengthen theoretical knowledge. With computer modeling we have got very powerful instruments for developing any kind laboratory setup for almost every subject. Labs LO Agent may shadows for student while he or she follows to labs instructions and assesses every single student's action. If help will be needed to student the agent may propose one of the stored standard instructions how to overcome the problem or send notification with request to the teacher. As well as the Assignment LO Agent, this type of agents is implemented in the same way.

Last type of learning activities is testing. We consider only tests with formal response like yes/no, single-choice, multiple-choice, and so on. An attractive feature of formal test is that they are particularly easy to score. Test LO Agent analyses student's results and find gaps in his/her knowledge. Agent's output depends not only from the student's results, but also from a type of the test: preliminary, self-test, monitoring, or final and grade tests. In any of these cases the agent has different type of actions which triggered when appropriate state is satisfied. We believe that this kind of agent can be successfully realized with simple model-based reflex agent.

Accordingly, we propose to develop Pro-Active e-Learning System as an organization of agents like:

- Simple model-based reflex agent for Theoretical and Tests Learning Objects
- Utility-based planning agent for Workshops Learning Objects
- Goal-based planning agent with library of preprogrammed plans for Assignments and Labs Learning Object.

5.3 Standard beliefs, goals, and plans for learning object agents

For defining an agent in Jadex, we need to formalize its beliefs, goals, and plans. Different types of agents listed above have different properties. Beliefs, goals, and plans also may differ from one LO to another. Here we describe most common ones for all types of LO agents.

Begin with goals. The most general goal for all Learning Object Agents is to fulfill a learning sequence and reach most good results with a student. Let's call it "Success" and prerequisites for it are completed goals "Sequence-Finished" and "Satisfactory-Mark". Both of these goals are achieve goals in term of Jadex. For checking of satisfaction of the mark level we add the fact "Current Student Mark" and "Mark Threshold" into agent's beliefs base. For the purposes of determining the level of the mark we also add "Mark Scale" class into agent's believe base. While "Satisfactory Mark" goal is achieved, the Jadex generates an event "Passed".

"Sequence Finished" becomes achieved when every single step of the LO plan has been done and an event "Done" is generated. An array of "LO Steps Completed" with boolean values is included into standard LO agent's beliefs base.

Some LO agent orients only on completion of the LO sequence, while others on reaching of satisfactory student's marks. We include "Priority" fact in belief base for dealing with importance of the goals.

A core goal for every single LO agent is "Show content" goal. This goal is a maintain goal type and must continuously to reestablish the "Show on Schedule" and "Student Will" conditions. Plans which maintained the goal use following facts of belief base: "Content.xml", "Glossary.xml", "Quizz.xml", "Sequence", and "FAQ.xml", and also can send event messages "Step Done: X" and "Mark is changed: +/-X".

Under some circumstances, achieving of a goal "Need Help" may be initiated. It can be doing by student itself or as a result of maintaining another goal – "Monitoring Student's Actions". Agent generates "Help" event message by using "List of Forced Errors.xml" and "Unforced Error" elements of belief base. "Need Help" goal's plans do a conversation with another LO agents in order to find a solution.

LO Agent during the whole lifecycle has to keep student's history. These data are stored into "Student's Behaviour History.xml" fact of belief base.

The last general goal for all types of LO agents is "Restart" goal. This goal operates with following facts: "Open Date", "Close Date", "Schedule", "Started Date/Time", "Restart from", "Student.xml". Achieving of the goal results in the agent drops all previous student's results and advancements (with exception of "Student's Behaviour History") and starts its function again (or from some point).

The "Restart goal" achieves at the beginning of LO Agent work or as a result of sending "Restart Forced by Student/Teacher" event message from "Monitoring Student's Actions" goal.

As output for LO Agent we define two types of ACL messages. First one is "Help Message.acl" and as it mentioned above it is used for establishing communications with another agents with the object to get an additional information to the student. Another one is "LO is Complete Message.acl". Agent sends this message when the "Success" goal is achieved (even with negative result).

For defining agent in Jadex we need to create XML agent definition file and Java classes for plans implementation. In Jadex if we want to define some abstract agent definition we need to use a capability definition. Some extraction from LO Agent capability definition is shown on the Figure 3.

Developing of the more specific types of LO agent has to be started with including a reference on the capability defined for the abstract LO agent.

5.4 Elements of multi-agent system for pro-active learning

Multi-agent systems are the system of different kind (software or hardware) in which a set of intellectual agents interact to each other, have some particular sets of goals and try to fulfill some sets of tasks. The finite set of agents and message channel preside of student's flow through the learning module. The student has access to the one learning object agent at one moment of time and follows to the instructions generated by agents according to her believes, goals and plans. Under some certain circumstances, an agent sends message to another agent with different aim, e.g. to request for an information or to readdress the student to the next part of learning educational program. The transitions and links between agents form the student's preferable educational trajectory. That helps to the student to avoid a static sequence precompiled by a teacher. The coordinator agent controls the

succession order of transitions. The main goal of the agent is to achieve the best covering of logical channel and to show the student on relatively optimal paths.
Interaction between user and agent depends on type of the agent. The skeleton of the interaction includes: a request of distant module content; its display to the student; an assessment of student progress; a determination of the next action. Some of these steps may be skipped or changed depending on the LO agent type. Most volume of messages concerns with educational séance completion or actuation of some particular agent's activity. Whenever collisions happen in communications between LO agents, the coordinator agent must resolve this problem and re-establish the smooth flow of educational process. It is so, because its goal is not only an execution of educational requirements but a support for order and sequence.

```
<capability xmlns="http://jadex.sourceforge.net/jadex"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://jadex.sourceforge.net/jadex
            http://jadex.sourceforge.net/jadex-0.94.xsd"
 abstract="true"
 package="jadex.planlib"
 name="Learning Object"
 package="ctde.jadex.paelc.lms">
 <beliefs>
  <belief name="current_student_mark" class="long">
   <fact>0.0</fact>
  </belief>
  <belief name="mark_threshold" class="long">
   <fact>60.0</fact>
  </belief>
  <belief name="mark_scale" class="long[]">
   <fact>{60.0, 66.0, 75.0, 90.0, 96.0}</fact>
  </belief>
 ...
 </beliefs>
 <goals>
 <!-- Restart LO Sequence. -->
  <achievegoal name="restart">
   <parameter name="open_date" class="Date">
    <bindingoptions>$beliefbase.openDate</bindingoptions>
   </parameter>
   ...
  </achievegoal>
 ...
 </goals>
 <plans>
 <!-- Send Help Message. -->
  <plan name="help_message_conversation">
    <body>new HelpConversationPlan()</body>
    <trigger> <goal ref="need_help"/> </trigger>
```

```
  </plan>
 ...
 </plans>
 <events>
 <!-- Student attended all LO Sequence. -->
  <internalevent name="done"/>
  <messageevent name="LO_is_complete" direction="send" type="fipa">
   <parameter name="performative" class="String" direction="fixed">
    <value>SFipa.INFORM</value>
   </parameter>
   <parameter name="language" class="String" direction="fixed">
    <value>SFipa.JAVA_XML</value>
   </parameter>
  ...
  </messageevent>
 ...
 </events>
</capability>
```

Fig. 3. Learning object agent's capability example

E-learning module may contain lectures, labs, tutorials, glossary items, tests, and some other type of resources in its structure. All of these types of LO can evoke different kind of user's actions.

A content agent (for example for a lecture LO) may be considered as a very simple one. That means he has very simple goals and model of quick functioning based on the learning rules. User can execute very simple actions which initialize conditions for firing agent's rules. The most complex case of the agent functioning is when the cycle of delivering, display, explanation, and assessment achieves of satisfactory level of the student's knowledge. The content agent can assess this student's knowledge via testing with or without a help of testing agent.

During laboratory works the students get some practical skills and improve their theoretical knowledge. So, the labs LO agent has to help to a student to do that. Every time when a student follows to the instructions of labs LO agent, the agent assesses his action. Whenever the student needs help, the content agents may provide one of the stored procedures which could help to overcome the unresolved problem or with the aid of the chat agent to send a message with a help request to a teacher. The student also may get an auxiliary help about interesting concept or phenomenon from context sources like glossaries and thesaurus by means of a context help agent even without involving of the teacher or tutor.

The resource agent is responsible for management of a search, selection and delivering of learning objects to the student. Its functioning is based on the operating with LO metadata which are compound into ontological model of learning resource.

The personal user agent is in action for establishing full-scale communication between the user and e-learning system. The basic task of the agent is maintaining a user profile with constantly incoming data (personal and educational).

Thus, for discipline and maintaining of process of active e-learning we are proposing to develop multi-agent system model which would include all of described above intelligent agents. A diagram of interaction between the agents is shown on the Figure 4.

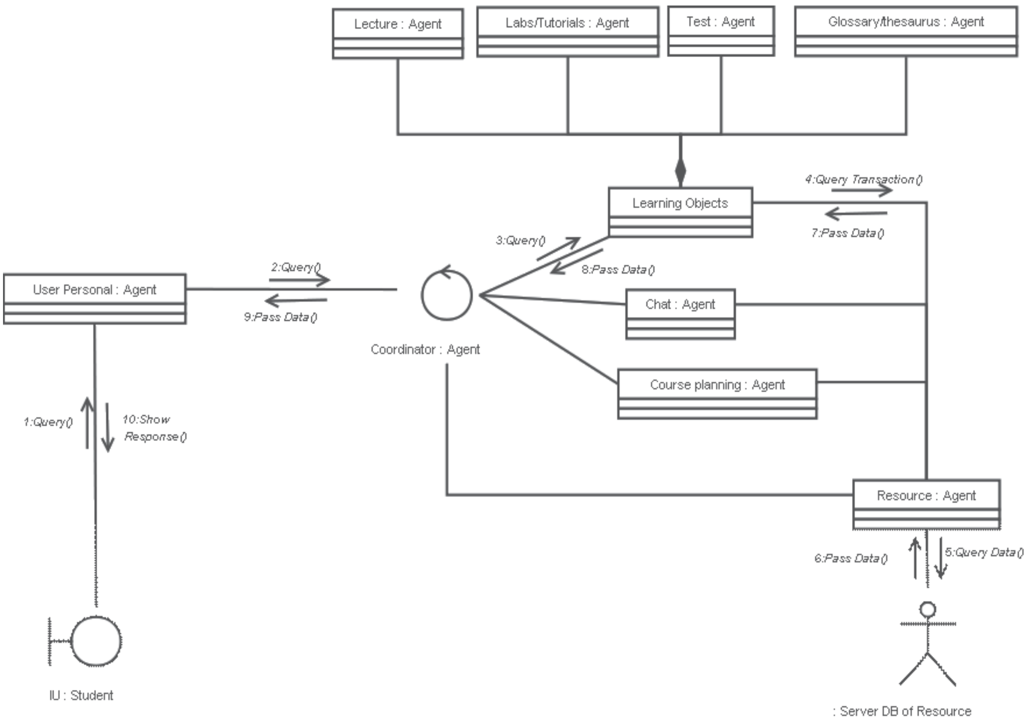


Fig. 4. Agents Collaboration Diagram for Multi-Agent Active e-Learning System

One can see the diagram of flow control sequence in proposed system on the Figure 5. The system contains following agent groups: 1. agents of communication: the coordinator agent, the resource agent, the chat agent, the user personal agent; 2. content agents: the lecture LO agent, the labs/tutorials LO agent, the test agent, the glossary/thesaurus agent. This set of agents is not comprehensive with listed kinds of agents and may be expanded. This expansion may be done on the basis of requirements analysis and need in enlargement of e-learning system functions.

As we mentioned above, we are supposing to use Jadex extension for JADE platform for developing the proposed system. Agents in Jadex are described with lists of believes, goals, and plans (Pokahr et al., 2006) .

All of listed agents have different properties and thus different believes, goals, and plans. For agent definition in Jadex one has to create two types of files: ADF file (Agent Definition File) – XML file with complete definition of agent and Java class files for execution of agent's plans.

Resources Agent. Allowing to obtain the search, selection and delivering of LOs to the student based on the ontological model. This model contains metadata elements for LO in accordance with LOM standard. Goals: main goal – data search (QueryInform) in metadata due to desired request from the coordinator agent, processing of query from the coordinator agent, transfer of discovered information (TransmitInform). The agent has five plans: Restart – resources agent restart plan; Search – requested data search plan; Message – plan for sending an error message to the coordinator agent in case of data not found; Transmit – result data transferring plan; Sleep – plan for sleep mode change-over.

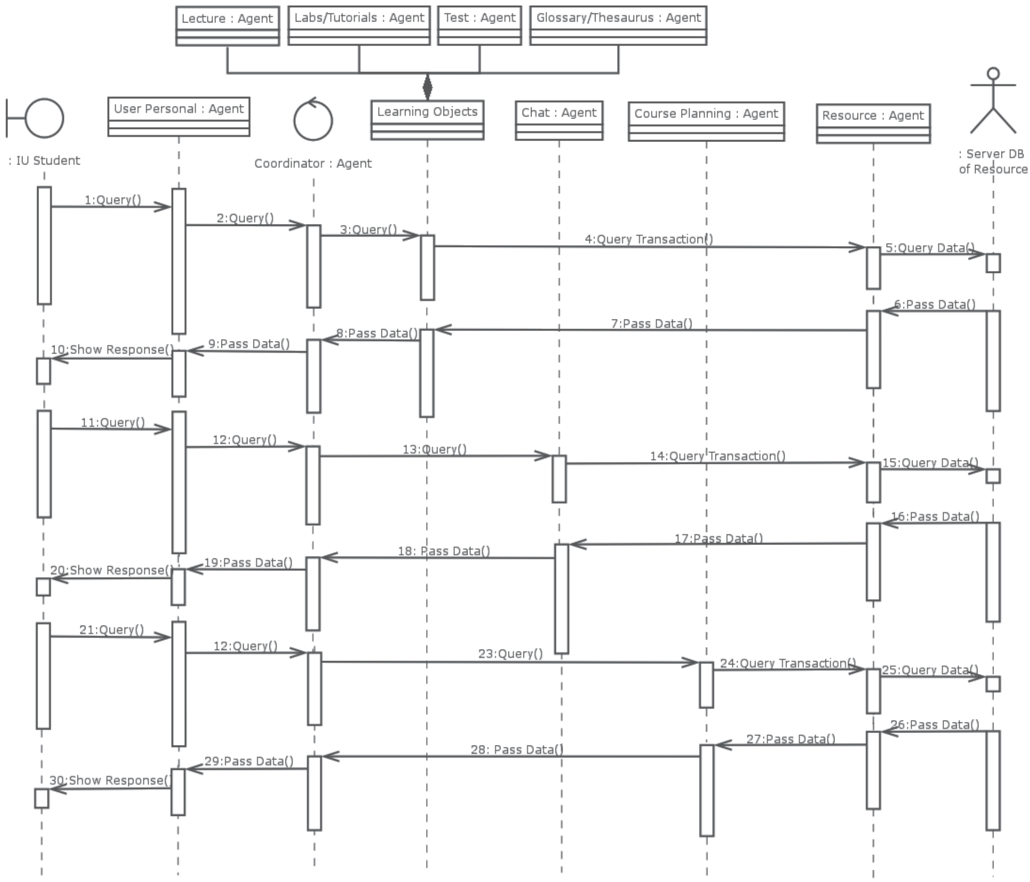


Fig. 5. Sequence Diagram

Agent's knowledge. Restart belief fact is setting true after the agent's start. QuerySearch fact shows a presence of data search query. InformNotFound represents a state after sending the error message to the coordinator agent when the search failure occurs. TransmitInform is a fact on success of information searching and of sending these data to the coordinator agent. SleepMode fact defines the moment when the agent moves to the sleep mode. That may happen if found information has been sent to the coordinator agent or the error message has been sent. Definition of the Resources Agent consists of two xml files: ResourceAgent.xml – agent's definition and Metadata.xml – e-Learning module metadata. The last one contains description of whole module content, locations of separate module elements, and short description of lectures, labs, tests, and other Los (Bratman, 1987) .

User Personal Agent. We introduce the user's personal agent the main goal of which is to maintain a user's profile and to support interaction of a user and the system. The agent collects all incoming information about the user and his activities. On the basis of these data it builds the user's model. Goals: authentication of a user; users questionnaire preparing based on the ontological model of the learner with the aim of getting personal preference; information sorting and structuring in accordance to the conceptual model of learner; educational goals tree forming, the tree consists of hierarchy of educational material subjects and corresponds to the subject area classification. Each user at the first moment has to be registered in the system before starting his educational activity. That means he must obtain the login and password for system access and his personal account in a database has to be created. The agent provides an identification of the user on login and registers all his actions into the database.

The Course Planning agent. It executes the specific tasks directly related to the course planning by using agent-specific aspects, which details are given below.

System access is implemented through registration. There are two kinds of users in the system - an administrator and a student. Each of them has certain rights to access. Administrator can add or delete courses while student can choose courses or change selected courses if they are not started yet. The student can select available courses using a number of possible features: curriculum and student course; keywords; area of interests for the student (subject); course level (A, B, C).

Agents are active objects, which, unlike general (passive) objects they are not «sleeping » until they receive the next message (from user or environment) and perform it, but permanently operate solving the assigned tasks. Courses planning system can be supplemented by other agents, built on the principle of courses agent, such as coordinator agent or scheduling agent. Agents working in the system must ensure their proper functioning by adding resources of JADE-based agent technologies. The agent allows selecting for each student individually the courses for studying. The agent shows courses accessible to the concrete student (student interests, course level, and already completed courses are considered), allows to select several courses for learning (no more than four), it counts the credits for the selected courses. Also an available number of students on the selected course takes into consideration. After courses selecting the agent records it in individual sheet in which they are put in queue to learn. In this sheet it is possible to view when the course was started and whether it was finished.

Beliefs: User_Name; User_Password; User_Id; Max_Credits – default credit points equal 30 (basic knowledge). Goals: the main goal is to search courses based on student's preferences and education level.

Agent needs plans to work in an environment. With the help of certain expressions (or without its) the events are handled in the body of a plan; triggers, parameters, messages or purposes can be used here. Plans: Restart plan executes a new start of the agent after requesting from the coordinator agent; SignIn - extracts user data from database; Filter - allows user to choose different filters before showing available courses; SearchAvailableCourse displays a list of available course for concrete student taking into account user's filters; CheckCourses plan checks the correctness of selected courses (total number of credits, number of courses etc); AddCourses - allows adding courses to the student timetable; ListBadCourse plan allows user to exclude uninteresting courses from list of available courses; StartCourse plan looks at start and finish of each course for each student; CoursesList - displays list of all course (this plan is available only for administrator access); AddNewCourse - allows the administrator to add a new course to the courses list or edit data; DeleteCourse plan allows the administrator to delete course from courses list.

The Coordinator Agent. The agent is used for distribution of actions between agents and supporting them with necessary information. All information is stored in a database and marked up XML files on the e-learning server. An access to the files and database is accomplished by mean of inter-agent queries. Goals: Events monitoring; Selection of a module (course); Course material loading and analysis; LO agent initialization; Logging. Plans: display authorization form and get user information, and store it in beliefs set; load a courses to appropriate belief; start the system and create the first event (Authorization, CourseInit), load a course content to appropriate belief; authorization; user's profile loading; get the list of course; display the course selection window; load the course; load the agent; get the resources; send a message; execute an application. Beliefs: User's profile, LO studying in progress, Current element of the course, Course structure, Course metadata files path.

The Glossary Agent. The main issue for the agent development is to support students with a context help. There are two main goals of this agent: Query and Main goal. The Query goal consists in displaying of information necessity, carrying out of condition search. The main goal is satisfied after displaying to the student all needed information. Beliefs: Restart agent fact; QuerySearch – search information query performance fact; ContentOpen – opening the content from a vocabulary fact; TermNotFound – a message on error would be sent to the coordinator agent; NoTerm – the fact about absence of necessary information and query to the teacher to enter the definition; SleepMode – the agent is in sleep mode fact. Plans: Restart, Reference, Search, ContentOpen, Coordinator, Chat, Student, and Sleep: the Restart plan execute a new start of the agent after a call from the coordinator agent; the Reference plan determines of hyperlinks existence in a text; the Search plan seeks for looking term; ContentOpen – displaying of vocabulary entry; Coordinator – sequence of communications with the coordinator agent. The agent performing this plan sends an error message on term not found. It is an external plan. Chat is an external plan for communication with the Chat Agent. It sends a message with info that there is no required information; Student – this plan display a message about incorrect link to the student; Sleep – after fulfilling this plan the agent falls into sleep mode.

The Lecture Agent. The agent supports the process of lecture notes reading by the student.

Goals: Content displaying; Monitoring of the lecture window scrolling; Timing of the lecture reading; Putting testing questions; Getting a message about termination. Plans: Check if the student's name exists in the database, Load the lecture, Control of the reading time, Send a message to the helper, Display questions, Save the result. Beliefs: StudentEnrolment, ScrollPossition, ReadingTime, FinishButtonPushed, CheckFinalMark.

The Test Agent. The student may check the level of his knowledge after finishing some part of the educational material by means of the test agent. The agent may be summoned by the coordinator agent as well as any other LO agent directly. Goals: Run test on a chosen topic; Display test questions, Time control of testing; Calculate the score, Summarize the test result; Analysis result. Beliefs: Answer to a question is available; End of test; Test beginning; Screenshot Saving; Rating scale correspondence; Test time-out; Analysis of testing results. Plans: set up the subject at the test beginning; check whether the answer question is correct; assessing the mark; control of the test flow in depending on the test type; number of gathered points; time-out; the complexity of a question, and some another parameters.

The Labs Agent. The main issue of the agent is to maintain the process of laboratory works and tutorials carrying out by the student. Another goal is to give the students an opportunity to pass labs in their wished order with or without binding to the course content structure and sequence. Goals: to show labs content; to save the completed labs results; to send the report to the teacher. Beliefs: labs subject; labs sequence; labs tasks; an environment description; time for completion; acceptable idle time gap; labs report; address for sending the report. Plans: sending the subject for the coordinator agent; making the report for the labs; monitoring of the current time; checking of idleness time.

The Chat Agent. This agent is designed for supporting of the users' communication. The students may send some messages with their question to the teachers in case of needs and problems. Another useful application of the agent is to communication between the student and the agent as with a chat-bot, e.g. the student may ask some general questions or questions with formal answers. The chat agent consists of two modules: the communication module and the users' support module which works in the tutor absence. The communication module uses the text messages delivering service by establishing dual connection. By responding on requests the agent processes an event and makes a decision on establishing channel with the tutor. If the tutor is absent, then the agent passes control to the coordinator agent. The support module waits for receiving questions and then chooses an answer for them. The decision of the support modules is based on the ontological model of a dialog. This model includes concepts which are typical for questions and answers. The module compares request in question and templates in knowledge base and chooses an appropriate answers options. Then it checks the value for chosen answers and tries to find out which one is more applicable to the user's question. The agent allows to hold a conversation in user's language and to keep to main stages of common scenarios for conversation in e-Learning environment. Goals: Restart and ReplyBot. The first goal Restart is invoked when the request to the chat is initiated. The ReplyBot goal continues the function of the first goal and is applied when the user tries to connect to the tutor in off-line mode. Plans: For fulfilling the Restart goal the initialization plan is used. The plan shows the dialog window, loads the list of users by querying the coordinator agent and sets up a trigger for execution of chat-bot. A plan for achieving the ReplyBot goal is at finding of correct answer on asked question with help of ontological model. Triggers for query response are used for maintaining of conversation. In appropriate cases the plan uses agent's knowledge base and discovers what kind of answer the bot has to send to the user supplying with a virtual environment for users support.

6. Conclusion

The concept developing a Multi-Agent System for Pro-active Learning was promoted in (Grebenyuk and Repka, 2006). Since that the elements of this Multi-Agent System have been further developed and approbated. (Grebenyuk et al., 2006), (Repka et al., 2008) and others.

In proposed Multi-Agent System the intellectual agents interact with each other and allow the process of the active e-Learning. It means that the student is not anchored to a specific structure and sequence of the course learning, established by teacher. The student has an opportunity to learn the content in accordance to personal preferences, abilities, year level, and previous experience in using of distant education/learning systems.

The multi-agent system is able to self-learn from user responses and events during their course, and then determine and recommend help, thus, as much as possible simplify operations of the teacher and transfer this role from teacher to the tutor. In such a system the main role of the teacher is to prepare the qualitative learning material, and all teaching supporting processes will be conducted by intellectual agents.

To develop learning objects IEEE 1484.12.1-2002 Standard for Learning Object Metadata must be implemented. In the latest release of Multi-agent System one more agent was implemented – the Webbridge agent. (Kliushnyk, 2010) Webbridge enables agents to work through the browser, that standard Jadex features do not allow. The main goal within Webbridge is to support the communications of a user with the applied agents.

Webbridge allows combining the capabilities of agent technologies with the capabilities of web applications. Webbridge is a "glue tier" therefore it distinguishes between details of the planning agent and the web layer. By developing a system using Webbridge we can concentrate on developing agent-based business logic, and not on sending or receiving data to the browser. This allows you to develop a system on the base of MVC pattern, as Model 2 architecture. In Model 2 the data model is stored in data bases and the data are in Java beans for transmission and presentation. The accent in this architecture is fixed on the Controller, which is responsible for data transferring between the browser and agents.
Thus the proposed architecture of Pro-Active Multi-Agent System can be extended to any new types of agents in accordance with requirements of virtual education environment.

7. References

Bellifemine, F. L., Caire, G., & Greenwood, D. (2007). *Developing multi-agent systems with JADE*. Chichester, England ; Hoboken, NJ: John Wiley.

Bratman, M. (1987). *Intention, plans, and practical reason*. Cambridge, Mass.: Harvard University Press.

Braubach, L., Petta, P., Hoek, W., Pokahr, A., & SpringerLink (Online service). (2009). Multiagent System Technologies 7th German Conference, MATES 2009, Hamburg, Germany, September 9-11, 2009. Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg.

Burkhard, H.-D. (2007). *Multi-agent systems and applications V : 5th International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2007, Leipzig, Germany, September 25-27, 2007: proceedings*. Berlin, New York: Springer.

Fuhua, O., Lin (Ed.). (2005). *Designing distributed learning environments with intelligent software agents* Idea Group Inc

Grebenyuk, V., & Repka, V. (2006). *Pro-active E-Learning*. Paper presented at the International Conferences for Internet Technology and Secured Transactions, September 11-13, 2006, London, Great Britain.

Grebenyuk, V., Repka, V., & Shatovska, T. (2006). *Multi-Agent Systems for Providing a High Quality Distant Learning*. Paper presented at the International Conferences for Internet Technology and Secured Transactions, May 21-23, 2006, London, Great Britain.

Kliushnyk, K. (2010). *Research of model communication between JADE agents on the Jadex platform*. Linneus University, Växjö.

Lin, H. (2007). *Architectural design of multi-agent systems : technologies and techniques*. Hershey: Information Science Reference.

Nguyen, N. T. D. S., Kowalczyk, R., & Chen, S.-M. (2009). *Computational collective intelligence: semantic web, social networks and multiagent systems. first international conference, ICCCI 2009, Warsaw, Poland, October 5-7, 2009. proceedings* (1st ed. ed.). Berlin: Springer.

Pechoucek, M., Petta, P., & Varga, L. Z. (2005). *Multi-agent systems and applications IV : 4th International Central and Eastern European Conference on Multi-agent Systems, CEEMAS 2005, Budapest, Hungary, September 15-17, 2005: proceedings*. Berlin ; New York: Springer.

Pokahr, A. (2010). Jadex Software Projects.

Pokahr, A., Braubach, L., & Walczak, A. (2006). Jadex User Guide, Vol. 2010: Distributed Systems Group University of Hamburg, Germany.

Pollack, M. E., & Ringuette, M. (1990). *Introducing the Tileworld : experimentally evaluating agent architectures*. Menlo Park, Calif.: SRI International.

Pospelov, D. (1997). From collective of automata towards the multiagent systems. *The distributed artificial intelligence and multiagent systems*: 319-325.

Pospelov, D. (1998). Multiagent systems - today and the future. *Information technologies and computing systems*, Vol. 1: 14-21.

Rao, A., & Georgeff, M. (1995). BDI Agents: from theory to practice. *Proceedings of the First International Conference in Multi-Agents Systems*, San Francisco, USA.

Repka, V., Kliushnyk, K., & Kozopolyanska, A. (2008). The Agent Model of Testing Student Knowledge in Distant Learning System. *Bulletin of Kherson National Technical University*, Vol.1(30): pp. 417-421.

Samuel, P. (2007). *E-learning networked environments and architectures: a knowledge processing*: Springer.

Shih, T. K., & Hung, J. C. (2007). *Future directions in distance learning and communications technologies*. Hershey, PA: Information Science Pub.

Singhal, S., & Zyda, M. (1999). *Networked Virtual Environments: Design and Implementation*. New York: ACM Press.

Uhrmacher, A., & Weyns, D. 2009. *Multi-agent systems : simulation and applications*. Boca Raton: CRC Press/Taylor & Francis.

Weiss, G. (2000). Multiagent systems a modern approach to distributed artificial intelligence: 648 p. Cambridge, Mass.: MIT Press.

Woda, M., & Michalec, P. (2005). Distance Learning System: Multi-Agent Approach. *Digital Information Management* 3(3): 198-201.

Wooldridge, M. J. E., & Jennings, N. E. (1995). *Intelligent agents : Workshop on agent theories, architectures, and languages : Papers*: Springer-Verlag.

21

Simulating a Land Development Planning Process through Agent-Based Modeling

Michael Kieser[1,2] and Danielle J. Marceau[2]
[1]*Tesera Systems Inc., Cochrane, Alberta,*
[2]*Department of Geomatics Engineering, University of Calgary, Calgary, Alberta*
Canada

1. Introduction

Like most urban centers in North America, the City of Calgary has been experiencing steady population and unprecedented land-cover growth over the past six decades due to the strong Alberta economy, natural increase, and net migration (City of Calgary 2009). Between 1951 and 1961 the population exploded by 93% from 129,060 to 249,641 inhabitants and the city's jurisdiction swelled by 276% from 104 to 392 km^2. In each decade since, the population has increased by approximately 35% to the current population of 1,043,000 inhabitants and the city's municipal lands have expanded by 14% to the current area of 745 km^2 (Applied History Research Group 1997-2001; City of Calgary 2008). The City of Calgary's "Population Picture" predicts a population of 1.6 million by 2037 and just fewer than two million inhabitants by the 22nd century (City of Calgary 2009). If these population predictions are correct and planning and land development decisions continue to be made based on current planning policy and developers' self-interest, the City of Calgary will continue to sprawl and housing costs will continue to climb.

This competition and cost increase for land within the City of Calgary force young families unable to afford a house in the city to purchase outside of the city and commute to their place of employment. It also encourages retirees wanting to get away from the city to move out to the country. This creates demands on towns located in the vicinity of Calgary to accommodate this net migration. This is the case for the Town of Strathmore, located at about 40 km east of Calgary on the Trans-Canada Highway. It currently has a population of 11,335 inhabitants and covers approximately 15.5 km^2. The population is expected to reach 56,731 inhabitants by 2056 and to cover 26.8 km^2, an increase of 38% in population and 12% in area per decade (Brown and Associates Planning Group 2008).

Several environmental, social and economic problems are associated to population growth, including: (a) increased demand on and cost for resources such as land and water; (b) increased intensity of use on and competition for land; (c) change in settlement patterns; (d) increased interaction, and conflict or required cooperation with adjacent municipalities; (e) increased demand on existing infrastructure, such as roads, utility distribution, collection and treatment facilities; (f) increased cost for new infrastructure like roads, utilities, schools, and other community facilities; (g) increased environmental ground, water and air pollution; and (h) increased health and emergency costs. Some of these costs can be avoided;

some cannot, but most can be reduced. This brings several questions to mind: (1) who is making the planning policy decisions?, (2) what are the goals and objectives of the decision makers?, (3) how are the decisions makers interacting when making their decisions?, (4) do the decision makers know the impact of their decisions?, and (5) do the decision makers have the tools to: (a) predict the future impact of their decisions should they continue making similar decisions, and (b) predict the future impact of their decisions if they were to change their goals?

The development of a parcel of land for residential purposes requires the planning of the physical and legal changes to the land. The physical changes include designing the size and configuration of the proposed site infrastructure such as site grading, road design, storm water control system, and the servicing of water, sanitary sewer, electrical, communication and gas utilities. In Alberta, the documents regulating the legal changes include subdivision and land-use redesignation. Land-use is regulated through the following documents in hierarchical order: the Alberta Municipal Government Act (MGA), the Inter-Municipal Plan (IMP), the Municipal Development Plan (MDP), planning documents of the municipality, Land-use Bylaw, and Land-use (LU) Redesignation (Fig. 1).

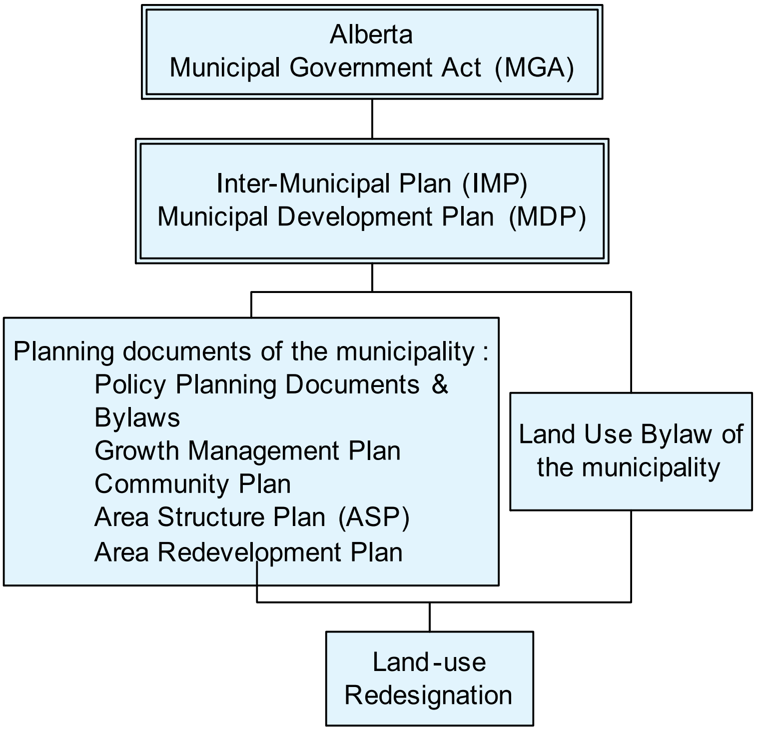


Fig. 1. Hierarchy of planning documents regulating land use in Alberta

The following is a summary of the Land-use Redesignation and concurrent Outline Plan process, from here on referred to as the *land development planning process,* in the City of Calgary assuming a MDP and an ASP are in place (City of Calgary 2008). This process is similar in the Town of Strathmore. However, the administration is smaller and is not divided into as many individual departments resulting in fewer decision makers making more decisions. First, the landowner and/or developer attend a pre-application meeting with the city planning authority to discuss the proposed redesignation. Then, the landowner/developer voluntarily presents the plan to the neighbours and local community association. The landowner/developer submits the application for land-use redesignation and outline plan that he has worked on with professional planning and engineering consultants and sub-consultants, to the planning authority. Then, the application is circulated to the various City departments, the community association, the Alderman, and any applicable special interest groups and a notice is posted. The various planning authorities review the application and the comments and make a recommendation to City Council. Notices are sent to adjacent owners, a sign is posted, and an advertisement is placed in the local newspapers regarding a public hearing on the application. Finally, the proposed land-use redesignation and outline plan application are presented in a public hearing and the City Council makes its decision.

As one can see, this process involves many stakeholders including: the landowners, the developers and their engineering and planning consultants, the city/town authorities and their various departments, the neighbours and other citizens, the community associations, the city/town political figures, the utility providers, and other special interest groups. One can imagine each stakeholder or group of stakeholders having different visions, opinions and interest in the proposed development. The authorities must also have in mind the broader picture and overall goals of the city/town itself, and attempt to see how the proposed land development fits into the future plans of the community. Throughout the process, communication occurs between different stakeholders on many occasions, both within an organization and with other organizations. Communication also occurs at many formal and informal levels, including: pre-application meetings with city/town authorities, meetings with neighbours and community associations, meetings with private planning and engineering firms, open houses, application reviews by authorities and the public, decisions by city/town authorities, public hearings, and possible appeals. Prior to these communication sessions, stakeholders are trying to devise ways to fit their goals into the proposed development. During the sessions, negotiations take place to balance goals and resolve issues and hopefully make decisions. As stakeholders make decisions, they might weigh social need, environmental impact, economic advantages or disadvantages and political support or opposition of a proposed land development.

Municipal and inter-municipal planners use various methods and tools to create a municipal plan that best suits the ideals, values and vision of the community in terms of future social need, economic feasibility, and environmental sustainability. These methods include forecasting based on present conditions using historical data, past success and failures. Some of the tools include statistical census analyses, and community economic models to predict employment creation. The growth plans and planning policy developed for a municipality set the direction of the community growth, which may leave limited choice for developers and citizens. One of the outcomes, which is the focus of this research, is a decision that changes land use allowing for development to occur on a particular parcel of land. Despite the array of methods and tools available, these decisions are still often made in the face of uncertainty. The central issue addressed here is that town planners have a limited, although improving, ability to forecast the cumulative effect of individual decisions made by stakeholders having different goals on the overall environment over which they make their decisions. They need a tool that can model how environmental patterns and trends emerge from the intricate interactions and complex behaviour of several stakeholder groups who might have conflicting goals and views. Having access to such a tool would enable environmental impact forecasting of current goals, decisions and policies, and would allow stakeholders to perhaps modify their goals and analyze possible future impact before the implementation of their decision. Increasingly, computer simulation models, such as agent-based models (ABMs) are being used to support decision making in complex environmental management situations (Marceau 2008). The *land development planning process* is the type of complex systems where ABM can provide this support.

ABMs are an abstraction of real-world entities called agents having typically the following properties: they are autonomous, they control their own decisions and actions; they are social and can negotiate and cooperate with one another; they are able to perceive changes in the environment and react to them; they have goals and are able to take initiative to achieve them (Wooldridge 2000). ABMs are typically discrete, disaggregate, dynamic and spatially explicit, meaning that they simulate the processes that occur over time between individual agents that interact and act upon a simulated geographic region. Over the past fifteen years, ABMs have contributed to modeling in the natural and social sciences in the areas of human/wildlife interaction (Bousquet et al. 2001; Anwar et al. 2007; Beardsley et al. 2009; Bowman and Thompson 2009; Musiani et al. 2010), human/landscape interaction (Gimblett et al. 2001; Gimblett et al. 2002), urban pedestrian movement (Batty 2001; Waddell 2002), water/forest/agriculture resource management (Janssen et al. 2000; Feuillette et al. 2003), spatial planning (Ligtenberg et al. 2001), and land-use and land-cover change (Lim et al. 2002; Parker et al. 2002; Monticino et al. 2007; Moreno et al. 2007; White et al. 2009). When used for spatial planning, ABMs are often linked to a cellular automata model (Parker et al. (2002). In such a case, the ABM component represents humans making decision and interacting over their environment as agents. The cellular automata component is a cell-based map that simulates the environment that agents view and act upon.

The objective of this research is to develop an ABM to simulate the land development planning process in a particular case study, which is a proposed residential subdivision in the Town of Strathmore called Strathbury. The land development planning process includes the Land-use Redesignation and Outline Plan process as shown in Fig. 1. The model will then be used to investigate the impact of changes to governmental regulations, planning policies, design standards and stakeholder goals on land-use resources. For the purpose of this research, land-use resources are defined as parcels of land having a potential for development that currently do not have the land-use designation to allow for development but that could be redesignated.

2. Methodology

The following eight steps in creating the ABM (Kimmins et al. 2004, Wainwright et al. (2004) have been applied in this research: (1) identification of the study area, (2) abstraction of the real-world system through a conceptual model, (3) collecting the information needed for the implementation of the model, (4) implementation of the model, (5) the computational logic of the model, (6) calibration and verification of the model, (7) scenarios simulation, and (8) validation of the model results. They are presented in details in the following sections.

2.1 The study area

The study area is a proposed residential land development project called Strathbury corresponding to 80 hectares of undeveloped piece of property located at about 0.5 km northwest of the downtown core of the Town of Strathmore; it is within the adopted Strathmore Lakes Estates Area Structure Plan (ASP). The simulation scenarios were tested over an area of approximately 3000 hectares that includes the Town of Strathmore and 1.6 km of the surrounding Wheatland County (Fig. 2).

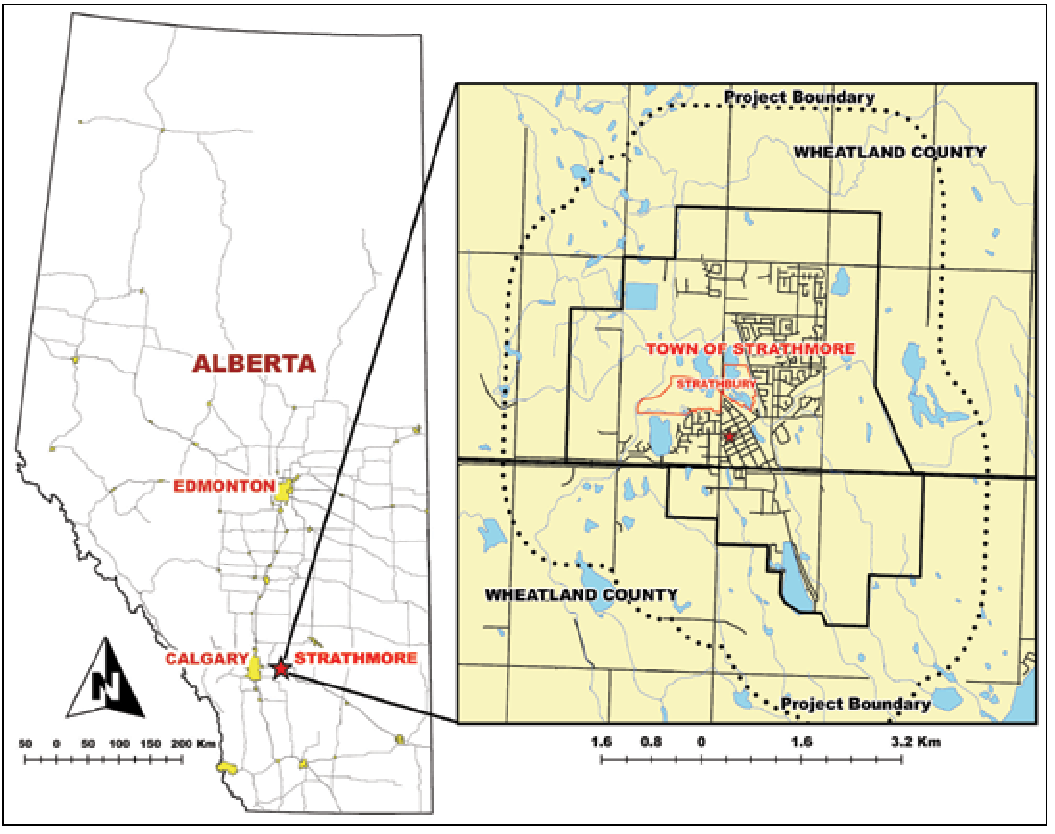


Fig. 2. Location of the study area; the dashed line represents the extent of the model simulation; the continuous line represents the Town of Strathmore boundary

2.2 Abstraction of the real-world process

The conceptual model, expressed as a UML diagram (Fig. 3), displays the two model components, the agent-based model and the raster-based land-use change model that were built to simulate the *land development planning process*. The agent-based model simulates the planning process including the goals, interactions, and decision making of stakeholders, taking into account economic factors, social factors, and regulations. The raster-based land-use change model simulates the environment that is deliberated, applies the land-use change from an approved development, and simulates into the future.

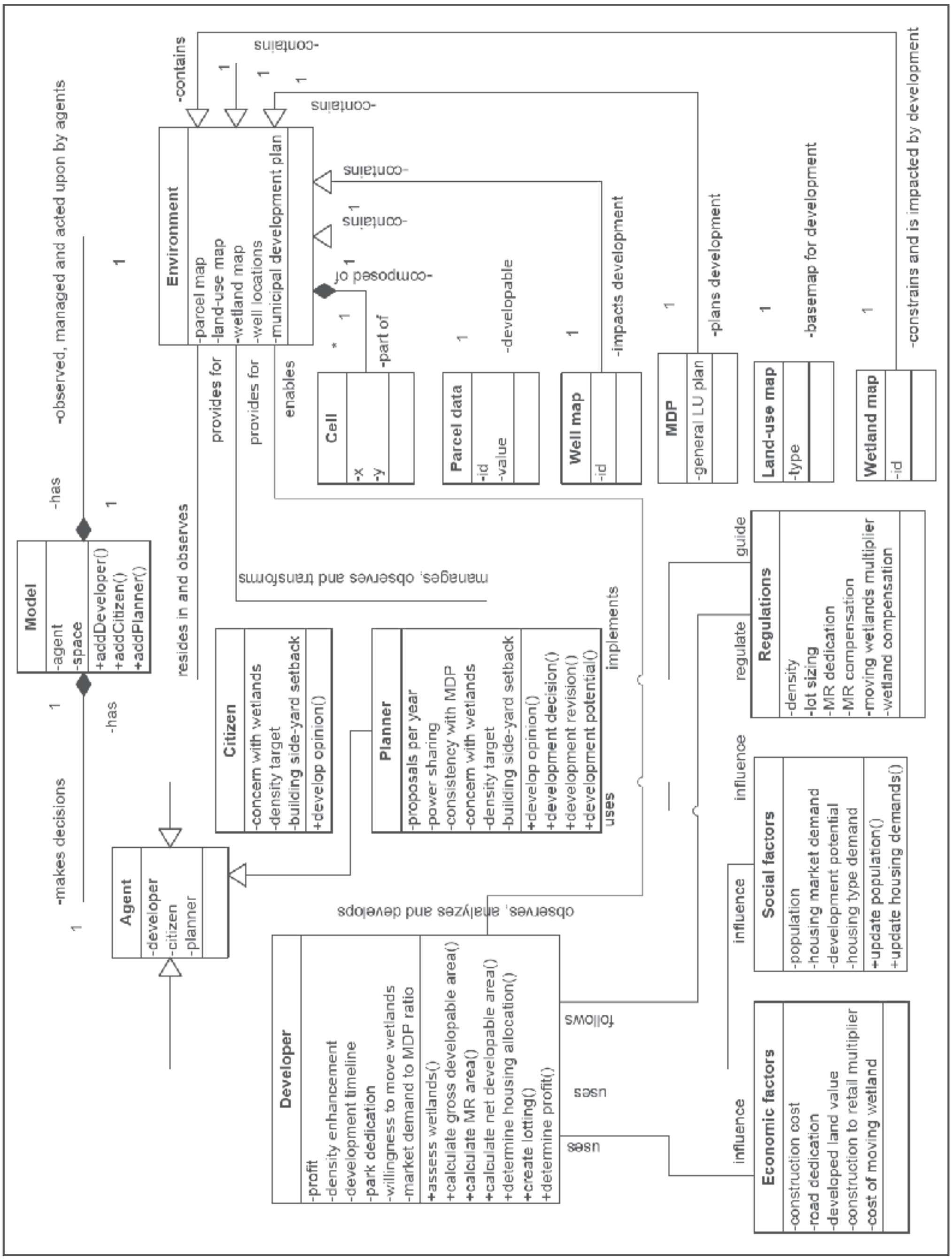


Fig. 3. The conceptual model

2.2.1 Agent-based component

The key stakeholders that were abstracted from the real-world *land development planning process* are the Developer, the Planner and the Citizens. The Developer combines those stakeholders that have financial interest in the development of the Strathbury project: the land developer who will profit from the conversion of the land into developable lots, and the engineering and planning firm who was providing consulting services to the land developer who will profit during the planning and preliminary engineering stages. The Planner combines those stakeholders that ensure the proposed land development follows governmental regulations with those in political positions that make decisions on the approval of proposed developments in the Town of Strathmore. They are the municipal planning department, the infrastructure department, the parks and recreation department, the emergency services department, and the Strathmore Town Council. The Town of Strathmore is a small municipality having only one Town Planner that advises the Town Council on a proposed development. The political arena, the Town Council, is not included in this model; therefore the assumption is made that the advice of the Town Planner on a proposed development is the decision of the Town Council. The Citizen includes those stakeholders that neighbour the proposed development, community associations, and other citizens in the Town of Strathmore. The citizens of the Town had particular environmental concerns regarding the Strathbury development and are therefore considered a key stakeholder.

Although electrical, gas, and communications utility providers do have financial interest in a proposed development, typically they are not involved in the decision making and are therefore not included in the model. Typically, if some utility infrastructure exists within the proposed development, such as high pressure gas, electrical transmission or communication towers, the proposed development may need to be planned around the existing facilities. If they can be relocated, the utility company may financially make the development more or less feasible by either absorbing the cost of utility relocation or passing the cost of relocation onto the developer.

2.2.2 Raster-based land-use change component

The key environmental state that was abstracted from the real-world *land development planning process* was the land-use designation on each land parcel. The environment was abstracted into raster-based cells of a uniform size with land-use values as attributes.

2.3 Information used in implementing the model

Five different types of information needed to represent the *land development planning process* were gathered and implemented in the model: 1) agent information, 2) social factors, 3) economic factors, 4) government regulations, and 5) spatial datasets.

2.3.1 Agent information: goals, decisions and factors influencing decisions

The first step in developing the agent-based model was to determine for each of the stakeholders, namely the Developer, the Planner and the Citizen, the specific goals they are attempting to fulfill, how they make decisions, the factors that influence their decisions, and how they communicate with each other. The document entitled "A Community Guide to the Planning Process" outlines the general Land-use Redesignation and Outline Plan process that is applied in southern Alberta and by the Town of Strathmore (City of Calgary (2002). This document guided the collection of information from the stakeholders.

For the Developer and the Planner stakeholders, information was gathered in three stages: through a questionnaire, a formal interview, and an unstructured interview. Individuals that were directly involved in the Strathbury *land development planning process* included the Planner for the Town of Strathmore, and the Strathbury land developer from WestCreek Developments. A Professional Civil Engineer from Eclipse Geomatics and Engineering, who was contracted by the developer to do the engineering for the proposed development and who is considered an expert in the field of land development, was also contacted to provide information on the status of the Strathbury application, public hearing and council decision process, as well as insight on the *land development planning process* in the Town of Strathmore, the governmental regulations affecting land development, and the land-use change allocation.

A questionnaire was prepared based on preliminary discussions with the Town Planner and the Professional Engineer. It was provided to each representative stakeholder to answer. The information collected from the questionnaire was used to guide the structured interview. During the structured interview process, it was discovered that although the questions provided a general understanding of the factors influencing the decision process of the stakeholders, the depth was insufficient. An unstructured interview ensued allowing the representatives to talk freely about the planning process, their goals, how the decision process occurs, the information they use to make their decisions and how the *land development planning process* was going with the Strathbury project.

For the Citizen stakeholder, it was concluded that questioning only one citizen would be biased and questioning many citizens was unwarranted. During the unstructured interview, the Planner provided information on feedback he received through written and verbal communication with concerned citizens. In the public consultation for the Outline Plan and Land-use Redesignation associated to the Strathbury development, some motivated citizens provided written comments on the proposed development. These comments became part of the public documentation for the development and are summarized and addressed by the developer. Although this information may not represent the opinion of all citizens, it was the best available to represent the perspective of the Citizen stakeholder and was used in the model.

2.3.1.1 Developer stakeholder: goals, decision and influence

During the interview, the developer explained the general objectives that his company attempts to meet with all proposed land developments, including: profit, density, construction cost and timeline, and lot retail value. The developer discussed the general infrastructure issues that the Town of Strathmore needs to address prior to approving any development projects including: sweet gas well buffers, water sourcing and water treatment, sanitary sewer disposal, and storm water management. He also discussed specific details that pertain to the Strathbury development of which wetlands were the most controversial (Developer 2007).

The land developer wishes to maximize the number of market demanded lots by minimizing the lot size and increasing the density, to minimize costs - hence maximize profit, to provide building lots quickly, to provide the required Municipal Reserve (park space) rather than monetary compensation to the Town, to move wetlands when they interfere with the proposed design, and to follow the market demand for housing rather than the Municipal Development Plan. From this information, six properties were abstracted to become the Developer stakeholder goals: 1) profit, 2) increase residential density, 3) development timeline, 4) park dedication, 5) willingness to move wetlands, and 6) market demand to MDP ratio.

Many regulatory factors dictate the decisions of the land developer when planning a development including: the municipal development plan, the current and adjacent land use, the municipal land-use bylaw, the environmental regulations, and the law. Economic factors also influence the decisions of the land developer such as the housing market demand, the market value of developable lots, the construction cost, the cost of developable land, the distance to existing infrastructure, and the presence of wetlands within the land parcel. When making the decisions on a proposed development, the developer looks at different development schemes, applies the regulations, assesses all the influencing factors, and then "calculates" the most suitable and profitable scheme. If the developer performs his/her "due-diligence", the proposed development plan on a parcel of land should be accepted (Developer 2007).

2.3.1.2 Citizen stakeholder: goals, decision and influence

Citizens are the source of values that define the community. They identify problems and provide feedback on solutions that are implemented. Typically the more involved citizens are in the community, the more influence they have on decisions affecting their community.

The comments provided by the citizens proved to be very useful in developing their general concerns regarding the Town's growth, the typical 'not in my back yard' (NIMBY) apprehension, the need for a park system, and the preservation of wetlands. The following quotations come from six different letters received by the Planner regarding the Strathbury project and are reflective of the comments given by the Planner during the interviews (Citizens 2006): (a) "strongly oppose the proposed amendments for the redesignation of the land directly behind our home; (b) the town "promotes green areas and Urban Reserves yet is proposing to build homes and condos on one of the most beautiful green areas remaining" in the town; (c) "the land has numerous ponds and we're very concerned about the water level specifically where the water will flow if homes are built in the area"; (d) "preserve (the Strathbury land) and further enhance it so that future generations of our residents and our wildlife will have the space to access"; (e) "we are desperately in need of areas to walk with our families"; (f) "we (want to) look well into the future and plan not only residential, business and shopping spaces, but areas (that) will enhance (the) quality of life"; and (g) "(the Strathbury land) is really an extension of the wetland across the road, and has several smaller wetland areas within it. Please preserve it with a plan for enhancement in the future".

The compiled information revealed these general desires of the citizens: they like the small town feel and they want to maintain it, they do not want the urban sprawl of Calgary, they like the network of walking trails within the town, and they feel the wetlands in their community are a great asset and want to maintain them as part of their park system. The Town planner verbally communicated a concern with the fire hazard associated with houses being excessively close. From this information, four properties were extracted to represent the Citizen goals in the model: 1) concern with wetland disturbance, 2) maintain Municipal Reserve (park space), 3) maintain density per the MDP, and 4) increase building side-yard setback.

In general the citizens' are greatly concerned by the impact on wetlands and the continuity of their park network. They evaluate the development proposal created by the developer mostly in terms of the impact on wetlands and the integration of park space and share their positive or negative opinion with the town planner.

2.3.1.3 Planner stakeholder: goals, decision and influence

During the interview, the Town planner described the Town's current zoning bylaws, the infrastructure issues, the trail network system, the municipal development plan, the density objectives, the future growth plans, and the wetland policy, which was recently updated following a public survey of the town's residents. He/she also talked about his/her role as a sounding board to residents' concerns and as an advisor to the Town Council. Details pertaining to the Strathbury development project itself were also discussed, including a goal of slightly increasing the density on account of its vicinity to the town centre, and the issue of wetlands (Planner 2007). Some frustration was expressed in having inherited an aged Municipal Development Plan and having to work with existing Planning Policies that really didn't match the sustainability, density and growth goals that were now desired. For over a year, the town planner had worked with the Town council on a new Municipal Development Plan (MDP) and almost had it adopted by the Council. At the time of the interview, the Town had just had a Municipal election that completely changed the Council members. As a consequence, the Planner would have to go through the entire process again before adopting the new MDP.

The town planner must interpret planning regulations for other municipal decision makers and be able to educate citizens about the benefit of community planning. He is the moderator between the land developer and the citizens over the wetland issues while meeting the needs of the growing community. He is also the citizen educator providing an open door for citizens wishing to discuss community planning and future plans of the Town. Opinions of citizens showing an interest and a genuine concern for the direction of the community planning within the Town of Strathmore are given more credence by the Planner. The information compiled from the interview revealed specific desires that the planner wishes to achieve for the Town: to implement a new community growth strategy (MDP) with a transit-oriented design, allowing for an increase in density in redevelopment areas near the town centre, to provide direction for the new developments within the recently annexed Town boundaries, and to solve storm and sewer infrastructure problems. From this information, seven properties were extracted to represent the planner goals in the model: 1) development approvals per year, 2) weight of citizens' opinion, 3) consistency with the town's municipal development plan, 4) concern with wetland disturbance, and the increase/maintain/decrease of 5) Municipal Reserve (park space), 6) density, and 7) building side-yard setback.

The regulatory factors that dictate the decisions of the planner when planning a development includes the municipal development plan, the current and adjacent land use, the municipal land-use bylaw, the environmental regulations, and the law. As the planning division in a city/town is the authority on the municipal land-use bylaws and the Municipal Development Plan, the planner has the ability to interpret them differently. This flexibility has been captured in the "consistency with the town's municipal development plan" property of the planner. In addition, social factors influence the decisions of the planner including: the citizen involvement, the housing demand, the urban development potential, and the population growth. A key decision that must be made by the planner is related to the sharing of decision-making power with the citizens: the greater the involvement of the citizens, the more decision-making power they are given. The planner must evaluate the development proposal created by the developer in relation to the town's goals and the existing regulations. Then, a decision is made to accommodate the opinion of the citizens,

the housing demand and the town's municipal development plan, and the right of land owner, represented by the land developer, to develop his/her property.
The planner is also responsible for updating the MDP every five to ten years, a process that involves public hearings, public consultation, and growth prediction. Since the new MDP had not yet been approved by Town council, it was not public documentation and therefore could not be supplied; however the planner provided some information on its general direction (Planner 2007).

2.3.2 Social factors influencing stakeholders' decision

Population growth and housing demand are influential in the decision making of the Planner, Developer and Citizen, making it necessary to balance the appropriate type and quantity of housing. The following is a list of the social factors impacting decisions abstracted for the model and how they were quantified as parameters in the model. The first factor is the population growth. The population of the Town of Strathmore in 2007 was 10728 persons (Planner 2007); the estimated population growth was calculated on a yearly basis from a population projection report (CH2M HILL 2007). The second factor is the housing market demand. The initial approximate value for the Town is 50 units (130 persons) in 2007 (Planner 2007). The future demand was calculated in the model on a yearly basis as the population growth, less the number of new homes created that year, multiplied by the average household size (2.6 persons/household) (Statistics Canada 2007). The third factor is the development potential. It is a value estimated by the Planner that is based on the residential construction in approved residential land developments in the Town. Based on the approved developments, an estimated 125 units (325 persons) could be built in 2007/2008 (Planner 2007). The fourth factor is the housing type market demand: initial values (50% R1 (residential one), 15% R2 (residential two) with a 25′ wide lot, 10% R2 with a 30′ wide lot, 15% R2Xatt (residential two-X attached), and 10% R2Xdup (residential two-X duplex) for each housing type for the Strathbury development were provided by the Developer during the interview (Developer 2007). Since the market study only covered the Strathbury development, these numbers are assumed constant.

2.3.3 Economic factors influencing stakeholders' decision

The land cost, construction costs, and market value for developable lots are influential in the decisions made by the Developer. The following is a list of the abstracted economic factors and how they were quantified as parameters in the model. The first economic factor is the land value. Assessed land values ($/hectare) were obtained for each developable cadastral parcel from the Assessment Offices of Wheatland County (2007) and the Town of Strathmore (2008). Knowing that the market land value for the Strathbury development land is approximately 2.5 times the assessed value, an assumption was made that all market values were 2.5 times the assessed land value. The second factor is the construction cost per metre of lot frontage. From experience, the developer has determined an approximate cost per linear metre of lot frontage that includes all the construction costs of $3000/foot of lot frontage ($10000/m of lot frontage) (Developer 2007). The third factor is the percentage of road dedication. From experience, the developer has determined an average percentage of the developable area that is dedicated to roads of 29%, of which 34% are 22 m wide collector streets and 66% are 15 m wide local streets (Developer 2007). The fourth factor is the developed land value. From experience, the developer knows the approximate retail value for the different housing types (residential-1 lot (R1): ~$450000, residential-2 25′ wide lot (R2-25): ~$250000, residential-2 30′ wide lot (R2-30): ~$300000, residential-2X duplex lot (R2Xdup): ~$225000, residential-2X attached lot (R2Xatt): ~$200000) (Developer 2007). The fifth factor is the construction to retail value multiplier. From experience, the developer has devised a multiplier to use when determining the development feasibility and profit. The minimum retail value should be three times the construction cost. If the construction costs are lower or the retail value is higher than a profit is made. Conversely, if the construction costs are higher or the retail value is lower than a loss is declared (Developer 2007). The last factor considered is the cost of moving wetlands. According to the Alberta Government (2000), the cost of constructing wetlands varies between $12,000 and $60,000 per hectare. Included in these costs are the land, design, earth moving, planting, monitoring and maintenance.

2.3.4 Governmental regulations

Municipal governmental regulations, implemented by the Planner, must be followed by the Developer when proposing a development. The following is of list of the abstracted land-use regulations and how they were quantified as parameters in the model. The first regulation concerns density. The average density as outlined in the Land-use Bylaw and MDP is 15 units/hectare (6 units/acre) (Town of Strathmore 1998). The second regulation is the lot sizing. The minimum lot width and lot area values were obtained from the Strathmore Land-use Bylaw (Town of Strathmore 1989). The third and fourth regulations are the minimum lot area: R1=464 m^2, R2-25=255 m^2, R2-30=302 m^2, R2Xdup = 232 m^2, R2Xatt = 185 m^2, and the minimum lot width: R1=15 m, R2-25=7.6 m, R2-30=9 m, R2Xdup = 7.5 m, R2Xatt = 6 m (Town of Strathmore 1989; Town of Strathmore 1998). Finally, the percent MR dedication or MR compensation was considered. The Alberta Municipal Government Act requires that proposed subdivisions provide part of the land as municipal reserve (MR) not to exceed 10% or monetary compensation (Government of Alberta 2000).

Two environmental regulations were abstracted and quantified as parameters in the model. The first one is moving wetland multiplier. In some situations, developers may consolidate existing wetlands within a proposed development. Relocating wetlands can be damaging and they are typically less productive. Wetland of 100 to 300% of the original wetland size may be required. The second environmental regulations concern the wetland compensation. In situations where wetlands are destroyed, a monetary compensation of ~$18,000/hectare must be paid to an environmental protection organization.

2.3.5 Spatial datasets

A GIS (Geographic information system) database was developed using the ArcGIS software from Environmental Systems Research Institute (ESRI) to integrate the relevant information about the study area. Among the spatial datasets used in the model are the cadastral parcels: AltaLIS (2007) cadastral parcel data were obtained from the University of Calgary Maps, Academic Data, Geographic Information Centre (MADGIC) in vector format. The assumption was made that large undeveloped or Greenfield land parcels, closer to existing infrastructure, and closer to the center of the Town of Strathmore will be developed first. Therefore, urban reserve (UR) and agricultural (AG) parcels were weighted according to their size, distance to existing roads, and distance to the downtown core of Strathmore. Parcels were sorted by their weight, highest to lowest, and given a unique parcel identifier (Parcel No. 1, 2, 3…); the Strathbury project was Parcel No. 2 in the sequencing.

A land-use map of the study area was produced. AltaLIS (2007) vector cadastral parcel data were attributed land-use values from the Town of Strathmore (2007): 1, 2, 3, 4 = various types of commercial, 5 = public service, 6, 7 = various types of industrial, 8, 9, 10, 11, 12, 13 = various types of residential, 14 = municipal reserve, 15 = environmental reserve, 16 = open space, 90 = agriculture, 91 = urban reserve, 92 = Western Irrigation District (WID) canal, 99 = road.

Another source of data is the Municipal Development Plan. The quantifying of the MDP required the rework of the land-use map in the existing MDP (Town of Strathmore 1998) based on the information provided by the planner during the interview. In general the MDP requires 45% residential one (R1), 25% residential two (R2) and 30% residential two-X (R2X). Each residential parcel was given land-use percentages based on the information provided and stored as: [Parcel No., 45, 25, 30]. The Strathmore Lakes Estates Area Structure Plan further regulates the percentages of the Strathbury lands to 30% R1, 45% R2 and 25% R2X; therefore its values are stored as: [2, 30, 45, 25].

Finally, the Strathmore Wetland Inventory map from the Town of Strathmore (2007) was digitized into vector format. Gas well locations were provided by the Alberta Energy Resources Conservation Board (ERCB) (2007) in vector format.

2.4 Implementation

This section presents how the five types of information described in the previous section were implemented within the model.

2.4.1 Agent implementation: properties and decision functions

The goals of each stakeholder type and the factors influencing their decision were abstracted to become the properties of the agents, and the decision making was abstracted into decision functions with property variables. The properties and results of decisions from each agent type were quantified as numerical values, stored as arrays of numbers, or tuples.

In the implementation, each agent type was given an *opinion* property that ranges from -1 to 1 (negative opinion to positive opinion) and a *happiness* property that ranges from 0 to 10 (unhappy to happy). At different steps throughout the model, an agent evaluates the results of a decision and develops an *opinion*. A comparison is done between values contained in the decision tuple and values contained in each of the agent's properties tuple. If the result of a decision is contrary to an agent particular property, it will have a negative (-1) impact on his opinion regarding that property; if the result of a decision is similar to an agent property it will have a positive (+1) impact regarding that property. The average opinion is calculated and is weighted by 10 less the happiness and is stored as the agent *opinion* property; therefore the *opinion* of an unhappy agent will be stronger. The *happiness* property of an agent fluctuates according to how his *opinion* is accepted. If his *opinion* is ignored in the following development decision, it will lower his *happiness* and if it is well received, it will increase his *happiness*. Fig. 4 provides an example of the calculation.

Provision was also made for weighing each agent property allowing for different properties to be given more or less importance when developing an *opinion*. This was implemented in a Multicriteria Decision Analysis fashion using an Analytic Hierarchy Process (AHP) method called the pairwise comparison (Malczewski 1999). Each pair of criteria, or properties, is evaluated separately; one property is given an intensity of importance value over another property. The values range from 1 to 9 (equal importance to extreme importance) and they are entered into a matrix form. The values in pairwise comparison matrix are then checked for consistency by normalizing the eigenvector by the eigenvalue of the reciprocal matrix. If the consistency ratio is less than a certain value, then the values are said to be consistent; if the consistency ratio is greater than the value, the importance values are not consistent and they must be re-evaluated. A weight for each property is also derived, the sum of which equals 1. The weights are then normalized with the smallest weight being equal to 1. The normalized weight is applied to each opinion (+1/0/-1) before the agent *opinion* is developed, as previously discussed.

AGENT HAPPINESS:
- Citizen: 8
- Developer: 7
- Planner: 8

AGENT PROPERTIES:
- Citizen: maintain density of 6.0 and do not disturb wetlands
- Developer: increase density and willing to move wetlands
- Planner: increase density and do not disturb wetlands

PROPOSED DEVELOPMENT:
- Developer: density of 6.5 and wetlands are disturbed

AGENT OPINIONS:
- Citizen: -1 on density, -1 on wetlands, average = -1, weighted = -2
- Developer: +1 on density, +1 on wetlands, average = +1, weighted = +3
- Planner: +1 on density, -1 on wetlands, average = 0, weighted = 0

DEVELOPMENT DECISION:
- Planner: approve

AGENT HAPPINESS:
- Citizen: 6
- Developer: 10
- Planner: 8

Fig. 4. Example of agent opinion and happiness calculation

The benefits of using this method over a straight rank weighting are twofold. First, the resultant weights are not only relative to one another, but they also have absolute values; second, the user only compares two goals at a time rather than subjectively weighting all goals at the same time (Malczewski 1999). The pairwise comparison method was used by Malczewski et al. (1997) in a multicriteria group decision-making model to analyze environmental conflict. In the model, stakeholders in planning or resource management positions evaluate the suitability of land for different socio-economic activities. The research of Malczewski et al. (1997) showed that the pairwise comparison method allowed the stakeholders to objectively derive weights for the various land uses, rather than subjectively assigning them.

2.4.1.1 Developer agent properties and decision functions

The properties of the Developer agent are stored in the *developer tuple* and were implemented as follows: (1) *Profit*: the goal on the return on the capital investment put into the land parcel: 5 to 20%; (2) *Density enhancement*: the goal to increase the allowable density: 0 to 2 units/acre; (3) Development timeline: the goal on the start and expected completion of construction. Also used to derive the *development potential* per year, (a) *Start construction*: 1 to 5 years, (b) *Finish construction*: 2 to 10 years; (4) *Park dedication*: The Developer's goal regarding the creation or monetary compensation of Municipal Reserve: create MR = 1, provide compensation = 2; (5) *Willingness to move wetlands*: The Developer's view on the displacement of wetlands to accommodate the proposed development: Move = 1, Don't move = 0, (5.1) *Size of wetland moved*: The maximum size of a wetland the Developer is willing to displace: 1000 m^2 to 40000 m^2; and (6) *Market demand to MDP ratio*: The Developer's stance when weighing the housing market demand versus the MDP: 0.1/1 to 4/1.

Typically an application for a development submitted to the town planner contains a report, several plans and other required independent studies. In the model, these documents have been abstracted as a sequence of numbers that translate the content of those documents submitted as a proposed development into a *development tuple*. The *development tuple* contains the results of the above Developer decision functions, which will be discussed next. The values contained in the *development tuple* include: the proposed cadastral unique parcel identifier, the proposed density, the timeline for the land development project, the percentage of each land-use type, the residential lot dimensions and number of lots of each residential type, and the wetlands proposed to be displaced.

The following is a description of how the eight decision functions of the Developer were implemented: (1) *Wetland assessment function*: The Developer calculates the size of wetlands impacting the development and determines if any wetlands are below their maximum *Size of wetland moved property (5.a.)*. If wetlands are to be moved, their total area is multiplied by the *Moving wetlands multiplier* and the area is added to the existing wetland area; (2) *Gross developable area function*: The Developer determines the amount of developable area, which is the gross area less the Environmental Reserve (ER) or wetlands from the *wetland map*; (3) *Municipal Reserve (MR) function*: The Developer determines the amount of land to be dedicated as MR (park) from the developable area based on the *Percent MR dedication* parameter; (4) *Net developable area function*: The Developer determines the area of developable area that will be residential and that will be road based on the *Percentage of road dedication* parameter; (5) *Housing allocation function*: The Developer uses the *Market demand to MDP ratio* property and weights the market demand for residential lot types: R1-detached, and R2(X)-semi-detached(attached), with the allocation in the *Municipal Development Plan (MDP)*; (6) *Lotting function*: The Developer determines the number of lots based on the *Minimum lot area* and *Minimum lot width* for each residential lot type, the *Density* regulation parameters, and its *Density enhancement* property; the lot depth is optimized to use all the developable area; (7) *Profit determination function*: The Developer determines the profit in the proposed development as the market value for sold lots, comparing the *Developed land value* to the construction cost, which is a function of the *land value*, *Construction cost per metre of frontage*, the amount of lot frontage, and the *Construction to retail value multiplier*; and (8) *Opinion function*: The Developer compares, as discussed in 2.4.1, the appropriate values in the *development tuple* with the first five values in the *developer tuple*. Although the Developer follows its properties when initially proposing a development, due to the social and economic influences and governmental regulations the resulting proposed development may not meet its goals, impacting the *Developer opinion*. The Planner may also ask to revise particular aspects of the development, discussed in section 2.4.1.3, that do not meet the goals of the Developer.

2.4.1.2 Citizen agent properties and decision functions

The properties of the Citizen agent are stored in the *citizen tuple* and were implemented as follows: (1) *Concern with wetland disturbance*: The Citizen's view on the displacement of wetlands to accommodate the proposed development: Concerned = 1, Not concerned = 0; (1.1) *Size of wetland moved*: The maximum size of a wetland the Citizen is willing to see moved: 1000 m^2 to 40000 m^2; (2) *Density target*: The Citizen's goal regarding the density in proposed developments: increase the density = +1, maintain the current level in the bylaws = 0, decrease the density = -1; and (3) *Building side-yard setback*: The Citizen's goal regarding the distance between residential buildings as a fire protection measure: increase the current building setback = +1, maintain the current building setback = 0, decrease the current building setback = -1. The *opinion function* of the Citizen was implemented through a comparison of the appropriate values in the *development tuple* with the values in the *citizen tuple*.

2.4.1.3 Planner agent properties and decision functions

The properties of the Planner agent are stored in the *planner tuple* and were implemented as follows: (1) *Consistency with MDP*: the Planner's goal on how consistent the proposed developments must be with the town's Municipal Development Plan: no varying from the MDP = 0%, to quite flexible = 20%; (2) *Concern with wetland disturbance*: the Planner's view on the displacement of wetlands to accommodate the proposed development: Concerned = 1, Not concerned = 0; (2.1) *Size of wetland moved*: The maximum size of a wetland the Planner is willing to see moved: 1000 m^2 to 40000 m^2; (3) *Density target*: the Planner's goal regarding the density in proposed developments: increase the density = +1, maintain the current level in the bylaws = 0, decrease the density = -1; (4) *Building side-yard setback*: the Planner's goal regarding the distance between the residential buildings as a fire protection measure: increase the current building setback = +1, maintain the current building setback = 0, decrease the current building setback = -1; (5) *Power sharing*: the Planner's view on the weight given to the opinion of the Citizen: 0.1 to 4; and (6) *Proposals per year*: the Planner's goal for the number of proposals to review per year: 1 to 10; this goal can also vary based on the housing demand.

The following is a description of how the decision functions of the Planner were implemented: (1) *Opinion function*: the Planner compares, as discussed in section 2.4.1, the appropriate values in the *development tuple* with the land-use allocation in *the Municipal Development Plan (MDP)*, the *housing demand* and the d*evelopment potential* of the town, the land-use bylaws (*density*, *minimum lot area*, and *minimum lot width*), and the first four values in the *planner tuple*; (2) *Decision function*: The Planner weights the *Citizen opinion* based on the *power sharing* property. The sum of the *opinions* of the agents is calculated. If the sum is positive the decision is an approval; if the sum is negative, the Planner requests revisions. A decision of rejection occurs after four revisions; (3) *Revision function*: A request for revisions includes simple recommendations to the Developer regarding the proposed development. These recommendations are based on the *opinions* of the Citizen and the Planner: "(Increase/Decrease) density", "(Increase/Decrease) lot width", "(Increase/Decrease) development time", "(Increase/Decrease) MR dedication", "Follow MDP more closely"; and (4) *Development potential function*: The Planner evaluates the *development potential* on a yearly basis based on the *development potential* of the previous year, less the *housing market demand* for that year, plus the *development potential* of approved residential land development projects whose construction timeline contributes to the *development potential* for that year. As an example, if in the current year a development containing 200 units is approved having a *start* and *finish construction* timeline of one and five years respectively, the development will contribute 50 units per year to the *development potential* for the following four years.

2.4.2 Social and economic factors and governmental regulations implementation

The social factors, the economic factors, and the governmental regulations were abstracted into model parameters. The following describes how each of the parameters is stored: (1) *Population growth*: variable calculated yearly; (2) *Housing market demand*: variable calculated yearly; (3) *Development potential*: variable calculated yearly; (4) *Housing type market demand*: stored as a constant for each residential housing type; (5) *Land value*: stored as a two dimensional array with the unique parcel identifier; (6) *Construction cost per metre of frontage*: stored as a constant; (7) *Percentage of road dedication*: stored as a constant; (8) *Developed land value*: stored as a constant for each residential housing type; (9) *Construction to retail value multiplier*: stored as a constant; (10) *Cost of moving wetlands*: stored as a constant; (11) *Density*: stored as a constant; (12) *Minimum lot area*: stored as a constant; (13) *Minimum lot width*: stored as a constant; (14) *Percent MR dedication* and *MR compensation*: stored as constants; (15) *Moving wetlands multiplier*: stored as a constant; and (16) *Wetland compensation*: stored as a constant.

2.4.3 Agent-agent interaction

Agent-agent communication mimics the steps 2, 4 and 9 of the *land development planning process* (Fig. 3): (1) Developer - Planner: a development proposed by the Developer as a *development tuple* is submitted to the Planner and is circulated to the Citizen; (2) Citizen - Planner: the Citizen shares its *opinion* regarding the proposed development with the Planner; and (3) Planner - Citizen and Planner – Developer: the decision of the Planner on a proposed development is shared with the Citizen and the Developer. A request for revisions includes the recommendations from the Planner *revision function*.

2.4.4 Agent-environment interaction

Agent-environment interaction occurs on several occasions within the model as environment observations and environment transformations: (1) Observation by the Developer: the Developer observes the wetlands within the land parcel of the proposed development and evaluates them through the Developers *wetland assessment function*; (2) Observation by the Citizen: the Citizen observes the wetlands within the proposed development and generates an *opinion* based on how they are impacted; (3) Observation by the Planner: the Planner observes the wetlands within the proposed development and generates an *opinion* based on how they are impacted; (4) Transformation by the Planner: a decision by the Planner to approve a proposed development generates an immediate transformation of the *land-use map*. The transformation does not physically change the environment but it allows the Developer to begin construction.

Greenfield land-use change principles for residential developments, typically followed by developers, were developed for the environmental transformation of the *land-use map* with the assistance of an urban planner and the Civil Engineer. These principles are based on the existing land use, urban reserve (UR) or agricultural (AG), and its change to residential (R1, R2, R2X), municipal reserve (MR), environmental reserve (ER), or open space (OS) based on the land use of parcels adjacent to the proposed land development parcel, and desirability. The land-use transformation for the parcel of land is based on the percentage of each land-use type values contained in the *development tuple* of the approved development, as discussed in 2.4.1.1. Examples of these principles, hard coded as land-use change rules in the model, include: wetland areas are surrounded by a linear park (MR) buffer; low density residential (R1) housing is placed adjacent to the more desirable open space (OS) and park (MR) and are minimized adjacent to the less desirable main thorough fares, commercial, industrial, and higher density residential land-use areas; medium density residential (R2 & R2X) housing is placed adjacent to the less desirable main thorough fares, commercial, industrial, and higher density residential land-use areas; a sizeable area of MR is usually set aside either for recreational fields or a future public facility (school, church, or community center). The transformation of the *land-use map* then affects the future land use of adjacent proposed developments.

The following eight steps describe how the transformation of the *land-use map* occurs within the model using the values contained in the *development tuple*: (1) The wetlands proposed to be moved are amalgamated with the largest existing *wetland* within the proposed development parcel; (2) *Wetland map* areas within the proposed development become Environmental Reserve (ER) cells and are given a 4 m ER buffer of cells; (3) *Gas well locations* are given a 50 m buffer around well sites that must be maintained as Open Space (OS) cells which can be used for recreation purposes; (4) The remaining cells are divided based on the percentage of each land-use type values contained in the *development tuple*; (5) 8 m wide linear parks (MR cells) are generated adjacent to ER, and the WID (Western Irrigation District) canal cells; (6) Medium density residential (R2 cells) is placed adjacent to existing main thorough fares, commercial, industrial, and higher density residential; (7) Low density residential (R1 cells) is placed adjacent to parks and open space and if necessary higher density residential; and (8) The remaining land becomes a sizeable area of MR cells.

2.4.5 Environment implementation

The environment that agents view and act upon was implemented as a raster-based landscape with a 16 m^2 (4 m x 4 m) cell size. The 4 m x 4 m resolution was chosen to accommodate the narrowest strip of land use. The different spatial data types were converted into a series of coincident raster-based maps using the ESRI ArcToolbox Conversion Tools Polygon to Raster and Raster to ASCII. The *cadastral parcel* data were converted into raster format with cell values of the unique parcel identifier. The *land-use map* was converted into raster format with cell values of the land-use type. The *wetland map* was converted into raster format with cell values of the unique wetland identifier. The *gas well locations* were converted into raster format. The *Municipal Development Plan (MDP)* data were stored in a text document.

2.4.6 Spatial and temporal boundaries

The spatial extent of the environment over which the agents make decisions includes all of the newly annexed lands of the Town of Strathmore and 1.6 km of the surrounding Wheatland County. The Town's Municipal Development Plan attempts to plan for a 30 year future growth, but is typically revised every five to ten years depending on growth rate. The ten year temporal boundary chosen for this study lies within the future plans of the Town's Municipal Development Plan, but only slightly exceeds the MDP revision period so as to give a reasonable but not extreme possible prediction of land-use change. A one year incremental time step was also implemented to update the population growth, housing demand, and housing potential; however the number of developments approved within the one year increment can be varied by the planner agent.

2.5 The computer model

Many existing programming environments exist for simulating agent-based systems including Swarm (Minar et al. 1996; Swarm 2010), Repast (Crooks 2006; Repast 2010), Mason (Mason 2010), and NetLogo (NetLogo 2010). However, to meet the particular needs of this case study, Java was used to develop the model for its familiarity and its object-oriented features.

2.5.1 User interface

An interface was developed to allow a user to view and modify the initial default property values. The default values in the interface are those developed from the information collected for the Strathbury land development project. The model interface contains three panels: 1) initial conditions, 2) run-time variables, and 3) land-use map.

2.5.1.1 Initial conditions panel

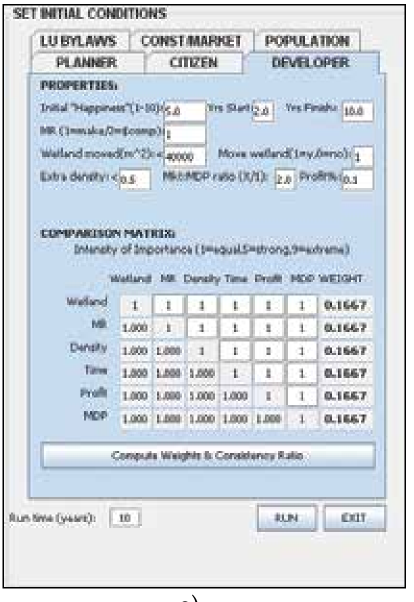
The Initial Conditions panel allows the user to modify the Developer, Citizen and Planner properties, the social and economic factors, and the governmental regulations. The matrix at the bottom of the interface for each agent is the pairwise comparison matrix that allows for the weighting of each agent property, as discussed in 2.4.1.

The following explains the interface that allows the user to set the nine Developer properties (Fig. 5a): (1) Initial "Happiness" (1-10): to set the initial *happiness* property of the Developer at the start of the model simulation; (2) Yrs Start: to set the number of years to *start construction* property; (3) Yrs Finish: to set the number of years to *finish construction* property; (4) MR (1=make/2=$comp): to set the *park dedication* property; (5) Wetland moved (m^2): to set the s*ize of wetland moved* property; (6) Move wetland (1=y, 0=no): to set the *willingness to move wetlands* property; (7) Extra density <: to set the d*ensity enhancement* property; (8) Mkt:MDP ratio (X/1): to set the *market demand to MDP ratio* property; and (9) Profit %: to set the *profit* property.

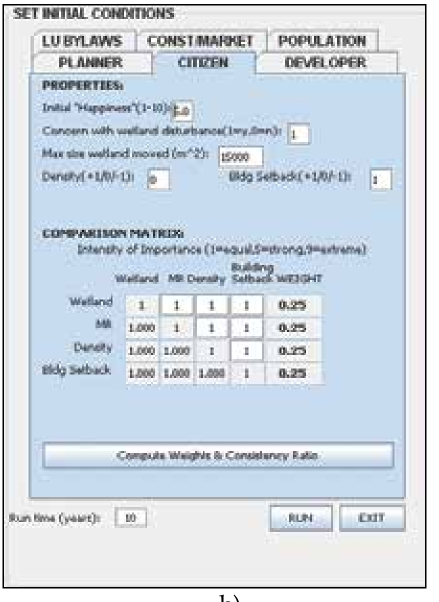
The interface for the Citizen properties (Fig. 5b) is similar to the interface for the Developer. It includes two unique properties of the Citizen: (1) Density (+1/0/-1): to set the density target property; and (2) Bldg Setback (+1/0/-1): to set the building side-yard setback property. The interface for the Planner properties (Fig. 5c) includes three unique properties: (1) Weight citizen opinion(X/1): to set the *power sharing* property; (2) Proposals/Year: to set the p*roposals per year* property; and (3) +/-% MDP: to set the *consistency with MDP* property.

The interface also allows the modeller to set the three social factors (Fig. 6a), the 18 economic factors (Fig. 6b) and the 14 governmental regulations (Fig. 6c). The social factors represented are (1) Base housing demand: to set the initial *housing market demand*; (2) Base Development potential: to set the initial d*evelopment potential*; and (3) Persons/household: to set the average household size.

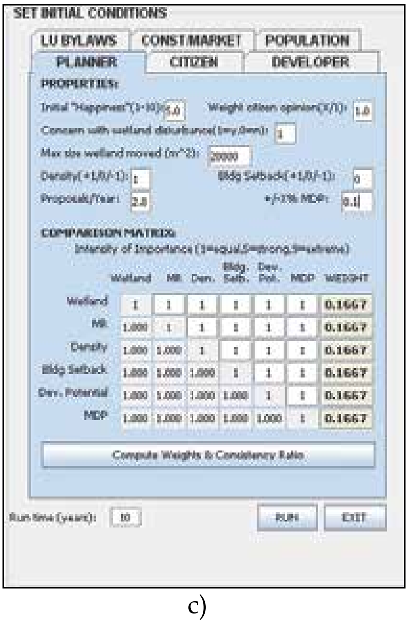
The economic factors include: (1) Const.&Marketing cost ($/m lot frontage): to set the *Construction cost per metre of frontage*; (2) Cost:Market ratio: to set the c*onstruction to retail*



a)



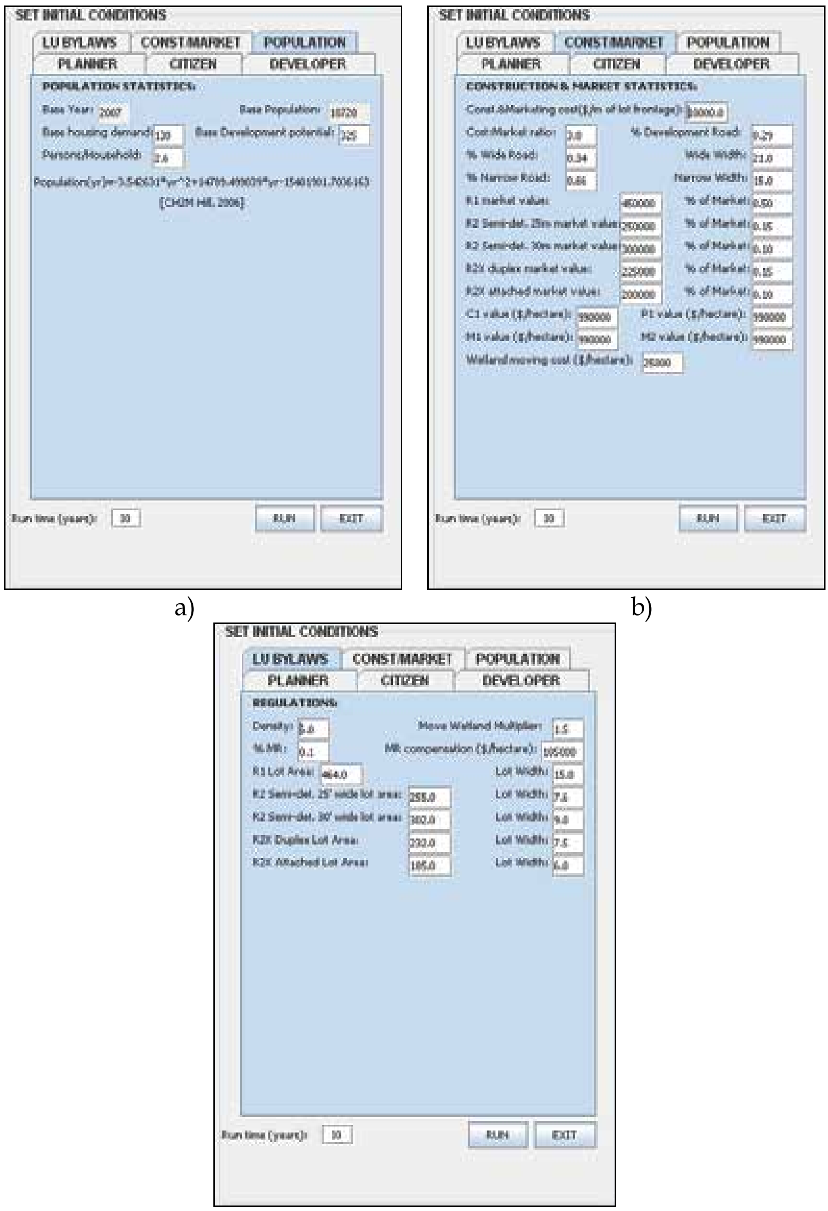
b)



c)

Fig. 5. Interface displaying a) the Developer, b) the Citizen and c) the Planner properties

value multiplier; (3) % Development road: to set the p*ercentage of road dedication*; (4 & 5) % Wide Road and Wide width: to set the percentage of collector roads and their width; (6 & 7) % Narrow Road and Narrow width: to set the percentage of residential roads and their width; (8-12) R1/R2/R2X market value: to set the *developed land value* for each housing type. The C1, P1, M1 & M2 market values were not used in the model; (13-17) R1/R2/R2X % of market: to set the h*ousing type market demand* for each housing type; and (18) Wetland moving cost ($/hectare): to set the w*etland compensation*.



c)

Fig. 6. Interface displaying a) the social factors, b) economic factors and c) the governmental regulations

The governmental regulations are (1) Density: to set the d*ensity* regulation; (2) Move wetland multiplier: to set the m*oving wetlands multiplier* regulation; (3) % MR: to set the *percent MR dedication* regulation; (4) MR compensation ($/hectare): to set the *MR compensation* regulation; (5-9) R1/R2/R2X lot area: to set the *minimum lot area* regulation for each housing type; and (10-14) R1/R2/R2X lot width: to set the *minimum lot width*.

2.5.1.2 Run-time variables panel

The run-time variables panel displays the decisions of agents, including the proposed development that has values contained in the *development tuple* discussed in section 2.5.1.1, the Citizens and Planners *opinions*, the *happiness* of each agent and the Planner's decision or recommendation (Fig. 7). The disabled check boxes in the run-time variables window are evidence of an attempt was made to implement agent behavioural change in the model and will be further discussed in the conclusion.

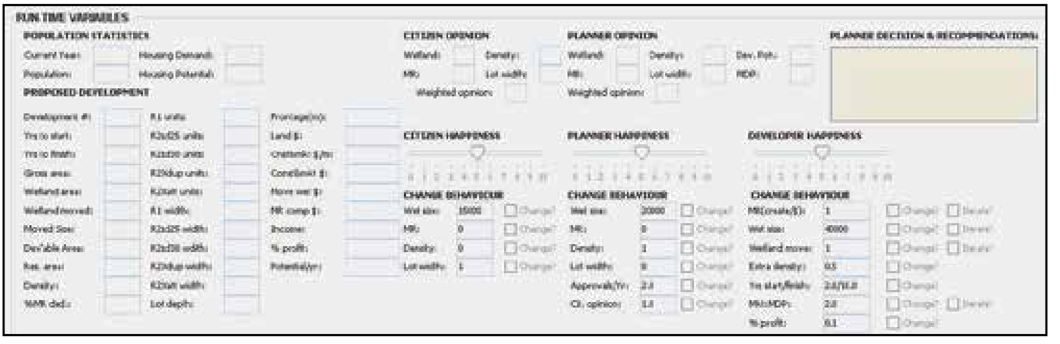


Fig. 7. Interface displaying the run-time panel

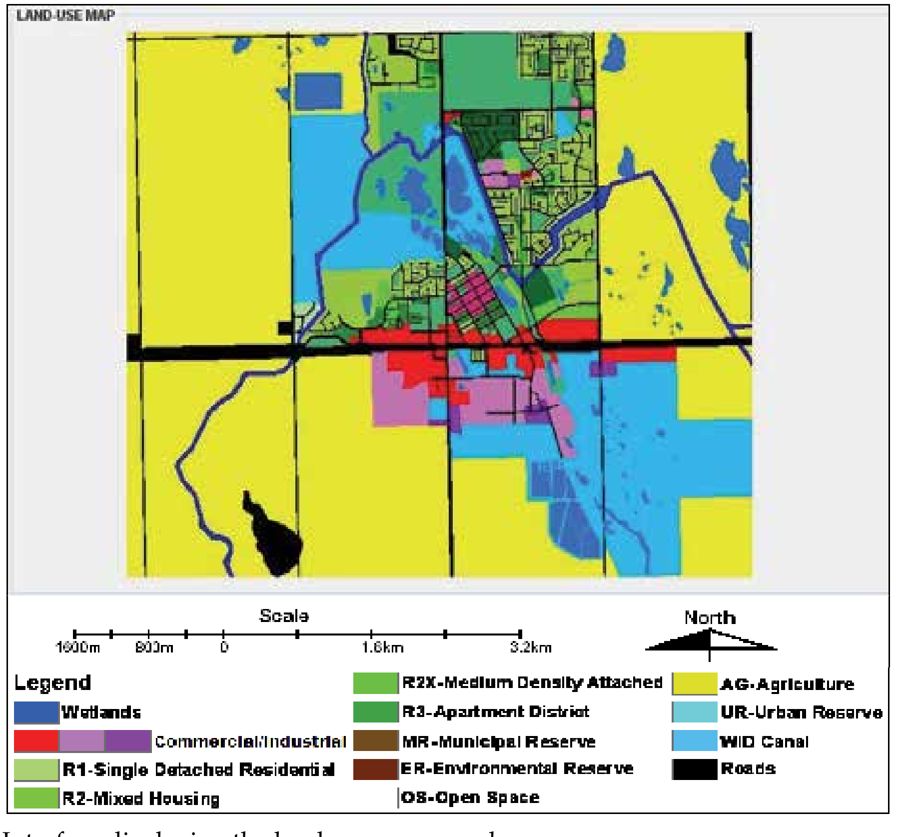


Fig. 8. Interface displaying the land-use map panel

In addition, the land-use map panel displays the changing land use as the development proposals are approved (Fig. 8).

2.5.2 The computational logic

The following ten computational steps were performed when simulating the *land development planning process* for a residential land development. 1) The software reads in the raster *land-use map*, the *wetland map*, and the *gas well locations*, the *cadastral parcel* sequencing, the *Municipal Development Plan (MDP)* information, and the *land value* data. 2) The parameters of the model initialization are populated with the default values, discussed in section 2.6, including: the agent properties, the social factors, the economic factors, and the governmental regulations. 3) The user is given the chance to change the model parameters. 4) The simulation year starts at 2007. 5) The next cadastral parcel in the sequencing, as discussed in 2.3.5, is observed by the Developer. The Developer performs the following functions: *wetland assessment*, *gross developable area*, *municipal reserve*, *net developable area*, *housing allocation*, *lotting*, and *profit determination*. A *development tuple* is produced and the Developer generates an *opinion*. 6) The Citizen performs its *opinion function* and shares its *opinion* with the Planner, as discussed in section 2.4.1.2. 7) The Planner performs its *opinion* and *decision function* and if required a *revision function*, as discussed in section 2.4.1.3. Based on the decision, the happiness of the Developer, Citizen and Planner are updated. 8) If the Planner decision is an approval, an environmental transformation to the *land-use map* occurs, as discussed in section 2.4.5. 9) If the Planner has not met the *proposals per year* approved property, steps 5 through 9 are repeated. 10) If the Planner has met the *proposals per year* approved property, the Planner's *development potential function* is performed. Finally, assuming that the maximum simulation time of 10 years has not been reached, the year is iterated and steps 5 through 10 are repeated.

2.6 Model calibration and verification

The model was calibrated using the agent goal parameter values, spatial datasets, social factor values, economic factor values, and governmental regulation values from the data that were collected from the stakeholders in the Strathbury land development project as model initialization parameter values. Verification of the computational logic of the model was done by walking through each calculation of the agent-based and raster-based land-use components and step by step comparing the calculated values to those calculated independently using a spreadsheet.

The *developer goal parameters* for the Strathbury development were initialized as follows: (1) has a construction timeline to begin construction in 2 years and finish in 10 years; (2) prefers to provide space for parks rather than having monetary compensation; (3) willing to move wetlands up to 4 hectares in size to suit the development; (4) tries to increase density by 0.5 units per acre; (5) housing market demand to Municipal Development Plan ratio is 2:1; and (6) wants a 10% rate of return on the investment in the project.

The *planner goal parameters* for the Strathbury development were initialized as follows: (1) gives equal weight to the opinion of the citizens regarding the proposed development during the decision-making process as a power sharing contribution; (2) is concerned with the movement of significant wetlands and would not like wetlands larger than 2 hectares to be displaced; (3) would like to increase the current residential density; (4) would like to maintain the current building setback per the zoning bylaws; (5) is quite concerned with the development potential; therefore it is given a greater weight; (6) in the past, has approved an average of two large development proposals per year which has kept up with housing demand; and (7) the consistency with the MDP is only +/-5% since the MDP values have been revised to match the more recently adopted Strathmore Lakes Estates Area Structure Plan (ASP).

The *citizen goal parameters* for the Strathbury development were initialized as follows: (1) is concerned with the displacement of significant wetlands and would not like wetlands larger than 1.5 hectares to be moved; (2) would like to maintain the current density goal per acre per the zoning bylaws; and (3) would like to increase the current building setback per the zoning bylaws, because of a fire hazard concern.

The *social factors*, *economic factors*, and *governmental regulations* for the Strathbury development were initialized from the values discussed in section 2.3.

The agent-based component was calibrated by comparing the values of the results in the model run-time panel (Fig. 8) to the values in the development summary table of the Strathbury Outline Plan and Land-use Redesignation application. The raster-based land-use change component was calibrated by comparing the model land-use map results to the actual land-use maps generated for the Strathbury Outline Plan and Land-use Redesignation application. Comparisons were also made between the model run-time variables and two pre-application development Strathbury concepts, produced by the developer. The development concepts proposed slightly different densities, wetland displacement options and percentages of each housing type. The calibration results are presented in section 3.1.

Although the pairwise comparison matrix that enables weighting of agent properties was implemented in the model, as described in 2.4.1, in this application, all agent properties were given an equal weight.

2.7 Development scenarios

The following five scenarios were run with the model over a period of ten years, each having different initial conditions. The first scenario called "business as usual" (BAU) assumes that the regulations, goals of the stakeholders, and decisions made in the Strathbury project are typical and that decisions will continue to be made in this manner. The second scenario called "reduction in development approvals per year" only permits one development approval per year controlling the development potential and the rate of growth. The "increase in density" scenario modifies the Land-use Bylaws and Municipal Development Plan allowing the developer to propose a higher housing density. The "change in market housing demand" scenario accounts for a prediction by Ewing (2007) that the demand for residential housing types is going to change over with the retirement of baby boomers, through changing the Land-use Bylaws, Municipal Development Plan and the market demand for smaller housing types. Finally the "sustainable development" scenario controls the development rate, assumes an increased demand for smaller housing types, decreases the areas of road infrastructure scarring the environment, and does not allow the disturbance of wetlands.

2.8 Validation of the results

The results of the model were validated using a method called "face validation" (Ligtenberg et al. 2001). This method asks persons, such as professional planners, who are considered to be experts in the subject matter to compare the simulation results to their knowledge of the real-world system and make judgements on the results.

3. Result analysis

This chapter is divided into three sections. The first one discusses the results of the calibration and verification of the model as it simulates the proposed Strathbury development. The second section presents the result of each of the six development scenarios that were simulated. The third section addresses the validation of the model results.

3.1 Calibration results: the Strathbury development

As mentioned in section 2.6, the model results from the Strathbury proposed development values were compared to values contained in the development summary table within the actual Outline Plan and Land-use Redesignation application. Fig. 9 shows the model output values from the Strathbury proposed development.

PROPOSED DEVELOPMENT

Development #:	2.1	R1 units:	305	Frontage(m):	20722
Yrs to start:	2.0	R2sd25 units:	211	Land $:	7689489
Yrs to finish:	10.0	R2sd30 units:	202	Cnst&mkt $/m:	11000.00
Gross area:	797904	R2Xdup units:	161	Const&mkt $:	227946101
Wetland area:	164400	R2Xatt units:	76	Move wet $:	92080
Wetland moved:	36832	R1 width:	15.00	MR comp $:	0
Moved Size:	55248	R2sd25 width:	7.60	Income:	301589620
Dev'able Area:	593586	R2sd30 wdith:	9.00	% profit:	28
Res. area:	593586	R2Xdup width:	7.50		
Density:	6.500	R2Xatt width:	6.00		
%MR ded.:	0.100	Lot depth:	37.54		

Fig. 9. The model output values from the Strathbury proposed development

A comparison of the model results with the Outline Plan and Land-use Redesignation application development is shown in Table 1. The values of the model results versus the values contained in the Outline Plan and Land-use Redesignation application are within 0.1 to 3% of each other, a reasonable consistency for this research.

The undeveloped Strathbury parcel designated as urban reserve (UR) contains eleven wetlands (Fig. 10a). A road splits the east and west portions of the parcel. The west portion is bounded by the Western Irrigation District (WID) Canal on the west, public service to the north, and residential to the south. The east portion contains more wetlands; it is bounded by wetlands to the north, the WID Canal to the east and residential areas to the south. Fig. 10b shows the model results of the approved Strathbury land-use redesignation to residential, environmental reserve, and municipal reserve and the displacement and consolidation of eight wetlands. A detailed view of the actual land-use allocation map contained in the documents for the Strathbury Outline Plan application is displayed on Fig. 10c for comparison purposes. The percentage of each land-use type allocated by the model matches reasonably well the values contained in the actual Strathbury Land-use Redesignation and Outline Plan application.

Variable	Strathbury model results	Outline Plan application	Difference (%)
Gross Area (m^2)	797904	797229	0.1%
Environmental reserve (m^2)	201232	203557	1%
Developable area (m^2)	593586	593673	0.01%
Number of residential units	955	954	0.1%
Number of R1 units (units)	305 (32%)	315 (33%)	3%
Number of R2 units (units)	413 (43%)	410 (43%)	0.7%
Number of R2X units (units)	237 (25%)	229 (24%)	3%

Table 1. Comparison of the model results and the Outline Plan application development

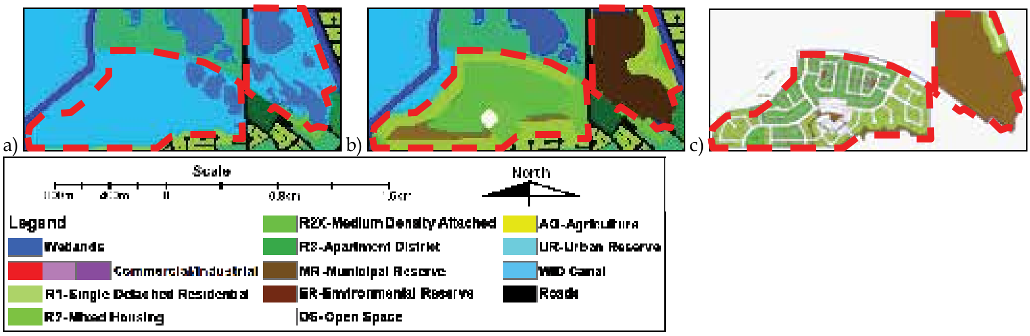


Fig. 10. a) The model land-use map prior to any development, b) the model land-use map of the Strathbury development, and c) the actual land-use map from the Strathbury Outline Plan and Land-use Redesignation application

Based on the initialization of the model with proper parameters values, the model produces results that are reasonably similar to the actual documents and plans of the Strathbury residential land development project. In general, the model adequately mimics the Strathbury Outline Plan and Land-use Redesignation application by the WestCreek developer, the opinion of the citizens, the decision by the Town to approve the Redesignation, and the land-use change.

3.2 Model run results

When analyzing the raster maps in the following scenarios, one must keep in mind that, contrary to reality in a land development process, the model does not include a detailed engineering infrastructure including water, gas and electrical supply and distribution, storm water collection and overland drainage, storm and sewer discharge, and road network.

3.2.1 Business as usual scenario (BAU)

The “business as usual” (BAU) scenario is based on an initialization from the Strathbury development goals and regulations and projecting into the future. Due to the age of the MDP the Developer is given additional leniency on the *Consistency with MDP* to 10%. Initially the Planner has a *happiness* of 9.0 to 10.0, continuously approving redesignation applications, trying to keep up with the *housing demand* and increasing the *development potential*. On three occasions the Citizen gives the proposed developments a negative *opinion* due to wetlands being displaced, but in general has a *happiness* of 4.0 to 6.0. During the years of approvals, the Developer has a *happiness* of 8.0 to 9.0 making profit. During year 6, the Planner requests a "decrease in development time", speeding up the potential for new housing, increasing the Developer *happiness* to 10.0. By year 9 the *development potential* has surpassed the *housing demand* and in the subsequent years the Planner requests revisions to "increase the development time", slowing the potential for new housing, decreasing the Developers *happiness* to 4. By the end of the simulation time, the *development potential* is 170% that of the *housing demand*.

Fig. 11a shows the land-use change over the ten year simulation period based on the BAU goals, regulations, standards and market parameter initialization. During the ten years of simulation, land-use change occurs over approximately 280 hectares contained within 17 land parcels. The percentage allocation of each residential land-use type follows the MDP regulations and *Housing type market demand*, and *Market demand to MDP ratio* of the Developer as expected: 47% R1, 23% R2 and 30% R2X.

3.2.2 Reduction in development approvals per year scenario

This model simulation starts with the Planner *proposals per year* reduced to one. During the simulation, the Planner and Developer maintain a *happiness* of 9.0 to 10.0, the Planner again continuously approving redesignation for proposed developments, trying to keep up with the *housing demand* and increasing the *development potential*, and the Developer content having all proposed redesignation applications approved. The Citizen has a slightly higher *happiness*, of 5.0 to 6.0, than in the BAU scenario, having fewer wetlands impacted. Over a ten year period, this scenario resulted in the *development potential* being 17% less than the *housing demand*.

Fig. 11b shows the land-use change over the ten year period based reduction in development approvals per year. Compared to the BAU scenario, the impact on land use occurs over a much smaller area, approximately 174 hectares contained within 10 land parcels; however, as expected, the land-use allocation and patterns are the same.

3.2.3 Increase in density scenario

This model simulation is initialized with a bylaw regulation *density* of seven units per acre. The Planner maintains a *happiness* of 9.0 to 10.0 throughout the simulation; the Developer's *happiness* is 9.0 to 10.0 for the first five years until the *housing demand* is met, when redesignation applications are no longer approved. The Citizen's *happiness* is 2.0 to 3.0 throughout the simulation due to the increase in density. This scenario results in the *development potential* being 70% more than the *housing demand* providing for a potential flood in the housing supply.

Fig. 11c shows the land-use change over the ten year period based on the business as usual parameter initialization, except for the increase in land-use bylaw density. Again, compared to the BAU scenario, the impact on land use is significantly less, being approximately 205 hectares contained within 14 land parcels, with similar land-use allocation; however essentially the same population is accommodated.

3.2.4 Change in market housing demand scenario

Ewing et al. (2007) predict that the demand for residential housing types is going to change over the next 15 years with the retirement of baby boomers. The demand for larger lot homes is going to decrease as the number of households with children decreases and the demand for smaller lots and attached homes will increase as the number of retired and single-person households' increases. The current demand for large lot homes, small lot homes and attached homes is about 50%, 25%, and 25%, respectively while the predicted demand is about 35%, 30%, and 35%, respectively (Ewing et al. 2007).

In order to accommodate this type of expected future change of community's needs, several key factors will most likely have to be modified including a new Municipal Development Plan (MDP), revisions to zoning bylaws, and altering public opinion. The creation of a new MDP and the revision of zoning bylaws are straight forward processes for the town planner and council, typically involving inter-departmental consultation, urban and infrastructure planning and limited public consultation. Changing public opinion on the other hand would require a considerable amount of time on the part of the town planner, who will have to educate the citizens on the benefits of planning to accommodate future change in the community. Unfortunately, changing community values and opinions is not typically an easy task.

In the case of the change predicted by Ewing et al. (2007), population distribution will create a change in demand for types of residential housing. Current young families in a community may not be that accepting of a future community of aging and single persons and therefore may not give any heed to planning for such a future, making the planner's job difficult. That being said, the implementation of this scenario in the model is actually quite easy; it involves: modifying the MDP values to allow for more small lots and attached homes and fewer large lot single family homes by increasing the percentages of R2 and R2X and decreasing the percentage of R1 land use; modifying the land-use bylaws to allow for higher density; and modifying the housing type market demand values to match the prediction.

The initialization of this scenario relies on the following assumptions: the MDP reflects the expected change in market demand; the planner requires more consistency with the new MDP; the density of the land-use bylaws is increased to seven units per acre; the Citizen has been educated in the changes to the MDP and land-use bylaws as well as the new building fire codes on external walls and relaxes his goal of increasing building setback; the Developer has been educated in the changes to the MDP and land use bylaw and therefore changes his density goal as well as one to one goal for market demand versus MDP.

With this scenario the Citizen's *happiness* is 6.0 to 8.0 throughout the entire simulation time. Like the business as usual scenario the Planner maintains a *happiness* of 9.0 to 10.0 throughout the simulation; the Developer's *happiness* is 9.0 to 10.0 for the first five years until the *housing demand* is met, when redesignation applications are no longer approved. During the following five years the planner continuously requests revisions, which increases the development time leaving the Developer with no choice but to not submit land-use redesignation applications causing the Developer *happiness* to fall to 2.0 to 3.0. By the end of the simulation the *development potential* is 30% greater than the *housing demand*.

Fig. 11d shows the land-use change over the ten year period based on change in market housing demand. Compared to the BAU scenario, the area over which land use has changed is significantly smaller, being approximately 176 hectares contained within 11 land parcels. The percentage of each residential land-use type follows the revised MDP regulations and *Housing type market demand* as expected: 33% R1, 31% R2 and 36% R2X, which is also visible in the land-use map.

3.2.5 Sustainable development scenario

This simulation is a combination of the "reduction in development approvals per year" scenario and the "change in market housing demand" scenario, but also includes a decrease in the dedication of wide streets from 34% wide (21 m in width) and 66% narrow (15 m in width) streets to 5% wide and 95% narrow streets, and a change of the Developers goal of *Willingness to move wetlands* to no displacement of wetlands.

Throughout this scenario, the *happiness* of all three agent types remains between 7.0 and 10.0. The Planner manages to meet the *housing demand* by year 9, the Developer has his proposed redesignation applications approved, and the Citizen has been educated on the new MDP and is pleased to that see wetlands are not being impacted.

Fig. 11e shows the land-use change over the ten year period based on the sustainable development scenario. Compared to the BAU scenario, the area over which land use has changed is significantly smaller, being approximately 198 hectares contained within 11 land parcels, and is only slightly larger than both the "change in market housing demand" and the "reduction in development approvals per year" scenarios. As expected the land-use allocation is similar to the "change in market housing demand" scenario. The change in the Developer goal to not disturb wetlands creates more intricate land-use patterns, presumably a more interesting community; however it possibly creates more complex roads and utility infrastructure.

3.3 Validation of the model

The results of the different scenarios were presented to the Planner of the Town of Strathmore and the Civil Engineer contracted by WestCreek Developments for the Strathbury development, both of whom are considered experts in their field; they were asked to provide comments and criticisms. The persons were selected based on their knowledge, their interest in this research, and their approachability. The following written comments were received from the experts on the model in general and on the results of the simulated scenarios (Engineer 2009; Planner 2009): (1) "I like the overall concept, and I think this is a great model" and "your model has some interesting elements."; (2) "You have tried to represent the data quantitatively, but I think this is a huge challenge to nail exactly."; (3) "Planning is very political and I don't know how you would accurately represent this in your model."; (4) "This type of a model would work better for a developer and their consultants."; (5) "The model did show how the various scenarios impacted the consumption of available land. This is a very positive outcome of the model."; (6) "The model predicted very similar results regardless of the scenario you used. The major exception was the ER/MR which is more obvious based on the selection of options."; (7) "The model could use work on ER allocation by including environment data (contours, trees, water, etc.), and MR allocation by including community needs (schools, recreation needs, etc.)."; (8) "The impact of higher density (although you used only a moderately high density) did not show significant differences in single or multifamily development. Perhaps because the density used was low or because of the decision process."; (9) "The municipality really has little say on the decision to build multi-family projects; rather it is the developer who keeps a keen eye on the market and will request this type of zoning when they feel the market is there ($$). The municipality should not (and normally does not) interfere in the market, choosing one development to get an advantage over another by agreeing to multi-family for one and not another."; (10) "I did not see commercial and industrial allocation on the plans."; and (11) "I feel this tool, as it matures, can be valuable to municipalities as they try to stay ahead of demand and look towards the future."

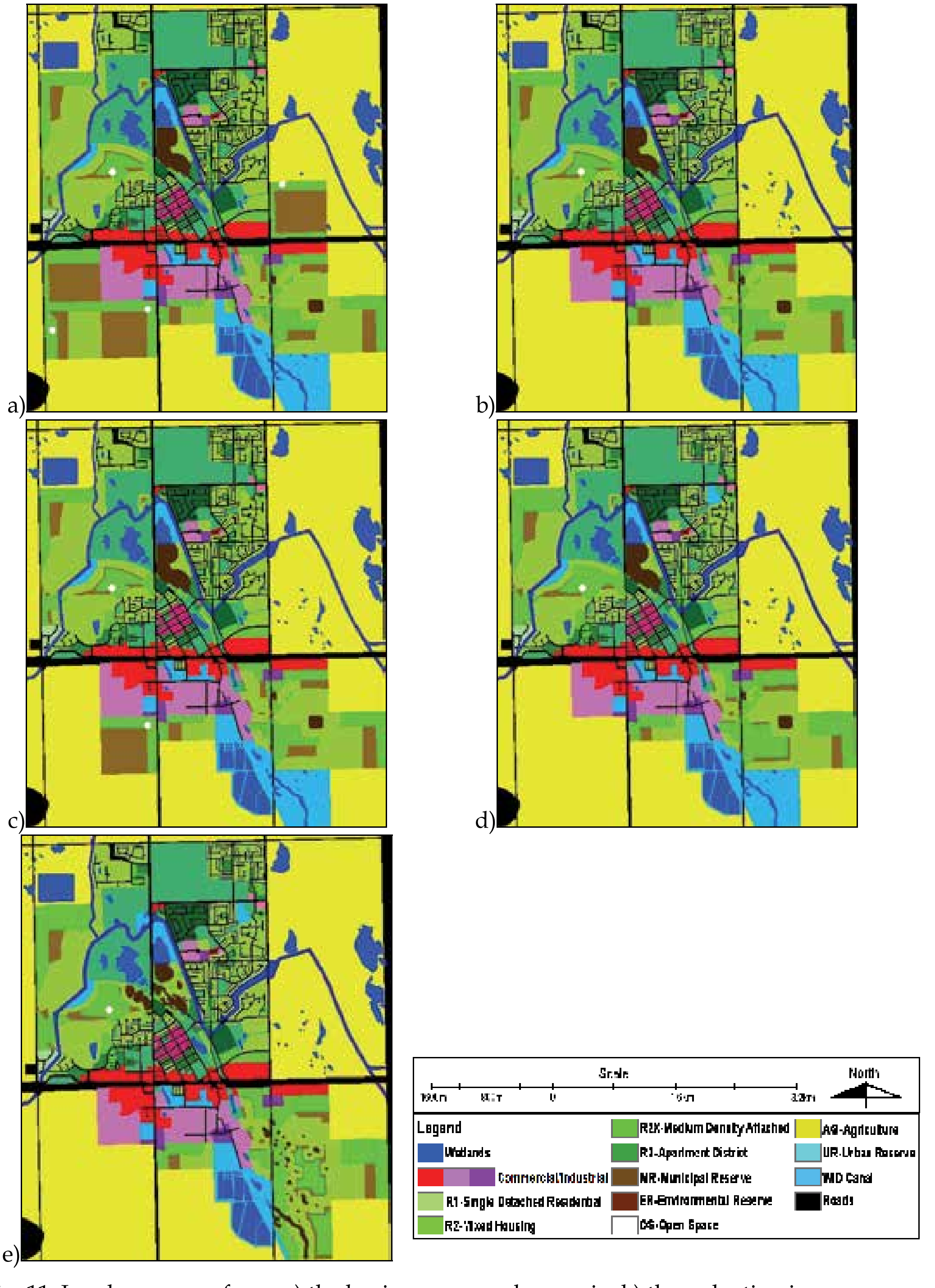


Fig. 11. Land-use maps from: a) the business as usual scenario, b) the reduction in development approvals scenario, c) the increase in density scenario, d) the change in market housing demand scenario, and e) the sustainable development scenario

The different scenarios shown above present the potential for this type of model in simulating the impact of agent goals, government regulations, design standards and market demand on land-use change, land-use patterns and pace of growth. The feedback offered by the experts provided input on the value and application of the model. It also provided interesting observations on the similarities and differences between the scenarios, as well as a potential problem in land-use allocation and the exclusion of some land uses. Constructive criticism was also given on the limitation of not including politics in the model, but also the understanding of the complexity of attempting to include politics. The comment regarding the usefulness of the model exclusively for a developer probably originates from the fact that the *land development planning* was approached from the perspective of an engineer, which is discussed in the conclusion, rather than the perspective of an urban planner and as such excludes the politics of the process. The model was also criticized for the misplacement of development control onto the Planner, as development applications approved per year, rather than onto the Developer, as development applications submitted per year.

4. Conclusion

This model is among the first attempts to contribute to the field of agent-based modelling in the Geomatics Engineering, Civil Engineering and Land-use Planning disciplines. This research has identified the fundamental parameters relevant to the *land development planning process* through a case study. The process was formalized into a simple agent-based model that accounts for social, economic, regulatory, and environmental factors. The results of the model and the comments provided by the experts show that this model has the potential to provide insight into the impact of municipal planning policies and stakeholders' goals in residential land development planning.

This research was primarily focused on the decision-making process during the *land development planning process* of the Strathbury residential land development case study. The creation of this model from an engineering aspect gave a very quantitative approach to the interview process and conceptual model development, establishing stakeholder goals as numerical figures, quantifying those factors impacting the decision-making process, and deriving formulas to mimic the decision-making process. The model allows stakeholders to test different goals and policies, providing the opportunity to quickly analyze possible future impacts before their implementation.

Since social, economic, regulatory, and environmental factors throughout Canada share several similarities, the model developed in this project has the potential to be replicated for use within another small community almost anywhere in Canada with relatively minor modifications to the parameter values. However, adapting the model to different stakeholder goals would require modifications to the computer code.

By proving the value of a simulation model having few stakeholders and a small geographical extent, the model could in the future be enhanced to include a larger number of stakeholders having more complex interactions and expanded to cover a much larger and more complex region. There are many regions in the Province of Alberta and elsewhere that would benefit from such a model.

4.1 Future improvements to the model

As mentioned by one of the experts when validating the model results, residential development cannot exist without the addition of various service sectors. The model therefore needs to be expanded to include other land development in other sectors that co-exist with residential including commercial, industrial and service.

A large number of parameters were used in this model. A sensitivity analysis can identify the parameters that are highly correlated, possibly allowing for the removal of some of them while still achieving comparable results. A sensitivity analysis can also determine the parameters that are the most influential on the system, whereby slight changes in the parameter value give rise to significant changes in the system. Based on the model results, it is believed that density and population are sensitive parameters. Since the willingness to disturb wetlands is a binary parameter, it would also be a sensitive parameter. Future work should include a sensitivity analysis of the model parameters.

The model attempts to simulate the change of behaviour of the stakeholders as the change of goals of the agents in a binary quantitative fashion. The direction of behavioural change is quite often easily derived; however the magnitude of change is not so easy. Determining the level of happiness or unhappiness, or the amount of behavioural change is quite subjective and falls into the realm of "fuzzy logic". Ligtenberg et al. (2001) state that fuzzy set theory should be explored in agent-based modelling as a way to enhance the decision making for agents. Implementing such techniques would require the expansion of interview questions to ask about past experiences, the decisions that were made and the resulting change in behaviour. The questionnaire could ask what if scenarios posed from different standpoints attempting to get a consistency in answers.

This research overlooked the "NIMBY" (Not In My Back Yard) apprehension as a behaviour of the citizen agent. In the real world, although it is not required by the developer, communities are quite often pleased when a developer requests their opinions regarding the development proposal in a voluntary pre-application meeting. They are often displeased and alarmed when they are not included only to receive the formal application and typically voicing disapproval during the public hearing. We hypothesize that "NIMBY" is a contributor to urban sprawl. Since residents of an established community have not dealt with construction and disturbance for many years, their community has most likely developed an identity and values, they may have developed a bond with their "backyard", and they have most likely established community organizations. A proposal to develop adjacent lands would create disturbance during construction, create additional traffic upon completion, and it may not fit their identity or values. Established communities will most likely oppose the proposed development and organize to defend "their backyard". This may cause developers to choose not to propose a development adjacent to an existing community having a large opposition, but rather choose land that is further away that is the "backyard" of only a few land owners driving the "urban sprawl" machine. Residents in a new community are already dealing with the construction and disturbance within their own community; their community perhaps has not developed an identity or community values, they may not have a bond or with their "backyard", and neighbourhood organizations have not been established. A proposal to develop lands adjacent to a new community creates little additional inconvenience; the residents have little bond with their adjacent lands, and they are not organized to defend their community. New communities will most likely give little consideration to the impact of new adjacent developments on their community; this lack of opposition may also be a contributor to the "urban sprawl" machine. The "NIMBY" apprehension could have been implemented as a function of the number of existing homes directly adjacent to the proposed development.

There are improvements that should be made to the model including: the shift of development control from the Planner to the Developer, acquiring actual property market value data, and updating the Municipal Development Plan (MDP). Other additions that would increase the value of the model include: (1) changing housing market value based on supply and demand; (2) adding a landowner agent to make decisions regarding the sale of their land; (3) creating a module that varies property value based on the adjacent land-use; (4) having environmental factors (topography, vegetation, and habitat) in the allocation of land use; (5) adding a utility company agent and utility infrastructure data impacting land-use change patterns; and (6) creating a dynamic Municipal Development Plan that can slowly implement the long term goals of the planning authority including major transportation corridors.

As mentioned in section 2.5.1.2, an attempt was made to allow for behavioural change of the agents. The properties of the agents would vary based on their *happiness* and how successful their *opinion* was in a decision to which it contributed. Behavioural change proved to be too difficult to implement primarily due to a lack of information on actual behavioural change from the stakeholder representative, but also due to the multiple possibilities for change. Behavioural change could come as extreme approval (greed) or disapproval (protest), concede or persevere, or remain unchanged. As an example, a happy agent whose opinion was noticed might concede a little, might remain unchanged or might persevere further. An unhappy agent whose opinion was noticed might not change, might persevere a little or might become greedy. An unhappy agent whose opinion was ignored might protest or might concede.

5. Acknowledgements

The authors would like to extend their sincere thanks to the citizens and the planner at the Town of Strathmore, and WestCreek Developments for their participation in the study. This research has been funded by a NSERC grant awarded to D. Marceau and a research grant awarded to M. Kieser by the University of Calgary. We also thank the two experts for their valuable comments on the model validation.

6. References

Alberta Government of (2000). Guidelines for the Approval and Design of Natural and Constructed Treatment Wetlands for Water Quality Improvement. E. S. D. Municipal Program Development Branch, Environmental Service, Alberta Environment.

AltaLIS Ltd. (2007). AltaLIS Cadastral. Calgary, AltaLIS Ltd.**:** cadastral parcel data from Maps, Academic Data, Geographic Information Centre (MADGIC), University of Calgary

Anwar, M., C. Jeannert, et al. (2007). Conceptualization and implementation of a multi-agent model to simulate whale-watching activities in the St. Lawrence estuary in Quebec, Canada. *Environmental Modelling and Software*(22): 1775-1787.

Applied History Research Group (1997-2001). Calgary & Southern Alberta. Calgary, University of Calgary. 2009**:** Calgary's Growing Population chart.

Batty, M. (2001). Agent-based pedestrian modeling. *Environment and Planning B* 28(3): 321-326.

Beardsley, K., J. H. Thorne, et al. (2009). Assessing the influence of rapid urban growth and regional policies on biological resources. *Landscape and Urban Planning* 93(3-4): 172-183.

Bousquet, F., C. Le Page, et al. (2001). Multiagent simulations of hunting wild meat in a village in eastern Cameroon. *Ecological Modelling* 138(1-3): 331-346.

Bowman, T. and J. Thompson (2009). Barriers to implementation of low-impact and conservation subdivision design: Developer perceptions and resident demand. *Landscape and Urban Planning* 92(2): 96-105.

Brown and Associates Planning Group (2008). Town of Strathmore Growth Study. Strathmore, AB, Town of Strathmore.

CH2M HILL (2007). CRP Regional Servicing Study: Population Projections. Calgary, Calgary Regional Partnership,.

Citizens (2006). Letters (6) RE: Bylaw #06-17. Town of Strathmore.

City of Calgary (2008). The City Calgary Population Growth 1998-2008. Calgary, City of Calgary.

City of Calgary (2008). Guide to the Planning Process. L. U. P. Policy, City of Calgary.

City of Calgary (2009). Population Picture. Calgary, City of Calgary.

City of Calgary and Federation of Calgary Communities (2002). A Community Guide to the Planning Process. C. o. Calgary.

Crooks, A. (2006). *Exploring Cities using Agent-based Models and GIS*. Agent 2006 Conference on Social Agents: Results and Prospects, Chicago, USA.

Developer (2007). Interview with land developer stakeholder conducted by Michael Kieser: to determine his goals, behaviour and factors influencing his decision during the Outline Plan and Land Use Redesignation process of the Strathbury residential land development project.

Energy Resources Conservation Board (2007). Well sites. Calgary, ERCB: cadastral parcel data from Maps, Academic Data, Geographic Information Centre (MADGIC), University of Calgary

Engineer (2009). Letter: expert opinion validating the results of the model. Michael Kieser.

Ewing, R., K. Bartholomew, et al. (2007). *Growing Cooler: Evidence on Urban Development and Climate Change*. Chicago, Urban Land Institute.

Feuillette, S., F. Bousquet, et al. (2003). Sinuse: A multi-agent model to negotiate water demand management on a free access water table. *Environmental Modelling and Software* 18: 413-427.

Gimblett, H. R., C. A. Roberts, et al. (2002). An intelligent agent model for simulating and evaluating river trip scenarios along the Colorado River in Grand Canyon National Park. *Integrating Geographic Information Systems and Agent-Based Modeling Techniques for Simulating Social and Ecological Processes*. H. R. Gimblett. Oxford, U.K., Oxford University Press: 245-276.

Gimblett, R., T. Daniel, et al. (2001). The simulation and visualization of complex human-environment interactions. *Landscape and Urban Planning* 54(1-4): 63-79.

Government of Alberta (2000). Municipal Government Act. *M-26*. M. Affairs, Alberta Queen's Printer. RSA 2000.

Janssen, M. A., B. H. Walker, et al. (2000). An adaptive agent model for analysing co-evolution of management and policies in a complex rangeland system. *Ecological Modelling* 131(2-3): 249-268.

Ligtenberg, A., A. K. Bregt, et al. (2001). Multi-actor-based land use modelling: spatial planning using agents. *Landscape and Urban Planning* 56(1-2): 21-33.

Lim, K., P. Deadman, et al. (2002). Agent-based simulations of household decision making and land use change near Altamira, Brazil. *Integrating Geographic Information Systems and Agent-Based Modeling Techniques for Understanding Social and Ecological Processes*. H. R. Gimblett. Oxford, U.K., Oxford University Press.

Malczewski, J. (1999). *GIS and multicriteria decision analysis*. Toronto, John Wiley & Sons, Inc.

Malczewski, J. and R. Moreno-Sanchez (1997). Multicriteria group decision-making model for. *Journal of Environmental Planning & Management* 40(3): 349.

Marceau, D. J. (2008). What can be learned from multi-agent systems? *Monitoring, Simulation and Management of Visitor Landscapes*. R. Gimblett, University of Arizona Press: 411-424.

Mason (2010). Multi Agent Simulation Of Neighbourhood. 2010.

Minar, N., R. Burkhart, et al. (1996). The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations. Santa Fe, New Mexico, Santa Fe Institute.

Monticino, M., M. Acevedo, et al. (2007). Coupled human and natural systems: A multi-agent-based approach. *Environmental Modelling & Software* 22(5): 656-663.

Moreno, N., R. Quintero, et al. (2007). "Biocomplexity of deforestation in the Caparo tropical forest reserve in Venezuela: An integrated multi-agent and cellular automata model." *Environmental Modelling & Software* 22(5): 664-673.

Musiani, M., M. Anwar, G. McDermid, M. Hebblewhite, and D.J. Marceau, (2010). How humans shape wolf behavior in Banff and Kootenay national parks, Canada. *Ecological Modelling* 221: 2374-2387.

NetLogo (2010). Multi-agent Programmable Modeling Environment. 2010.

Parker, D. C., T. Berger, et al. (2002). Agent-Based Models of Land-Use/Land-Cover Change: Report and Review of an International Workshop. W. J. McConnell. Bloomington, IN, LUCC Focus 1, Indiana University.

Planner (2007). Interview with planner stakeholder conducted by Michael Kieser: to determine his goals, behaviour and factors influencing his decision during the Outline Plan and Land Use Redesignation process of the Strathbury residential land development project.

Planner (2009). Letter: expert opinion validating the results of the model. Michael Kieser.

Repast (2010). Recursive Porous Agent Simulation Toolkit. 2010.

Statistics Canada (2007). 2006 Community Profiles - Strathmore, Alberta. 2006 Census. Ottawa, Statistics Canada: Catalogue no. 92-591-XWE. Released March 13, 2007.

Swarm (2010). Swarm: A Platform for Agent-Based Models. 2010.

Town of Strathmore (1989). Strathmore Land Use Bylaw, Town of Strathmore,. Bylaw #89-20.

Town of Strathmore (1998). Municipal Development Plan, Town of Strathmore. Bylaw #98-11.

Town of Strathmore (2007). Land use mapping. Strathmore, Town of Strathmore: Digital shapefile.

Town of Strathmore (2008). Strathmore_Asmt_Data.xls. Strathmore, Assessment Office. 2008: Assessment data for land parcels having specific LINC (Land Identification Numeric Code) numbers in the Town of Strathmore.

Waddell, P. (2002). UrbanSim: Modeling urban development for land use, transportation, and environmental planning. *Journal of the American Planning Association* 68(3): 297-314.

Wainwright, J. and M. Mulligan (2004). *Environmental Modeling: Finding Simplicity in Complexity*. Chichester, England, Wiley and Sons.

Wheatland County (2007). Wheatland_Asmt_Data.xls. Strathmore, Accurate Assessment Group Ltd. 2007.

White, E. M., A. T. Morzillo, et al. (2009). Past and projected rural land conversion in the US at state, regional, and national levels. *Landscape and Urban Planning* 89(1-2): 37-48.

Wooldridge, M. J. (2000). *Reasoning about Rational Agents*. Cambridge, Massachusetts, The MIT Press.

Autonomous and Intelligent Mobile Systems based on Multi-Agent Systems

Adil Sayouti, Hicham Medromi and Fouad Moutaouakil
Laboratoire d'Informatique, Systèmes et Énergies Renouvelables
Team Architecture of Systems, ENSEM, Hassan II University
BP 8118, Oasis, Casablanca
Morocco

1. Introduction

The control of current robotic systems in manufacturing industry and the service sector has remained separate and independent. In other words, these robotic systems are isolated from one another by different environments and have no effective way to communicate. This has made the current robotic systems expensive and requiring a long developing cycle, which has in turn seriously hampered the day-to-day deployment of robot technology. Therefore it is crucial to develop an integrated network environment for robotic systems based on today's Internet technology. With the rapid growth of the Internet, more and more intelligent devices or systems have been embedded into it for service, security and entertainment, including distributed computer systems, surveillance cameras, telescopes, manipulators and mobile robots. Although the notion of Internet robotics or web-based robotics is relatively new and still in its infancy, it has captured the huge interest of many researchers worldwide. Except for operating in hazardous environments that are traditional telerobotic areas, Internet robotics has opened up a completely new range of real-world applications, for example in the following fields (Le parc & al., 2008):

1. Tele-teaching: a lot of universities are using robots to teach the basics of electrical engineering. The profitability of these robots is of course really poor because they are only used a few weeks a year. Why not developing common centers where students may have access to real robots without being close to them? One of the problems of e-learning is to make practical experiments. Why not using Internet technologies to let distant students to manipulate real systems?
2. Tele-maintenance: when a company is shipping systems all over the world, it needs to technicians when one of its systems has some failures. With Internet technologies, it is now possible to make some remote diagnostics, to solve and repair some problems, to prepare the right equipment to send, etc.
3. Tele-expertise: some specific operations on robotic systems can only be made by expert. In a close future, it will become possible for experts to operate from their office a machine located somewhere in the World, just using classic web technologies.
4. Tele-production: the remote access possibilities and taking control will make work easier for remote users and will allow the performances of more tasks in the future.

The use of Internet will reduce the costs of these activities. The increase of Internet abilities in term of speed and bandwidth in the future, let us also think that the quality of the remote control and the comfort of the user will also increase. But, when developing such applications, we have to think that these activities rely all the time on an unpredictable network and that we have to build them taking into account this parameter. The organization of a mobile system - or its control architecture - determines its capacities to achieve tasks and to react to events. In this chapter we propose a remote control architecture based on multi agents systems to take into account the lack of quality of services of Internet.
This chapter is presented as follows: on the next section, we present the multi-agent approach. Then we will describe some projects of remote control of robotic systems. In section 4, we propose a system architecture that allows such a control on Internet. The state of the art of control architecture is detailed in section 5. In the section 6, we present the proposed control architecture based on multi-agents systems. Then we will describe our remote control software architecture. In section 8, we present some applications of remote control of an autonomous Lego mobile robot as illustrative examples. Finally, some conclusions and future developments are presented in section 9.

2. Multi-agent systems for autonomous control

The organization of a system - or its control architecture - determines its capacities to achieve autonomous tasks and to react to events (Novales & al., 2006). The control architecture of an autonomous mobile system must have both decision-making and reactive capabilities: situations must be anticipated and the adequate actions decided by the mobile system accordingly, tasks must be instantiated and refined at execution time according to the actual context, and the mobile system must react in a timely fashion to events. This can be defined as a rational behavior, measured by the mobile system's effectiveness and robustness in carrying out tasks.
To meet this global requirement, the control system architecture should have the following properties (Alami & al., 1998):

1. Tele-teaching: a lot of universities are using robots to teach the basics of electrical engineering. The profitability of these robots is of course really poor because they are only used a few weeks a year. Why not developing common centers where students may have access to real robots without being close to them? One of the problems of e-learning is to make practical experiments. Why not using Internet technologies to let distant students to manipulate real systems?
2. Programmability: a useful mobile system cannot be designed for a single environment or task, programmed in detail. It should be able to achieve multiple tasks described at an abstract level. The functions should be easily combined according to the task to be executed.
3. Autonomy and intelligence: the mobile system should be able to carry out its actions and to refine or modify the task and its own behavior according to the current goal and execution context as perceived.
4. Reactivity: the mobile system has to take into account events with time bounds compatible with the correct and efficient achievement of its goals (including its own safety). Consistent behavior: the reactions of the mobile system to events must be guided by the objectives of is task.

5. Robustness: the control architecture should be able to exploit the redundancy of the processing functions. Robustness will require the control to be decentralized to some extent.
6. Extensibility: integration of new functions and definition of new tasks should be easy. Learning capabilities are important to consider here: the architecture should make learning possible.

We note an interesting link between the desirable properties of intelligent control architecture for autonomous mobile systems and the behavior of agent-based systems (Ferber 1999):

1. Agent-based approaches to software and algorithm development have received a great deal of research attention in recent years and are becoming widely utilised in the construction of complex systems.
2. Agents use their own localised knowledge for decision-making (see Figure 1), supplementing this with information gained by communication with ist environment and the other agents.
3. Remaining independent of any kind of centralised control while taking a local view of decisions gives rise to a tendency for robust behavior.
4. The distributed nature of such an approach also provides a degree of tolerance to faults, both those originating in the software/hardware system itself and in the wider environment.
5. Multi-agent systems can manifest self-organization and complex behaviors even when the individual strategies of all their agents are simple.
6. Agents can share knowledge using any agreed language, within the constraints of the system's communication protocol. Example languages are Knowledge Query Manipulation Language (KQML) or FIPA's Agent Communication Language (ACL).

It is for these reasons that we consider an agent-based system to be a suitable model on which to base an intelligent control architecture for complex systems requiring a large degree of autonomy.

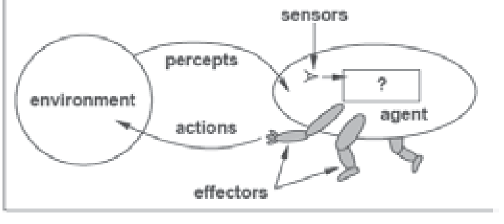


Fig. 1. Agents interact with environments through sensors and effectors

3. Remote control over Internet – state of the art

During the Nineties, several projects appeared of robotic systems control, using Internet as communication network (Goldberg & al., 2001) with various objectives.

The **Mercury Project** (Goldberg & al., 2000) is believed to be first system that allowed Internet users to remotely control robotics via Internet. This project launched the first system allows users to alter the real world. This project is initialized by an interdisciplinary team of anthropologists and computer scientists. They want to explore an area dubbed “Mercury site”. Because nobody can work in this dangerous area, the remote robotics is a good choice.

The remote control of the robot is designed to excavate the surface with short burst of compressed air and then the surface is revealed and the relevant data can be collected by the anthropologists. After the success of the initial exploration, the site is open to all the researchers who are interested in having a remote control of the robot via Internet. The successes of Mercury Project is not only on its excavation purpose, but also showed the possibility of control the robot via Internet. This is the milestone on Internet telerobotics, more and more Internet telerobotics projects were launched in the later years.

Telegarden is the second Internet telerobotics from Goldberg and al. This Telegarden system allows Internet users to view and interact with a remote garden filled with living plants (see Figure 2). Users can plant, water, and monitor the progress of seedlings via the tender movements of an industrial robot arm.

Fig. 2. Telegarden

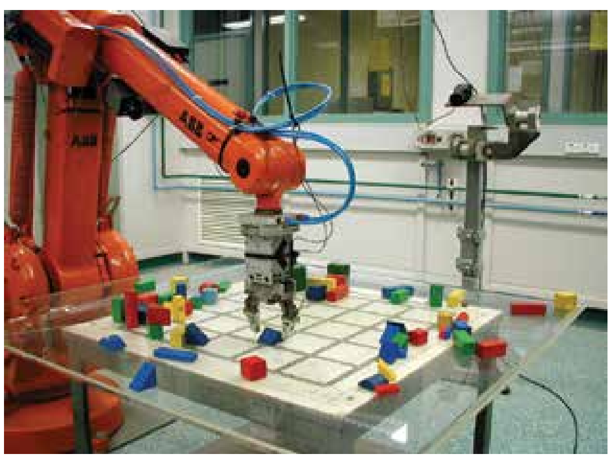


Fig. 3. Australian Telerobot on the Web

The **Australian Telerobot** on the Web project (Taylor & Taylor, 1997) is established by Taylor at The University of Western Australia. A six degree of freedom robotic arm is controlled through Taylor's Internet telerobotics system (see Figure 3). The robotic arm can

play the wooden block on a table. The user connects the server via Internet connection, and can log to the system with an identification check. This project is very successfully due to it's an interesting wooden block game as well as users can play it via Internet.
The launch of **RHINO project** (Schulz, 1997) indicates the possible potential of Internet robotics in daily life. RHINO project (see Figure 4) is initially launched for a museum of contemporary technology in Bonn Germany. Visitors of the "Deutsches Museum Bonn" will have the opportunity to be shown through the museum by a mobile robot "RHINO". RHINO can provide the user with the information they concerned as well as more information in deep upon request.

Fig. 4. RHINO robot

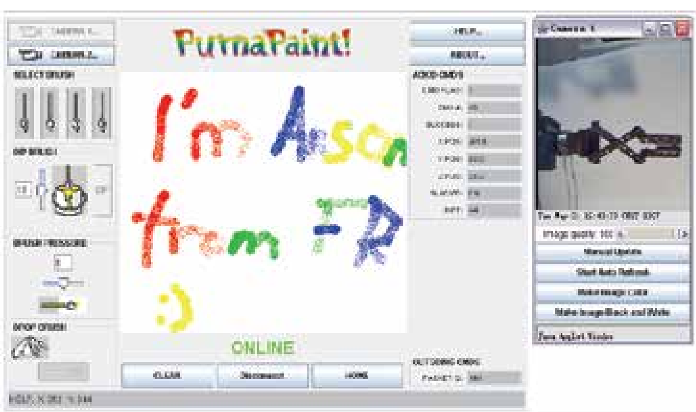


Fig. 5. PumaPaint interface

The research focus of **Xavier** (Simmons & Xavier, 1998) is to study the local intelligence of the robot as well as users' interface. The research team has considered the supervisory control which aims to give robot the command at a higher level. This scheme can reduce the influence of Internet time delay, but prevents more interaction between users and robot. In this situation, users can't interact with the robot immediately. Remote users need fast feedback (image) when controlling the robots on the web facing unpredictable Internet time

delay (limited bandwidth). The supervisory control can indeed reduce the bandwidth requirement, but at the same time reduces interactivity. These existed problems are highly concerned in the later years' research of Internet telerobotics.
The **PumaPaint project** (Stein, 1998) came out from the collaboration between University of Wisconsin and Wilkes University (see Figure 5).
The computer science department at UW needed a PUMA robot as undergraduate teaching resource. But if they develop and maintain the system themselves, it's money and time costly. So, they decide to share the installed one at Wilkes University. Students from University of Wisconsin can access the PUMA robot via Internet connection.
The system is developed in Java considering the cross platform advantage of java as well as the reusability of Java program. The Java Virtual Machines (JVM) is involved in the development, every machine need to install this before using the system, but the JVM is quite popular in most of web browsers.
The possibility of piloting a **Khepera robot** (Saucy & Mondada, 2000) was made available to the general public over the web in December 1996 (KhepOnTheWeb). By means of a WEB client (Netscape), the user can move forward or turn the robot and receive images of the remote environment (see Figure 6). He can also choose the point of view and receive images either from the robot's on-board camera or from a camera mounted on the ceiling.

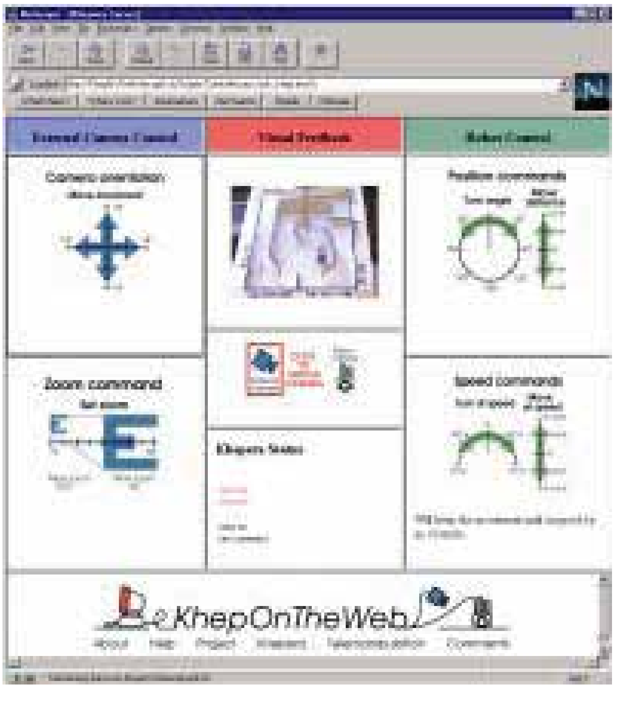


Fig. 6. The remote control interface

All these experiences are really interesting because they have treated the problem of remote control in different manners and in different contexts. To develop safe and evaluated web-based remote control, one has to take into account all of them. Nevertheless, we can notice that none of these works have tried to develop some control architecture and the unpredictable nature of the Internet is not really taking into account with all the consequences. In the next sections, we will propose a solution that consists in equipping the mobile systems with a high degree of local intelligence in order for them to autonomously handle the uncertainty in the real world and also the arbitrary network delay.

4. Proposed system architecture

With the rapid growth of the Internet there are many communications technologies available to execute requests in a networked environment. Currently the most widely used web browser is the Hypertext Transfer Protocol (HTTP). It can be executed with the Communication Gateway Interface (CGI) for remote control, which is one of methods used in many web-based telerobot systems (Kosuge & al., 1998). Through the Hyper Text Markup Language (HTML) form, a request can be passed from client to server to launch a process to perform some predetermined actions in the server. A dynamically generated HTML page will return the results to the client. But CGI has a number of limitations (Hu & al.) such as its slow response speed. Moreover, a complete HTML page must be generated with each request while the resulting page is still static. So it is not suitable for real-time remote control. In contrast, Java provides the capability to implement network connections and thus avoids the limitations of CGI. A Java applet can operate within the browser and hence is accessible by most computers on the Internet (Le parc & al., 2005). Rather than being static, a Java applet also enables an interface to dynamically change its content due to the fact that the Java applet is an executable within a web page.

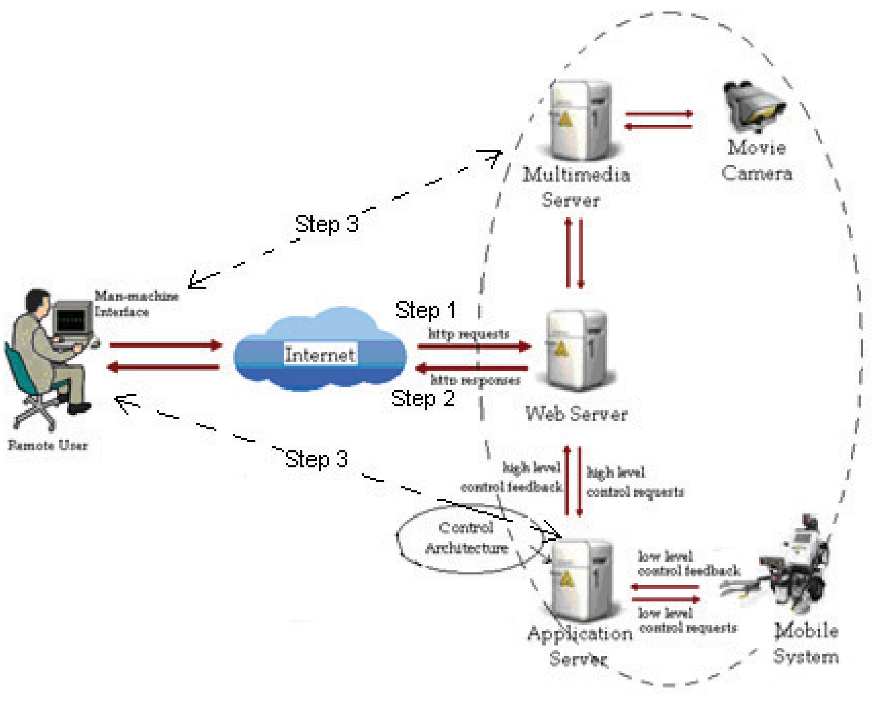


Fig. 7. Proposed System Architecture

From the study of the experiences made on Internet, a common frame (see Figure 7) can be described about the operational aspects of a remote control application (Sayouti & al., 2010).

The remote user, through his Internet navigator, addresses a request to a Web server (step 1) and downloads an application on his work station such as for example an applet Java (step 2). A connection is then established towards the server in charge of the management of the robot to control (step 3). The user is then able to take the remote control of it. In parallel to step 3, other connections are also established towards multi-media servers broadcasting signals (video, sound) of the system to be controlled.

5. Brief overviews of control architecture

One of the first authors who expressed the need for a control architecture was R.A. Brooks (Brooks, 1986). In 1986, he presented an architecture for autonomous robots called "subsumption architecture". It was made up of various levels which fulfil separately precise function, processing data from sensors in order to control the actuators with a notion of priority. It is a reactive architecture in the sense that there is a direct link between the sensors and the actuators (see Figure 8). This architecture has the advantage to be simple and thus easy to implement, nevertheless, the priorities given between the different actions to perform are fixed in time and do not allow an important flexibility.

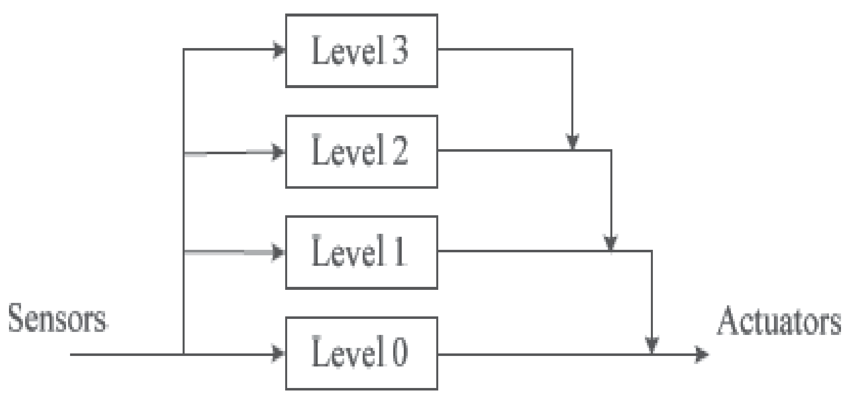


Fig. 8. Subsumption Architecture

Then other various architectures were developed based on different approaches, generally conditioned by the specific robot application that the architecture had to control.
The architecture 4-D/RCS developed by the Army Research Laboratory (Albus, 2002) has the main characteristic to be made up of multiple calculative nodes called RCS (Real time Control System). Each node contains four elements, performing the four following functionalities: Sensory Processing, World Modeling, Behavior Generation and Value Judgment. Some nodes contribute to the perception, others contribute to the planning and control. These nodes are structured in levels, in which one can find the influence of the reactive behaviors in the lower levels and of the deliberative behaviors in the higher levels.
The Jet Propulsion Laboratory developed in collaboration with NASA its own control architecture called CLARAty (Volpe, 2000). Its principal characteristic is to free itself from the traditional diagram on three levels (Functional, Executive, Path-Planner) and to develop a solution with only two levels which represent the functional level and the decisional level. A specific axis integrates the concept of granularity of the architecture for compensating the difficulties of understanding due to the reduction of the number of levels (see Figure 9). One

of the interests of this representation is to work at the decisional level only on one model emanating from the functional level. The decomposition in objects of this functional level is described by UML formalism (Unified Modeling Language) that allows an easier realization of the decisional level.

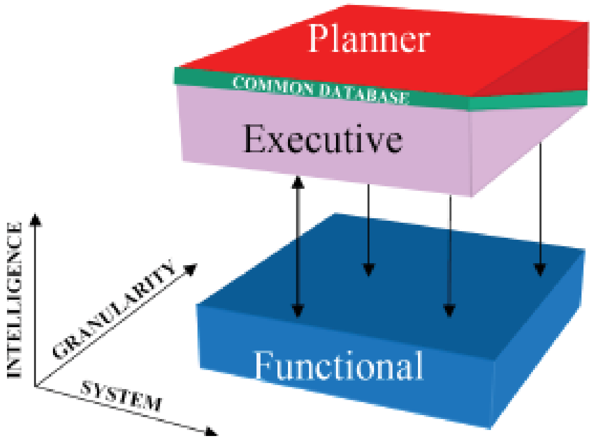


Fig. 9. Two level architecture

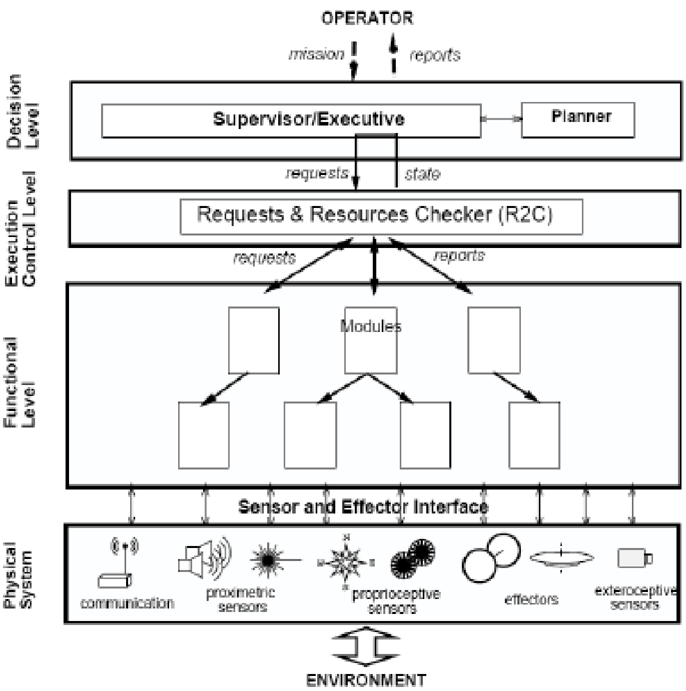


Fig. 10. LAAS Architecture

The LAAS architecture (Laas Architecture for Autonomous System) (Alami, 1998) is made up of three levels: decisional, executive and functional (see Figure 10). Its goal is to

homogenize the whole mobile robotics developments and to be able to re-use already designed modules. All the modules of the functional level are encapsulated in a module automatically generated by GenoM. These have to interact directly with the actuators and other modules of the functional level. The higher level is a controller of execution (Request & Ressources Checker). Its main function is to manage the various requests emitted by the functional level or the decisional level. The operator acts only at the decisional level by emitting missions which depend on the information incoming from the lower levels. This architecture has an important modularity even if the final behavior is related to the programming of the controller of execution.
A. Dalgalarrondo (Dalgalarrondo, 2001) from the DGA/CTA proposed another architecture. It presents a hybrid control architecture including four modules: perception, action, attention manager and behavior selector (see Figure 11). It is based on sensor based behaviors chosen by a behavior selector. The “perception” module carries out models using processing which are activated or inhibited by the “attention manager” module. The “action” module consists of a set of behaviors controlling the robot effectors. A loop is carried out with the information collected by the perception part. This is particularly necessary for low level actions. The “attention manager” module is the organizer of the control architecture: it checks the validity of the models, the occurrence of new facts in the environment, the various processing in progress and finally the use of the processing resources. The “behaviour selector” module must choose the robot behavior according to all information available and necessary to this choice: the fixed goal, the action in progress, representations available as well as the temporal validity of information.

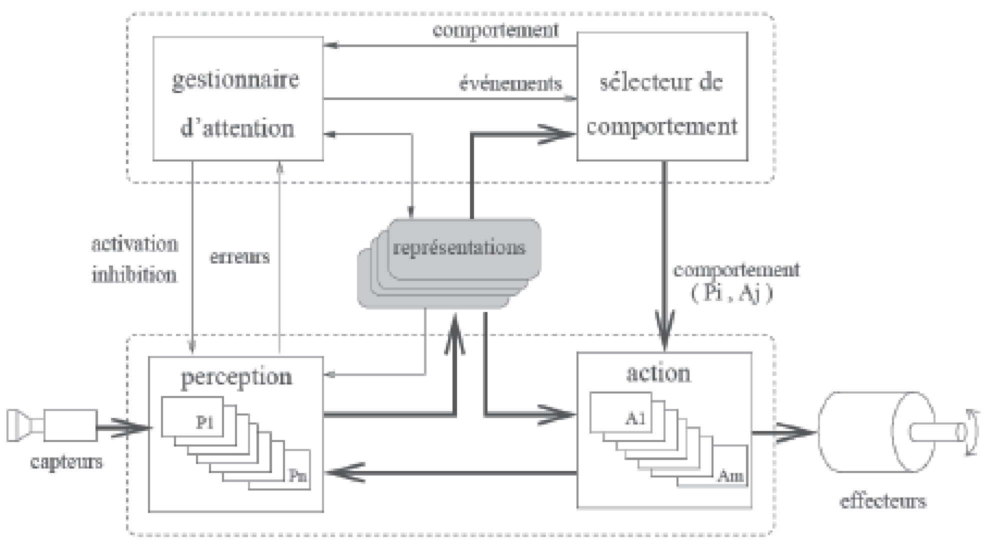


Fig. 11. DGA Architecture

The DAMN architecture (Distributed Architecture for Mobile Navigation) results from work undertaken at the Carnegie Mellon University (see Figure 12). Its development was a response to navigation problems.

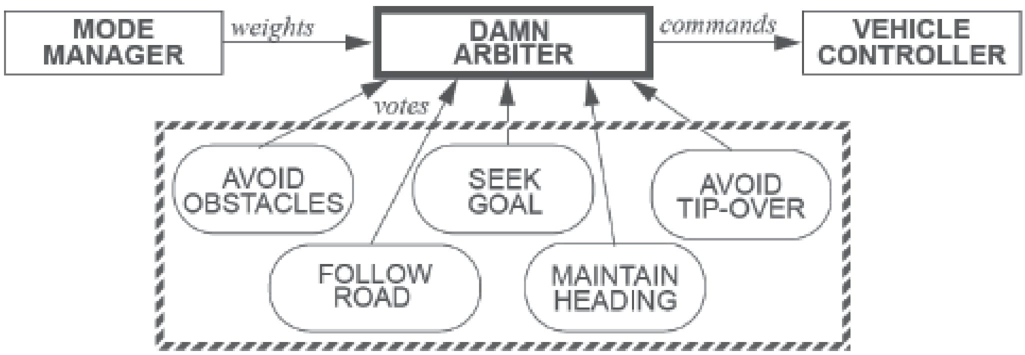


Fig. 12. DAMN Architecture

The principle of DAMN architecture is as follows: multiple modules share simultaneously the robot control by sending votes which are combined according to a system of weight attribution. Then, the architecture makes a choice of controls to send the robot, by a fusion of the possible solutions (Rosenblatt, 1995).

6. Proposed control architecture

Our architecture is based on the same architectures principles that have been presented in the previous section. It relies on the concept of levels initially developed by R. Brooks and which appear in architectures proposed by AuRA or LAAS. We propose a hybrid control architecture which combines aspects of classic control and behavior-based control. Our architecture called EAAS for EAS Architecture for Autonomous system (Sayouti & al., 2008b), including a deliberative part (Actions Selection Agent) and a reactive part. It is made up of two parts, each using distinct method to solve problems (see Figure 13). The deliberative part which uses methods of artificial intelligence contains a path planner, a navigator and a pilot. The reactive part is based on direct link between the sensors (Perception Agent) and the effectors (Action Agent).

Fundamental capacities of our architecture encompass autonomy, intelligence, modularity, encapsulation, scability and parallel execution. To fulfil these requirements, we decided to use a multi agents formalism that fits naturally our needs.

The Multi-Agent System paradigm is one of the most promising approaches to create autonomous, open and dynamic systems, where heterogeneous entities are naturally represented as interacting autonomous agents, which can enter or leave the system at will. Interaction among autonomous agents is fundamental to the dynamic of multi-agent systems. Agents need to interact and coordinate their activity to carry out their common global goal. The development of Multi agents systems is mainly due to its interactions with different scientific domains, in particular with biology. Biology, and especially ethology, inspired the first architecture and several distributed algorithms (Drogoul & Ferber, 1992)(Ferber, 1999).

The communication between agents in our architecture is realized by messages. Object oriented language is therefore absolutely suited for programming agents (we chose java). We use threads to obtain parallelism (each agent is represented by a thread in the overall process).

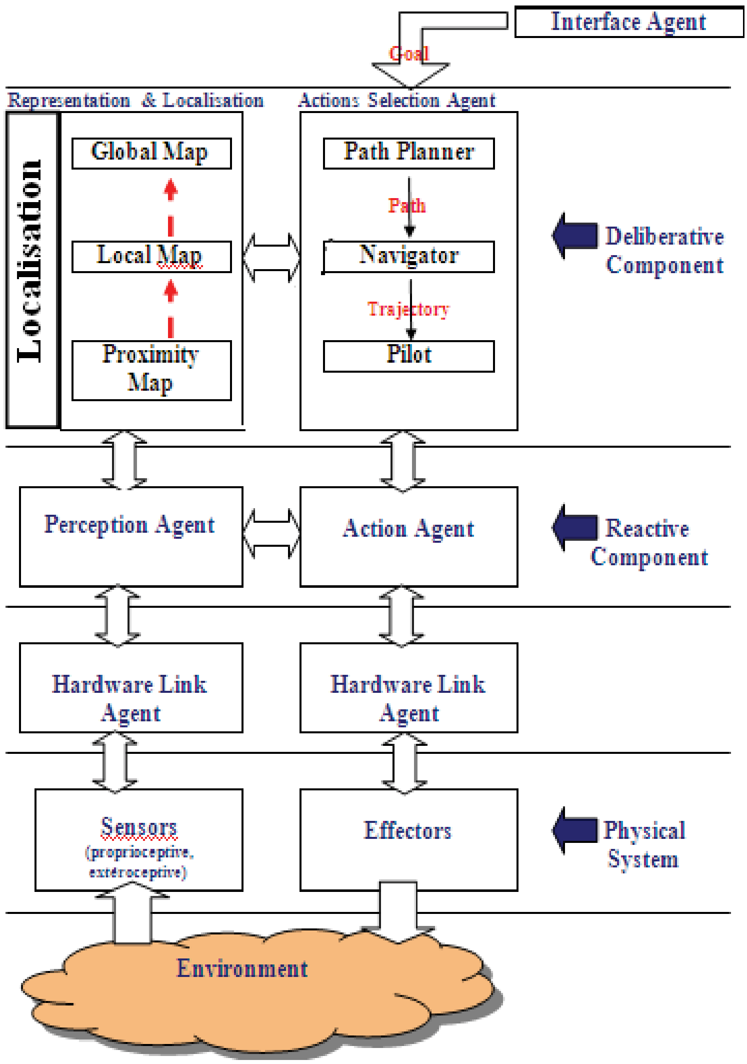


Fig. 13. EAAS Architecture

EAAS architecture consists in five agents: interface agent, actions selection agent, perception agent, action agent and hardware link agent. The interface agent is the high level of our control architecture. It must generate a succession of goal, or missions for the actions selection agent, according to the general mission of the robot. It is the “ultimate” robot autonomy concept: the robot generates itself its own attitudes and its own actions by using its own decisions. The perception agent manages the processing of incoming data (the sensor measurements) and creates representations of the environment. The actions selection agent must choose the robot behavior according to all information available and necessary to this choice: the fixed goal, representations and the robot localization. The actions selection agent contains a path planner, a navigator and a pilot. The path planner may take a goal as input and give a path for achieving the goal as output. The Navigator must translate a path into a trajectory for the pilot. The path does not take into account physical constraints of the robot, but the trajectory that it delivers must integrate them. The function of the pilot is to convert this trajectory into orders to be performed by the action agent. The action agent consists of a set of behaviors controlling the robot effectors. The hardware link agent is an interface between the software architecture and real robot. Changing the real robot require the use of a specific agent but no change in the overall architecture.

7. Remote control software architecture

A Software architecture has been defined to make remote control of mobile systems possible (Sayouti & al. 2008a) (Sayouti, 2009). Our software architecture is based on a set of independent agents running in parallel.

On the left side of figure 14, the server side is represented. It is basically composed of three main agents: "Connection Manager" which manages the different connected clients according to a Control Algorithm. This one is chosen by the designer of the system depending on the application: master/slave, priority, timeout... The "Media" agent communicates with the camera in order to broadcast signals (video, images) of the mobile system in its environment. The "SMA EAAS" (EAS Architecture for Autonomous Systems) which represents our control architecture. EAAS architecture is a hybrid control architecture including a deliberative part (Actions Selection Agent) and a reactive part. The reactive part is based on direct link between the sensors (Perception Agent) and the effectors (Action Agent).

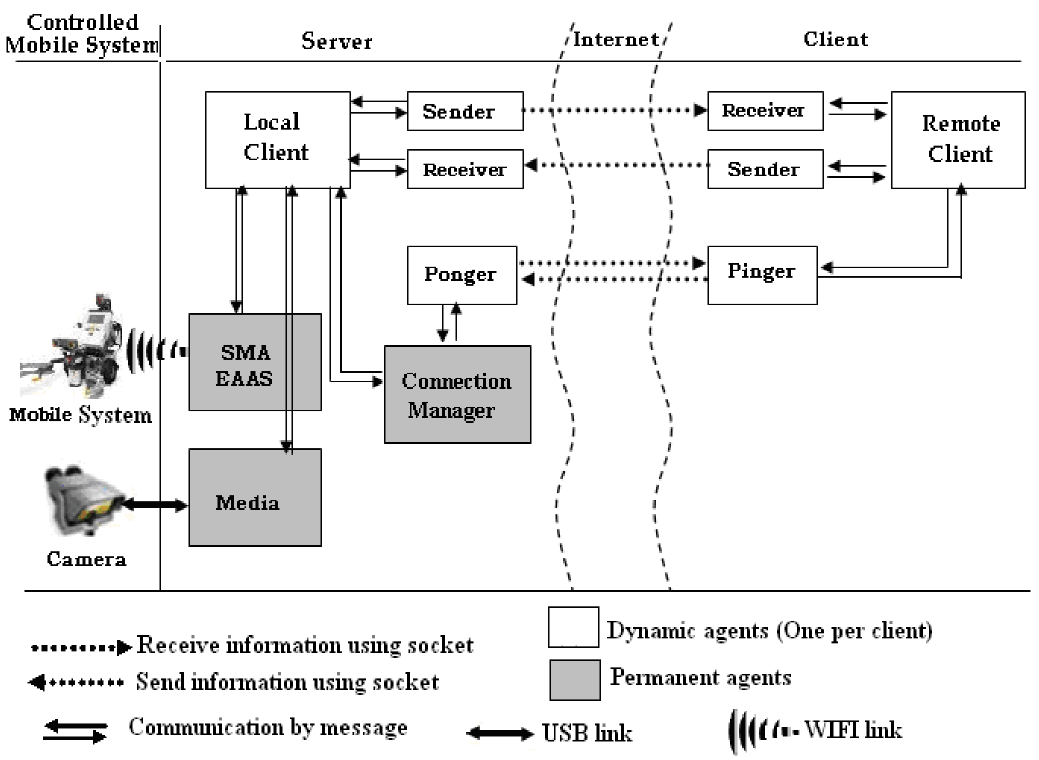


Fig. 14. Remote control software architecture proposed

The right side of the figure represents the client side. Agents are loaded in a web navigator. The “Remote Client” corresponds to a graphical user interface which allows the user to send orders to the mobile system and receive information about the environment. “Sender” and “Receiver” agents are used to allow the communication between the client and the server. “Pinger” and “Ponger” agents are used to observe dynamically the network. If the connexion is accepted, the “Connection Manager” will inform the “Local Client” agent which achieves the interface with the “SMA EAAS” to transmit orders transmission to the mobile system.

8. Applications

Different applications have been developed, in our laboratory, using concepts explained in the previous sections (Sayouti & al, 2009) .
The mobile system used in our realizations is a Lego robot. Lego Mindstorms (LEGO Company, 1997) is a development kit for manufacturing a robot using Lego blocks, and is gaining widespread acceptance in the field of technical education. By using a Mindstorm, a robot can be manufactured for various purposes and functions. It is beginning to be considered as a component of experimental equipment in robotics research.
The Lego mobile robot (see Figure 15) is powered by three reversible motors coupled to wheels and equipped with four sensors: sonar sensor, sound sensor, light sensor and touch sensor. The data produced by these sensors are used by perception agent to build a global map of the Lego robot environment’s. This global map, the goal and the Lego robot localisation are used by the actions selection agent to define a plan of actions to achieve its mission. The Lego robot is equipped with Bluetooth connection that permits the communication with the application server and facilitates its displacement in the environment in order to reach its objective.

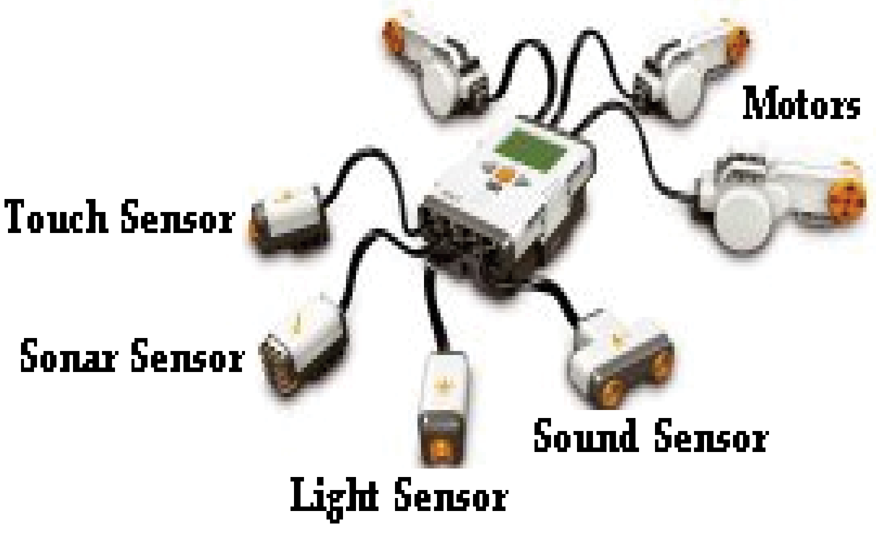


Fig. 15. General Architecture

During the project development, different configurations were tested in different environments. The aim is to develop a more reliable system architecture that can be used in the real world.

Fig. 16. Lego robot in its environment

Our Lego mobile robot was controlled to explore the laboratory while avoiding several static obstacles. In another test, the Lego mobile robot was controlled to push a ball to the goal.
The Web interface of our application (see Figure 17) is designed with the intention of making the remote control easy for researchers and students in order to interact with the Lego mobile robot. A simple interface is designed to provide as much information as possible for remote control. This user interface consists of several Java Applets as shown in Figure 8. It can work on any web browser.
The link between client(s) and server is based on two standard Internet protocols. First, the communication is based on the HTTP protocol, through a home page linking the applets used to move the mobile system. These applets are interpreted by the JVM of the browser. Then, the applets downloaded on the client communicate with the application server through the TCP/IP protocol.
The start button permits to start our application. The remote operator is invited to test the connection using the statistical or the dynamical way, before or during taking the control by clicking on the buttons labelled statistic or dynamic test respectively.
The remote users can pass online tests of knowledge in order to follow a formation answering to their needs. Different modules of formation (Cursus link) are available on our web site, to know: remote control, multi-agents systems, systems architecture, control architecture and autonomous mobile system. We have also set to the remote users a discussion forum within our application to interchange their ideas over the remote control subject.

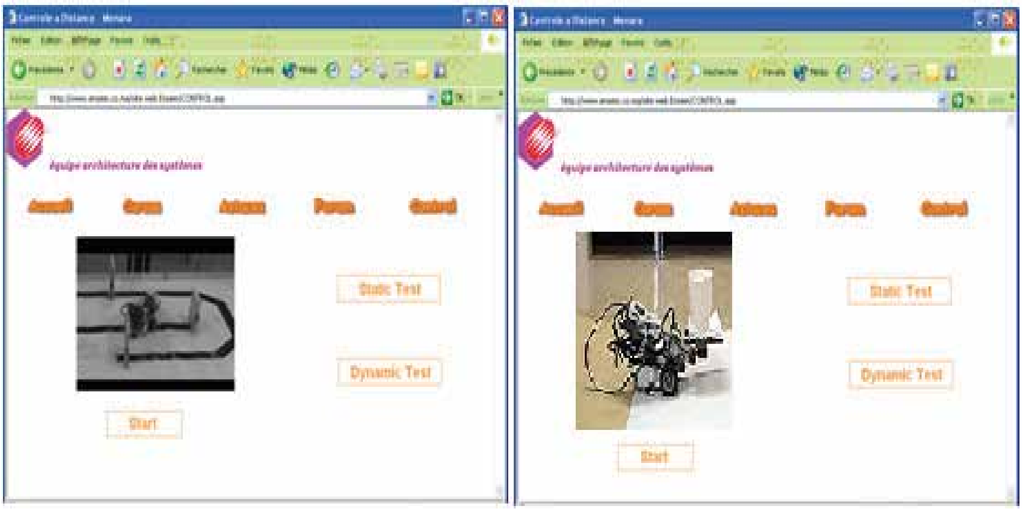


Fig. 17. Web Interfaces

With this simple web interface, one user can control the mobile robot from the web browser with the visual feed back. The other users only have the visual feedback at the same time, and have to wait in queue until the first user logout at this stage.

9. Conclusion

In this chapter, we have presented a Web-based remote control application so that Internet users, especially researchers and students, can control the mobile robot to explore a dynamic environment remotely from their home and share this unique robotic system with us. In the first part, an analysis of the existing control architectures and the approaches for their development has guided us to design a hybrid control architecture. It is called EAAS for EAS Architecture for Autonomous System. The proposed generic architecture consists in associating a deliberative approach for the high level and a reactive approach for the low level. The deliberative level allows decision-making and actions planning thanks to the use of the agent selection of actions. The reactive level, based on couples of agents perception / action, allows the mobile system to react facing the unforeseen events. Then, the software implementation of our architecture was presented. It is achieved under the shape of a multi-agents system by reason of its autonomy, intelligence, flexibility and the various possibilities of evolution. In the second part, in order to validate the choice of our architecture, we presented two applications achieved by the system architecture team of the ENSEM.

The long-term goal of our research is towards real-world applications such as tele-teaching, tele-maintenance, tele-expertise and tele- production.

In future work we hope to increase the intelligence of the agents in our control architecture to provide a telerobot with a high degree of local intelligence to handle restricted bandwidth and transmission delay of the network and to integrate multiple mobile robots into a telerobotics system to achieve redundancy and robustness. This will pave the way for the remote exploration of an unknown and complex environment through the Internet.

10. References

Alami, R., Chatila, R., Fleury, S. Ghallab, M. & Ingrand, F. (1998). An architecture for autonomy. *The International Journal of Robotics Research, Special Issue on Integrated Architectures for Robot Control and Programming*, vol. 17, No 4, pp. 315-337

Albus, J. S. (2002). 4D/RCS A Reference Model Architecture for Intelligent Unmanned Ground Vehicules, *Proceedings of the SPIE 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, 2002, Orlando

Brooks, R.A. (1986). A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation*, vol. 2, n. 1, pp. 14–23

Dalgalarrondo, A. (2001). Intégration de la fonction perception dans une architecture de contrôle de robot mobile autonome, *Ph.D. Thesis, Université Paris-Sud*, 2001, Orsay, France

Drogoul, A., Ferber, J. (1992). Multi-agent simulation as tool for modelling societies: Application to social differentiation in ant colonies, *Proceedings of the MAAMAW'92*, 1992, Viterbo

Ferber, J. (1999). Multi-agents systems, an introduction to distributed artificial intelligence *Addison-Wesley*

Goldberg, K., Siegwart, R. (2001) Beyond Webcams: an introduction to online robots, *The MIT Press*

Goldberg, K., Gentner, S., Sutter, C., Wiegley, J. (2000). The mercury project: a feasibility study for internet robotics, *IEEE Robotics & Automantion Magazine*, pp. 35-40

Hu, H. YU, L. Tsui, P. Zhou, Internet-based Robotic Systems for Teleoperation, *International Journal of Assembly Automation*, Vol. 21, No. 2.

Kosuge, K., Kikuchi, J., Takeo, K. (1998). A Teleoperation system via computer Network, *Proceedings of the IROS'98: Workshop on Web Robots*, 1998, pp. 61-66, Victoria, Canada

Le Parc, P., Vareille, J. & Marce, L. (2004). E-productique ou contrôle et supervision distante de systèmes mécaniques sur l'internet, *Journal européen des systèmes automatisés (JESA)*, Vol.38, No. 5, p. 525-558.

Le Parc, P., Vareille, J., Marcé, L. (2005). Long distance remote control over Internet: a reliability challenge, *Proceedings of the XVI Workshop on Supervising and Diagnostic of Machining Systems*., 2005, Karpacs, Poland

LEGO Company, *Lego mindstorms official page*, http://mindstorms.lego.com, 1997.

Novales, C., Mourioux, G. & Poisson, G. (2006). A multi-level architecture controlling robots from autonomy to teleoperation. *First National Workshop on Control Architectures of Robots*, April 6, 7, 2006, Montpellier

Rosenblatt, J. (1995) DAMN: A Distributed Architecture for Mobile Navigation, *Proceedings of the 1995 AAAI Spring Symposium on Lessons Learned from Implemented Software Architectures for Physical Agents,* AAAI Press, Menlo Park, CA

Saucy, P., & F. Mondada (2000). Khepontheweb: Open access to a mobile robot on the internet, *IEEE robotics and automation magazine*, pp. 41-47, March 2000.

Sayouti, A. (2009). Conception et Réalisation d'une Architecture de Contrôle à Distance Via Internet à Base des Systèmes Multi-Agents. Phd. Thesis, ENSEM, Hassan II University, 2009.

Sayouti, A., Lebbat, A., Medromi, H. & Qrichi Aniba, F. (2010). Remote Laboratory for Teaching Mobile Systems". *International Journal of Computer and Network Security (IJCNS)*, Vol. 2, No. 3, pp. 28-32

Sayouti, A., Qrichi Aniba, F. & Medrom, H. (2008a). Remote Control Architecture over Internet Based on Multi agents Systems. *International Review on Computers and Software (I.RE.CO.S)*, Vol 3, No. 6, pp. 666 – 671

Sayouti, A., Qrichi Aniba, F., Medrom, H. & Radoui, M. (2008b). Tele-robotic over Internet Based on Multi-agents System, *Proceedings of the Seventh International Conference on Distributed Computing and Applications for Business, Engineering and Sciences (DCABES)*, 2008, Dalian, China

Sayouti, A., Qrichi Aniba, F., Medrom, H. (2009). Modeling Autonomous Mobile System with an Agent Oriented Approach". *International Journal of Computer Science and Network Security (IJCSNS)*, Vol. 9, No. 9, pp. 316 – 321

Schulz, D., Burgard, W., Cremers, A. B. (1998). Predictive simulation of autonomous robots for tele-operation systems using the world wide web, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System*, 1998, Victoria, B.C.

Simmons, R., Xavier. (1998). An autonomous mobile robot on the web, *Proceedings of the International Workshop On Intelligent Robots and Systems (IROS)*, 1998, Victoria, Canada

Stein, M.R., Painting on the world wide web: the pumapaint project, Proceedings of the IEEE IROS'98 Workshop on Robots on the Web, 1998, Victoria, Canada

Taylor, K., Dalton, B. (1997). Issues in internet telerobotics, *Proceedings of the FSR'97 International Conference on Field and Service Robots*.

Volpe, R. (2000). CLARAty: Coupled Layer Architecture for Robotic Autonomy (*JPL Technical Report D-19975*

23

Multi-agent Systems for Industrial Applications: Design, Development, and Challenges

Atalla F. Sayda
University of New Brunswick
Canada

1. Introduction

Complex industrial facilities are characterized by their large size, complexity of the constituent sub-processes and their dynamics, and the massive information overload. For example, the maintenance and management of complex process equipment and processes, and their integrated operation, play a crucial role in ensuring the safety of plant personnel and the environment as well as the timely delivery of quality products. Given the size, scope and complexity of the systems and interactions, it is becoming difficult for plant personnel to anticipate, diagnose and control serious abnormal events in a timely manner (Venkatasubramanian et al., 2003).

This lays huge challenges on the design, development and deployment of a system to manage such facilities in different operating conditions. Multi-agent systems (MAS) and artificial intelligence (AI) are the corner-stone frameworks to design an intelligent real-time system for complex industrial facility management and control. This chapter addresses the different aspects of designing a multi agent system to manage and control complex industrial facilities from conceptualization to deployment. A thorough literature survey is done to give a big picture of what has been accomplished in academia and industry for this topic primarily. The design of an industrial MAS system will be discussed from conceptualization to validation in the following sections.

2. Challenges in developing industrial multi-agent systems

Although industrial multi-agent systems have great impact on complex process plants in terms of higher profitability and better management, the development of such systems is very difficult and exhibits many challenges (Mark et al., 1995; Mylaraswamy, 1996; Mylaraswamy & Venkatasubramanian, 1997; Vedam, 1999; Vedam et al., 1999; Venkatasubramanian, 2005; Venkatasubramanian et al., 2003):

- Diversity of solution techniques, where several approaches are available to perform the main tasks of an industrial facility management system: For example, fault detection and isolation can be performed using model-based quantitative and qualitative fault diagnosis techniques as well as non-model-based methods. Similarly, supervisory control can be performed using several AI techniques such as rule-based expert systems and case-based reasoning. These techniques are diverse in nature and use certain assumptions about the process and performance requirements. Hence, determining the best approaches for performing the individual tasks of industrial facility management is difficult. Moreover, the chosen approaches may not meet the goals of the overall system. Having these techniques integrated in one intelligent system may seem the only solution. However, the analysis of the accuracy, consistency and stability of such integrated systems is even more difficult.

- Diverse sources of knowledge, which stems from the incomplete and scattered nature of process knowledge such as process manuals, operational expertise, process models, and historical data: Techniques to integrate the knowledge sources into a form that can be used effectively in an intelligent system are of a great necessity. The ontology based knowledge organization approach is an example of such techniques.
- Uncertainty in process models and measurements, which may affect the performance of a complex industrial facility system: Most of the system's tasks depend on accurate process measurement and models. Noisy sensor measurements, process disturbances, and the highly non-linear dynamics of chemical processes in general can be a significant source of the failure of the entire system. Systematic analysis of such uncertainties and their effect on the system performance is required.
- Widely varying time scales of the different system tasks and the operating situations which may happen in the plant: Some operating situations might develop over a few minutes, while others might develop over hours and days. Likewise some tasks of the system might execute in few milli-seconds, while other tasks may take minutes and hours to make decisions such as the supervisory control task.
- Implementation for large scale industrial plants, which has effect on the system software architecture, real-time hardware, field testing and validation, user interface and operator training and acceptance.

3. Survey of industrial multi-agent systems (MAS)

The automation of industrial facility management has been recognized by academia and industry as a vital research area, which many research programs and industrial projects were initiated to investigate. Some projects focused on managing the process during normal operation while others gave abnormal situation management a higher priority. In the late 1980's, the European Commission funded a major research project called ARCHON, which was focused on the problem of getting a number of distinct expert systems to pool their expertise in solving problems and diagnosing faults in several industrial domains. ARCHON was recognized as one of the first real industrial applications of MAS (Jennings & Mamdani, 1996). In this section, an indepth survey of research and industrial projects will be presented. The early projects did not exploit the MAS approach, yet they gave an insight for later research projects, which benefited from the MAS approach. Here eight of the most relevant projects are reviewed.

3.1 The FORMENTOR research project

FORMENTOR, which is a joint venture of major European companies, is supported by the EUREKA program of cooperative international R&D projects with a budget of 33.4 million Euros. The research program, which lasted from 1986 to 1996, "aimed to develop real-time plant supervision software systems to support operators in their decision-making process by enabling them to make effective use of all this information, and avoid disturbances and any loss of production" (Kim & Modarres, 1987; Wilikens & Burton, 1996). The main technical features of the system are (Wilikens & Burton, 1996):

1. A goal tree-success tree (GTST), which is a representation of the functional model of the process.
2. The multi-layer model (MLM) used to represent the functional components of the plant in a hierarchical way to provide a global overview of the plant state and to guide the action planning process.
3. Two distinct but complementary reasoning modules for diagnosis, both based on this multi-layer structure.

In FORMENTOR the object oriented approach was used to implement the system, which in turn consisted of a collection of modules to perform the different tasks, a communication system, and a global controller to control the activities of the modules. The disadvantages of such a system can be summarized as follows:

1. The use of simple mathematical formulas to represent the behavioral models of the plant components, where the complex interactions among the plant variables are not captured.
2. The absence of a model identification module to address the dynamic and varying nature of chemical plants.
3. Fault diagnosis is largely qualitative, which leads to lack of resolution.
4. Fault mitigation is also qualitative and based on the current state of the system, making the system only reactive.

3.2 Advanced process analysis and control system (APACS)

The APACS project was designed to introduce intelligent agents into a modern nuclear operating environment. APACS was a PRECARN project supported by the Canadian government and several Canadian universities and companies, and developed by a team of more than 10 software designers and engineers. The 9.7 million-Canadian-dollars, five-year project began in the fall of 1990 and was completed in the fall of 1995 (Mylopoulos et al., 1992; Wang & Wang, 1997). "The goal of the APACS project was to develop a generic framework for building an intelligent system that assists human operators of power plants in noticing and diagnosing failures in continuous processes" (Wang & Wang, 1996).

The APACS system consists of three layers: the agent layer which implements the system functionality; the knowledge broker layer which manages communication between the agents; and the information repository layer which stores the system common knowledge. The agents of the APACS system perform the following tasks (Wang & Wang, 1997):

1. The data acquisition agent receives data from the plant's main control computer.
2. The tracking agent continuously updates the data links between the agent system and the actual plant sensor positions.
3. The monitoring agent analyzes the feedwater sensor values and feedwater alarms and then produces symbolic descriptions of the plant's behavior.
4. The human-computer interface (HCI) agent displays the APACS status to the plant operators and serves as the user interface.
5. The diagnostic agent takes the output from the monitoring agent and attempts to generate a qualitative causal explanation that will eventually be useful to the human operators.

6. The verification agent operates a faster-than-realtime numerical, model-driven simulator to measure the correlation of the diagnostic agent's output against the simulator's ideal values.

The entire APACS system was implemented in C++ using Expersoft's XShell (an extended C++ syntax for declaring distributed objects) as its communication environment and CLIPS (a rules-based inferencing environment constructed by NASA) (Wang & Wang, 1997). The APACS project had some of the FORMENTOR project disadvantages such as the absence of a model identification agent, the qualitative nature of the fault diagnosis task, and the absence of a fault mitigation (i.e., accommodation) agent.

3.3 The pilot's associate (PA) program

The first research program to address the intelligent facility management problem in the US was the Pilot's Associate (PA) program, which is a joint effort of the Defense Advanced Research Projects Agency and the US Air Force, managed by the Air Force's Wright Laboratory. The program began in February 1986 as an application demonstration for DARPA's Strategic Computing Initiative. A primary goal of the PA program was to enhance combat fighter pilot effectiveness by increasing situational awareness and decreasing their workload. DARPA wanted to advance the program's technology base, principally in the area of real-time, cooperating knowledge-based systems. The Air Force wanted to explore the potential of intelligent systems applications to improve the effectiveness and survivability of post-1995 fighter aircraft (Banks & Lizza, 1991; Small & Howard, 1991).
"The Pilot's Associate concept developed as a set of cooperating, knowledge-based subsystems: two assessor and two planning subsystems, and a pilot interface. The two assessors, Situation Assessment and System Status, determine the state of the outside world and the aircraft systems, respectively. The two planners, Tactics Planner and Mission Planner, react to the dynamic environment by responding to immediate threats and their effects on the pre-briefed mission plan. The Pilot-Vehicle Interface subsystem provides the critical connection between the pilot and the rest of the system" (Banks & Lizza, 1991). Another project, which followed the PA program to address the facility management problem in attack helicopters, is the Rotorcraft Pilot's Associate (RPA) program. The goal of the US Army funded RPA program was to develop and demonstrate in flight an advanced, intelligent "associat" system in a next-generation attack/scout helicopter (Miller & Hannen, 1999).

3.4 Abnormal situation management (ASM)

The PA and RPA projects paved the way for other projects to develop and automate the industrial facility management process for the process industry in the United States. AEGIS (Abnormal Event Guidance and Information System), which was developed by the Honeywell led Abnormal Situation Management (ASM) Consortium in the United States, is a very important project (Cochran et al., 1997). The AEGIS project proposes a comprehensive facility management framework from an industrial view point. AEGIS built on the experience of military aviation research projects, especially the Pilot's Associate (PA) and the Rotorcraft Pilot's Associate (RPA) (Cochran et al., 1996). It is really worth considering the project and its current status, since it is supported by major oil and gas companies allied with Honeywell and other automation industry key leaders. Furthermore, it is considered a research imperative to learn from it, in terms of experience, stages being successfully accomplished, limitations, and failures incurred during the course of the project. The research program life span started from 1994 and ended in 2008, where the program was funded by the National Institute of

Standards and Technology (NIST). The program focused on the development of a proof of concept system called AEGIS (Abnormal Event Guidance and Information System), which have gone through different development stages.

3.4.1 Hybrid distributed multiple expert framework (DKIT)

The diagnostic toolkit (DKIT) project was initiated as the first step in the design and development of the AEGIS system. The DKIT hybrid framework addressed the use and integration of multiple fault diagnosis techniques to meet the challenges of complex, industrial-scale diagnostic problems (Mylaraswamy, 1996; Mylaraswamy & Venkatasubramanian, 1997). The principle of DKIT is black-board collective problem solving, in which several modules are integrated (Mylaraswamy, 1996):

- Diagnostic experts: a collection of one or more fault diagnostic modules including a signed directed graph (SDG) technique, qualitative trend analysis (QTA), and probability density function based statistical classifier.
- A blackboard: a placeholder for various process states. This is implemented as pigeon holes, each of which corresponds to a well defined process state.
- A scheduler, which consists of a monitor that keeps track of new events or states that are posted on the blackboard; a switchboard which directs the information to relevant subscribers, and a mechanism for conflict resolution between the different diagnostic modules.
- A plant Input-Output Interface, which acts as a channel for all diagnostic modules to receive relevant process measurements.
- An operator interface for presenting diagnostic results to the operator.
- A process equipment library to represent the external process.

The DKIT system was fully implemented in the G2 expert system shell, and was validated on a simulation model of fluid catalyst cracking unit (FCCU). The DKIT framework demonstrated the feasibility of a complex fault diagnosis system, and was further enhanced through the development of the OP-AIDE system, which will be discussed in the next section.

3.4.2 Integrated operator decision support system (Op-Aide)

To address the qualitative fault analysis of previous projects (i.e., the FORMENTOR and APACS systems), an integrated operator decision support system, called Op-Aide, was developed based on the DKIT system architecture to assist the operator in quantitative diagnosis and assessment of abnormal situations (Vedam, 1999; Vedam et al., 1999). Op-Aide consists of six modules (or knowledge sources) and an Op-Scheduler that coordinates them. It provides the interface between different modules in the system and functions as a centralized data base for all the modules. The results of these modules are posted onto it, where they can be accessed by the other modules in the system (Vedam et al., 1999):

- Data Acquisition Module, which acquires on-line data from the plant and makes them available to other modules.
- Monitoring Module: This module monitors the process data for the presence of abnormalities using a principal component analysis (PCA) model of the process.
- Diagnosis Module, which identifies the root causes for the abnormalities. Multiple diagnosis methods are combined in a blackboard architecture.

- Fault Parameter Magnitude Estimation (FRAME) Module, which estimates the magnitude and rate of change of the root causes.
- Simulation Module, which performs a simulation to predict future values of the process outputs.
- Operator Interface Module, where the status of the process and the results of the different modules are constantly communicated to the operator through this module.

Op-Aide has been implemented using blackboard-based architecture in Gensym's expert system shell G2, MATLAB and C. The Op-Scheduler coordinates the functioning of other modules using event and time driven rules and procedures. The results of these modules are represented as objects that are pushed back onto specified slots in the OP-Scheduler. Most of the modules are implemented in G2 except for the FRAME and simulation modules, which are implemented in MATLAB and C respectively.

Although the OP-Aide project came to address the qualitative fault diagnosis disadvantage in the FORMENTOR and APACS systems by introducing two complementary quantitative fault diagnosis modules, it did not address the dynamic nature of the chemical process by embedding a model ID module. Furthermore, operating the situation assessment, which is achieved through the FRAME and simulation modules, is a semi-automatic process done at the request of the operator. OP-Aide did not address the whole performance aspect when it comes to managing large scale plants.

3.4.3 Abnormal Event Guidance and Information System (AEGIS)

The Honeywell ASM Consortium adopted the Dkit architecture as its AEGIS prototype, a next-generation intelligent control system for operator support (Venkatasubramanian et al., 2003). The AEGIS program successfully demonstrated the feasibility of collaborative decision support technologies in the lab test environment, with a high fidelity simulation model of an industrial manufacturing plant. As far as industrial environment testing is concerned, the focus was on abnormality diagnosis and early warning, and assessing and learning from experience, which resulted in effective operations practices and supporting services.

The AEGIS research program team has achieved several goals and developed a well established abnormal situation management awareness and culture through massive consultation, research, and collaboration with oil and gas industry key leaders. Achievements can be summarized in the following points as presented by the director of advanced development at Honeywell, Mr. A. Ogden-Swift, during the 2005 advanced process control applications for industry workshop (APC 2005) (Ogden-Swift, 2005):

- significant user interface (UI) improvements,
- 35% reduction in alarm flooding by introducing a new alarm reconfiguration philosophy,
- integration of operation procedures,
- equipment monitoring through intelligent sensor integration,
- fuzzy/PCA early error detection, and
- improved operator training.

Such achievements were deployed in the new generation of Honeywell's Experion distributed control system. Although the 12 year old AEGIS research program has resulted in a well defined abnormal situation management problem in terms of best practices, goals, and limitations, it did not address the following points, which aim to minimize the workload on process operators:

- full automation of massive process data interpretation,
- full automation of process fault diagnosis and accommodation,
- incorporation of state of the art fault diagnosis techniques which were developed during the past 25 years of academic research,
- reduced manual system configuration by process operators (for example, the operator has to choose the appropriate dataset for process model identification), and
- intelligent techniques such as expert systems to assist operators in the decision making process.

Only one technique was used for early fault detection, a statistical technique based on principal component analysis (PCA). To enable this, the operator has to manually adapt for operating point change by choosing the appropriate data set.

3.5 Advanced decision support system for chemical/petrochemical manufacturing processes (CHEM-DSS)

Another promising project is CHEM-DSS (Decision Support System for Chemical/Petrochemical Manufacturing Processes), which is an initiative of the European Community (EC) Intelligent Manufacturing Systems consortium in collaboration with Japan and Korea. "The aim of the CHEM-DSS project is to develop and implement an advanced Decision Support System (DSS) for process monitoring, data and event analysis, and operation support in industrial processes, mainly in refining, chemical and petrochemical processes" (Cauvin, 2004b).

The CHEM-DSS research project was initiated to compete and build on the two main initiatives in the United States, namely, the Abnormal Situation Management (ASM) Consortium led by Honeywell, and the Intelligent Control Program of NIST. However there was no clear system architecture that demonstrates the behavior of the integrated modules of the system during the course of the project (1998 - 2004). The research instead focused on analyzing the properties of the individual techniques of the system such as FDI, planning, artificial intelligence, signal processing, and scheduling, and twenty-three software toolboxes were developed during the project (from April 2001 to March 2004) (Matania1, 2005).

The heart of the CHEM-DSS integration platform is G2, which integrates the twenty-three software toolboxes. All developed software tools were integrated to a communication manager (CCOM) based on the message-oriented middleware (MOM). In this project the XMLBlaster open-source MOM was used to manage XML messages between the different tools. The data management and user interface functionalities were implemented in the G2 environment (Matania1, 2005).

Furthermore, "the toolboxes have been tested at pilot plants and industrial sites. It was applied to partner facilities to ensure rapid technology transfer. The industrial end-users provided different kinds of processes including a fluid catalytic cracking pilot plant, a paper making process, a gasifier pilot plant, a steam generator, a blast furnace and distillation process. End users can use the developed toolboxes to design their own intelligent diagnostic system according to their requirements" (Cauvin, 2004a).

3.6 Integrated system health management (ISHM)

The ISHM (Integrated System Health Management) research program, which is developed by NASA for space applications, "focuses on determining the condition (health) of every element in a complex System (detect anomalies, diagnose causes, prognosis of future anomalies), and provide data, information, and knowledge to control systems for safe and effective operation. In the case of NASA, this capability is currently done by large teams of people, primarily from the ground, but needs to be embedded on-board systems to a higher degree to enable NASA's new Exploration Mission (long term travel and stay in space), while increasing safety and decreasing life cycle costs of spacecraft (vehicles; platforms; bases or outposts; and ground test, launch, and processing operations)" (Figueroa, Holland & Schmalzel, 2006). The ISHM research program, whose life span started from 2003 and will end in 2009, was extended to address several applications including military/civilian space and aircraft systems in collaboration with several companies such as Boeing and Honeywell (Derriso, 2005; Figueroa, Holland, Schmalzel & Duncavage, 2006; Garcia-Galan, 2005; Karsai et al., 2005; Maul et al., 2005; Schmalzel et al., 2005).

The ISHM architecture is based on the open systems architecture for condition-based maintenance (OSA-CBM), which is an implementation of the ISO standard # 13374. The ISHM system is deployed as a distributed module system with different functions including anomalies detection, overall systems state identification, anomaly and failure effects mitigation, and systems elements condition evaluation. The ISHM research project supported by NASA used the G2 environment as their intelligent integration framework. In fact, six G2 servers are deployed to monitor International Space Station (ISS) subsystems, including the mechanical, structural, electrical, environmental and computational systems. The G2 servers continually inspect and analyze data transmitted from space during missions (Figueroa, 2005; Maul et al., 2005).

3.7 Distributed architecture for monitoring and diagnosis (DIAMOND)

The DIAMOND project was developed by the University of Karlsruhe in cooperation with three industrial partners and one research institute within the framework of the EU Esprit Program with a budget of one million Euros. The program started in 1999 and ended in 2001, where the program objective was to investigate the feasibility of fault diagnosis system for industrial applications.

The DIAMOND system architecture is a set of distributed cooperating tasks. Each task is associated with a specialized agent, namely the monitoring agent, which is interfaced to the industrial application, a set of diagnostic agents to identify the functional state of the plant, a conflict resolution agent to investigate whether the diagnostic results are contradicting or completing each other, a facilitator agent to manage networking and mediating between different agents, a blackboard agent for storing the diagnoses, and a user interface agent for presenting the results to the operator (Worn, 2004).

The DIAMOND system was implemented using the KQML-CORBA- (knowledge query and manipulation language) based architecture, in which the different agents are implemented as distributed CORBA objects.The system prototype was evaluated while monitoring and diagnosing the water stream cycle chemistry of a coal-fired power plant (Worn, 2004).

3.8 Multi-agent-based diagnostic data acquisition and management in complex systems (MAGIC)

MAGIC is developed by a joint venture of several European universities and companies. The European Commission Information Society of Technology (EC-IST) funded the project with a budget of 3.3 million Euros. The MAGIC research program is a multi-agent system realization of an intelligent fault diagnosis system. "The system aims at developing general purpose architecture and a set of tools to be used for the detection and diagnosis of incipient

or slowly developing faults in complex systems. The early identification of potentially faulty conditions provides the key information for the application of predictive maintenance regimes" (Köppen-Seliger et al., 2003).
The distributed architecture for MAGIC is based on a multi-agents/multi-level concept. The idea is that the task of the complex system's diagnosis and operator support is distributed over a number of intelligent agents which perform their individual tasks nearly autonomously and communicate via the MAGIC architecture. Such an architecture can easily be distributed on existing monitoring and control systems of large scale plants (Garcia-Beltran et al., 2003; Köppen-Seliger et al., 2003). The MAGIC system consists of several model-based and cause-effect diagnostic agents and a process specification agent to specify the process to be monitored and diagnosed. Depending on the process specifications, the appropriate data and knowledge acquisition is performed by another agent. A diagnostic decision agent and a diagnostic support agent propose a final diagnostic decision, which is displayed with other information to an operator interface agent. The MAGIC system prototype is developed for the metal processing industry (Garcia-Beltran et al., 2003; Köppen-Seliger et al., 2003).

3.9 Intelligent control and asset management (ICAM) system

Having shown the current status of asset management research in both academia and industry, we conclude that the AEGIS research program focused on the bookkeeping and human machine interaction tasks rather than a fully automated and functional facility management holistic approach. Furthermore, the CHEM-DSS research program did not give a clear picture of how the different techniques will be integrated, and what software development tools/plans will be used to develop a prototype of the system. A new research program ICAM (Intelligent Control and Asset Management of industrial facilities) was initiated by a joint venture of Atlantic Canadian universities and National Research Council of Canada (NRC) to benefit from the success and limitations of the AEGIS, CHEM-DSS and other projects, to build on their experiences, to complement their developed tasks, and to push the envelope by evaluating and incorporating state of the art of fault diagnosis, artificial intelligence (AI) and wireless sensor networks techniques (Taylor, 2004). This will be embedded in a fully automated system architecture, which will better support process operators and improve operability (Larimore, 2005; Laylabadi & Taylor, 2006; Omana & Taylor, 2005; 2006; 2007; Sayda & Taylor, 2006; 2007a;b;c; 2008a;b;c; Smith et al., 2005; Taylor & Laylabadi, 2006; Taylor & Omana, 2008; Taylor & Sayda, 2005a;b; 2008).
Figure 1 illustrates the ICAM system, to which measured data from the industrial facility are transmitted, and interpreted for better process control and management. ICAM system is composed of a group of servers and operator work stations linked to each other through a high speed Ethernet network. The wireless sensor network is managed by a real time communication server. The database server stores received data in its database after being preprocessed. A group of application servers are the backbone of the ICAM system. The application servers run the tasks of data preprocessing, model identification, fault diagnosis, fault mitigation and accommodation, human machine interaction, and supervisory control. Each server is a computer cluster, which is a group of loosely coupled computers that work together closely to achieve higher performance, availability, and load balance. This will result in better internal coordination among the different ICAM servers. The conceptual model of the ICAM system is discussed in the following sections along with the system requirements analysis.

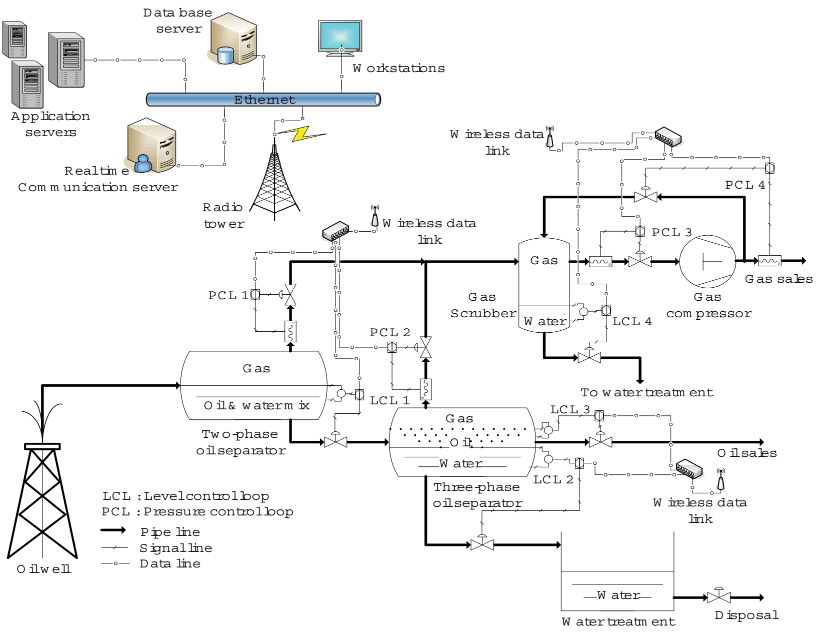


Fig. 1. ICAM Project schematic diagram

4. Conceptual model of the ICAM system

Several conceptual frameworks have been suggested for modeling complex intelligent systems. In the past two decades, the most popular design framework was the expert system, which has several advantages, namely, separation of knowledge and inference, ease of development and transparent reasoning under uncertainty. Moore and Kramer (Moore & Kramer, 1986) discussed the issues of expert system design for real-time process control applications; an intelligent expert system (PICON) was designed and implemented on several process plants to validate expert systems performance in real-time process environments. Implementation results revealed several drawbacks, namely, lack of learning mechanisms, knowledge base validation difficulties, and weak representation power. There are several expert system survey papers to which one may refer for further insight (Liao, 2004; Liebowitz, 1998).

Newell (Newell, 1990) introduced cognitive architectures as a more general conceptual framework for developing complex intelligent systems, based on a human cognition viewpoint. This approach assumes that human cognition behavior has two components, architecture and knowledge. The architecture is composed of cognitive mechanisms that are fixed across tasks, and basically fixed across individuals. These mechanisms, which define the properties of this approach, involve a set of general design considerations, namely, knowledge representation, knowledge organization, knowledge utilization, and knowledge acquisition. Newell argued that these considerations represent theory unification to model complex intelligent systems. Furthermore, this allows model (knowledge) reuse and helps create complete agents opening the way to applications. The performance of several tested cognitive architectures in solving different problems points to a promising future for modeling complex intelligent systems.

Multi-agent systems (MAS) which can be considered as an instantiation of distributed artificial intelligence, is another conceptual framework for modeling complex systems. A MAS is defined as a loosely coupled network of problem solvers that work together to solve problems, that are beyond their individual capabilities (Durfee & Montgomery, 1989). The MAS platform emphasizes distribution, autonomy, interaction (i.e., communication), coordination, and organization of individual agents. Agents in MAS can be defined as conceptual entities that perceive and act in a proactive or reactive manner within an environment where other agents exist and interact with each other based on shared knowledge of communication and representation (Wooldridge, 2002). Each agent contains processes for behavior generation, world modeling, sensory processing, and value judgment together with a knowledge database, as shown in figure 2.

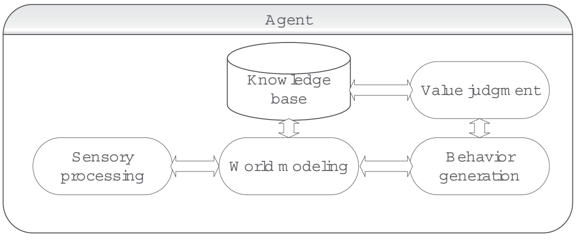


Fig. 2. Agent architecture

Sloman (Sloman & Scheutz, 2002) introduced H-Cogaff, a human-like information processing architecture, which contains many components performing different functions all of which operate concurrently and asynchronously. The H-Cogaff architecture seems to represent a combination of the cognitive architecture and the MAS conceptual frameworks. As illustrated in figure 3, Sloman's architecture provides a framework for describing different kinds of architectures and sub-architectures, and which, to a first approximation, is based on superimposing two sorts of distinctions between components of the architecture: firstly the distinction between perceptual, central and action components, and secondly a distinction between types of components which evolved at different stages and provide increasingly abstract and flexible processing mechanisms within the virtual machine (Sloman, 2001). The reactive components generate goal seeking reactive behavior, whereas the middle layer components enable decision making, planning, and deliberative behavior. The modules of the third layer support monitoring, evaluation, and control of the internal process in the lower layers.

Having reviewed the different conceptual modeling frameworks, it is our opinion that Sloman's H-Cogaff scheme is the best candidate, which would meet most of the requirements of an ICAM system for complex industrial plants. The architecture of the system and its functional modules will be discussed in subsequent sections.

5. ICAM System architecture and behavior model

Figure 4 illustrates the proposed architecture of the conceptual system, which consists of four information processing layers and three vertical subsystems, namely, perception, central processing, and action according to Sloman's H-Cogaff scheme. The lowest horizontal layer above the distributed control system (DCS) contains semi-autonomous agents that represent different levels of data abstraction and information processing mechanisms of the system. The middle two layers (i.e., the reactive and deliberative layers) interact with the external

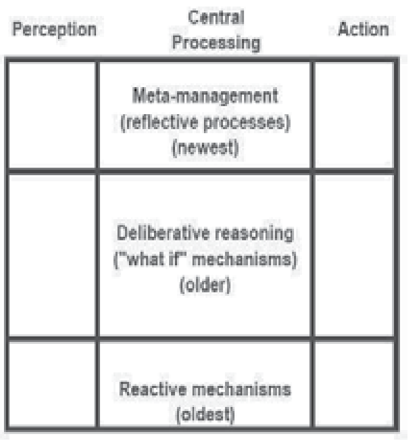


Fig. 3. Human cognition and affect (H-Cogaff) architecture Sloman & Scheutz (2002)

environment via the DCS and thus the industrial process by acquiring perceptual inputs and generating actions. The perceptual and action subsystems are divided into several layers of abstraction to function effectively. This can be achieved, for example, by categorizing observed events at several levels of abstraction, and allowing planning agents to generate behavior (actions) in a hierarchically organized manner.

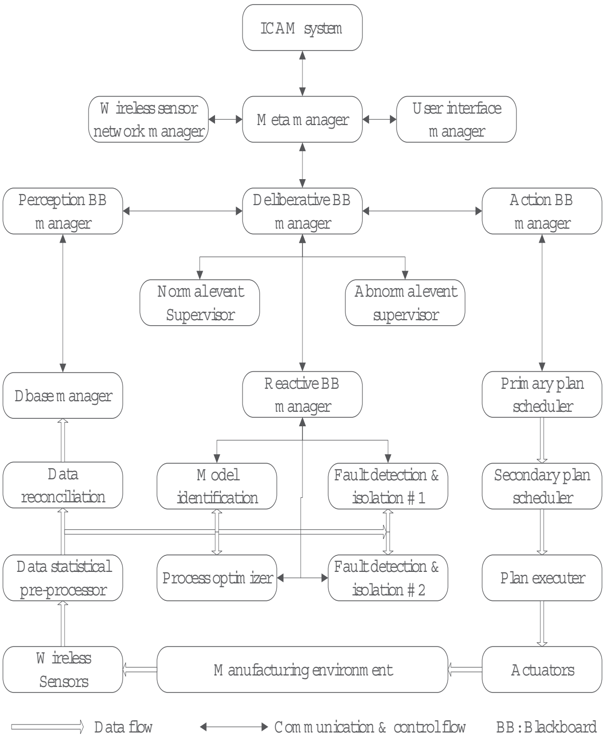


Fig. 4. ICAM system architecture and behavior model

The basic flow of control in the system begins when perceptual input arrives at the lowest level in the architecture. If the reactive layer can deal with this input then it will do so, otherwise, bottom-up activation will occur and control will be passed to the deliberative layer. If the deliberative layer can handle the situation then it will do so, typically by making use of top-down execution of reactive agents. Otherwise, it will pass control to the meta-management layer to resolve any internal conflicts in the architecture or notify the operator that it cannot do so. As illustrated in figure 4, the functionalities of the ICAM system agents in each layer are.

- Statistical pre-processing agent to clean the measured data from undesired discrepancies such as missed data and outliers.
- Data reconciliation agent to reconcile measured data according to mass and energy conservation laws.
- Database agent to store real-time data for historical and other purposes.
- Several fault detection and isolation (FDI) agents to detect any abnormal events using different FDI approaches.
- An optimization agent to generate the optimal operating plans.
- a system identification agent to identify the mathematical model of the industrial process.
- Two intelligent supervisory agents to manage the industrial plant during normal and abnormal situations.
- A meta manager agent to manage and coordinate the different system agents.
- A set of black board agents to facilitate asynchronous communications among the system agents.
- A set of task scheduling agents to execute the required plan according to different time frames.

Rigorous coordination of the behavior of the ICAM system layers and agents is crucial to success. A sound coordination scheme will allow us to assess its performance, and to evaluate how the internal agents of the system interact when certain internal/external events occur. Furthermore, it permits system behavior modeling to simulate the most critical design characteristics such as concurrency, autonomy, task distribution and parallelism, in order to guarantee robust and coherent performance. Due the complexity of modern manufacturing plants, intelligent systems (e.g., ICAM) have to be distributed, which makes the coordination of such systems very difficult and challenging.

Durfee *et al.* (Durfee & Montgomery, 1991) proposed an informal theory that integrates organizational behavior, long term plans, and short term schedules into one coordination framework, and treats coordination as a distributed search process through the hierarchical space of the possible interacting behaviors of the individual agents to find a collection that satisfactorily achieves the agents' goals. The theory emphasizes several topics such as:

- hierarchical behavior representation to express different dimensions of behavior at different levels of detail,
- metrics for measuring the quality of coordination between agents,
- distributed search protocol for guiding the exchange of information between agents during the distributed search,

- local search algorithm for generating alternative behaviors at arbitrary levels of abstractions, and
- control knowledge and heuristics for guiding the overall search process to improve coordination.

Durfee also suggested that introducing a meta-level organization in the intelligent system to manage coordination between agents, and separating knowledge representation into domain-level and meta-level types, would enhance coordination and make it more robust. Agents use domain-level knowledge to influence what goals they pursue, and use meta-level knowledge to decide how, when, and where to form and exchange behavioral models (Durfee et al., 1989). Durfee's informal theory and suggestions give the big picture of how agents should coordinate their activities within an intelligent system or even a society of intelligent agents. So far we have addressed the knowledge and organization separation issues by adopting the H-CogAff architecture proposed by Sloman. ICAM interacts with the external world through its reactive and deliberative agents, whereas the meta-level layer dictates the internal behavior of the system. Furthermore, domain-level knowledge is encoded in the deliberative agents and the meta-level knowledge is encoded in the self reflective layer.
As illustrated by figure 4, the proposed conceptual behavior model of the ICAM system was built upon our previous work in which we defined the architecture of the system, its functional modules, and its coordination mechanisms (Taylor & Sayda, 2005a;b). We adopted Sloman's H-Cogaff architectural scheme because it met most of our system requirements (Sloman, 2001). The behavioral model was drawn as a page hierarchy to make it compatible with hierarchical colored petri net (HCPN) terminology, which could be used to analyze the logical correctness and the dynamic behavior of the system; however, this has not been done.
The prime page in the model is called ICAM which contains all the subpages of the system. Each subpage represents an independent agent which interacts with others by means of communications (represented by thin bidirectional arrows). Other agents may process data received from the plant directly (data flow is represented by open thick unidirectional arrows). The meta manager is the main coordinator of the whole system, which guarantees more robust and coherent performance. The meta manager is basically a rule-based expert system, which codifies all possible system behaviors and agent interactions as a behavior hierarchy in its knowledge base. Agent behavior is represented in the behavior hierarchy by a single structure, which will use the same message structure communicated between agents. This will result in a better system performance. Table 1 illustrates the unified behavior conceptual structure.

Field name	Field content
Tag	Message ID
From	Sender
To	Recipient
What	Goals
How	Plans
When	Schedule
How long	Task length
Why	Meta reasoning

Table 1. Conceptual structure of behavioral message

6. Requirements analysis for the ICAM system

Having proposed a conceptual model, architecture, and behavioral model for the ICAM system, we define the autonomy, communications, and artificial intelligence (AI) requirements of the different agents of such a system. We also discuss the software implementation of the reactive and the supervisory agents.

6.1 Artificial intelligence (AI) requirements for the ICAM system

Among the industrial rule-based expert system shells, the G2 real-time expert system shell from Gensym Corporation (Gen, 2005) is considered the most versatile real-time expert system shell, as it integrates many software technologies and standards. The integration of the G2 expert system development environment with the ICAM system would benefit from and build on the previous G2 integration attempts. The G2 development environment offers a goal-based rapid prototyping design, in which requirements analysis, design, and development tasks are done simultaneously and incrementally during the ICAM system development life cycle. To meet the software requirements during the design and development of the ICAM system supervisory agent, AI design requirements such as the supervisory agent structure and knowledge representation have to be determined.

6.1.1 ICAM system supervisory agent implementation

Modules are the building blocks of complex G2 applications. A modular knowledge base (KB) consists of multiple G2 modules. The modules that make up an application form a module hierarchy, which specifies the hierarchical dependencies between modules (Gen, 2005). Decomposing a large project into multiple small modules allows developers to divide and merge work. Modules can be structural or functional ones. The structural modules contain classes or capabilities that need to be shared in large applications; functional modules implement well defined goals. The ICAM system supervisory agent, which potentially is a very complex artificial intelligence application, forms a good candidate for the modularization design approach. While the modularization design approach may add some overhead on the overall performance of the agent, it effectively organizes knowledge, and simplifies the development and deployment processes.

To meet the module reusability requirement, the guidelines for G2 application development recommend use of a four layer, two-module architecture, in which the graphical user interface (GUI) is in a separate module. The general architecture of the ICAM supervisory agent has two modules. The first module contains the agent's core functionality implementation layer and its application programmer's interface (API) layer, which protects the internal data structures in the core from corruption by other modules. The second module contains the public graphical user interface (GUI) layer and its GUI implementation layer, which interacts directly with the first module through its API layer. The ICAM system supervisory agent interacts with the other reactive agents through their external G2 links. The internal states of the ICAM system agents and the external environment are communicated to enable the supervisory agent to reason and make the correct and appropriate decisions for better system management.

6.1.2 Knowledge representation of the supervisory agent

The ICAM system supervisory agent may contain multi-faceted complex knowledge such as the internal structure of the ICAM system and the structure of the external environment (e.g., manufacturing plant topology, enterprise business structure). To represent such complex knowledge, organizing the knowledge structure in the core layer of the supervisory agent as a hierarchy of smaller modules would be the solution, as shown in figure 5. Each module is represented in the G2 development environment as a knowledge base (KB). Each KB represents an ontology of specified knowledge. An ontology is important for knowledge-based system development because it can serve as a software specification, similar to the function of a software architecture. Like a software architecture, an ontology provides guidance to the development process. The former provides guidance to the development process by specifying the interdependencies that deal with stages or aspects of a problem-solving process. By contrast to software architecture, however, an ontology involves not only the stages of a process, but also the taxonomy of knowledge types. The two aspects are referred to as task-specific and domain-specific architectures (Mark et al., 1995). The modular knowledge base design approach supports objected-oriented design principles, increases productivity, encourages code reuse and scalability, and improves maintainability.

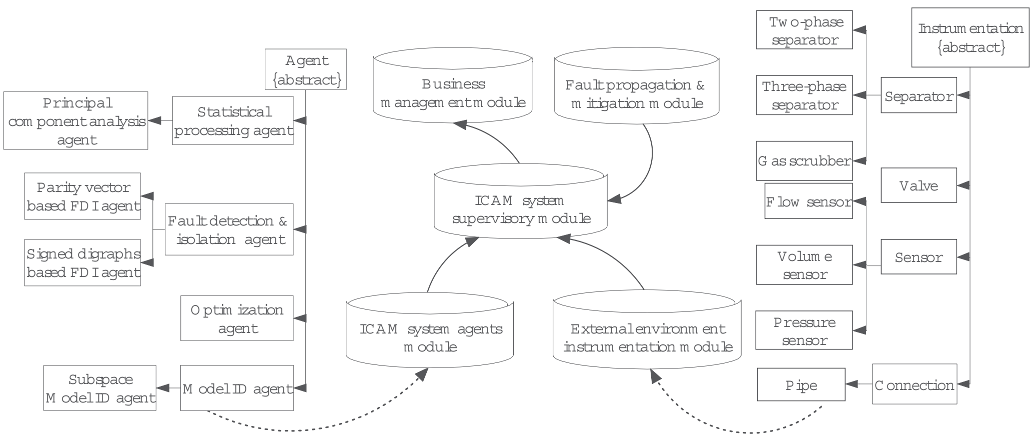


Fig. 5. Knowledge representation structure in the ICAM supervisory agent

The core (private layer) of the ICAM system supervisory agent has five modular KBs, which are organized as a module hierarchy. Basic knowledge about the ICAM system elements is represented in three lower level knowledge bases (i.e., ICAM system agents' KB, external environment instrumentation KB, and fault propagation and mitigation KB). The first knowledge base organizes the different conceptual agents of the ICAM system (e.g. fault detection and isolation agents, optimization agent, etc ...). In contrast, the second knowledge base maps the external environment physical instrumentation (e.g., valves, sensors, and other chemical process equipment) into its class hierarchy. Instrumentation and process faults and their mitigating actions are represented as classes in the third KB. Each basic element (i.e., object) in these knowledge bases has properties to represent its physical or conceptual characteristics; and has methods to represent its behavior. Elements are further organized as a class hierarchy to exploit object-oriented standards such as abstraction, inheritance, and information hiding and encapsulation.
The ICAM system supervisory knowledge base merges the knowledge from the lower level modules into a three-layer knowledge base, where each layer represents a subsystem of connected objects (i.e., classes). The first layer (i.e., the ICAM system structure layer) assembles the conceptual structure of the ICAM system from the agent class hierarchy of the lower level knowledge base. This layer is responsible for managing the internal behavior

of the ICAM system. Fault propagation and mitigating actions are assembled into object trees, and mapped into the second layer, which manages the external environment during abnormal situations. In fact, it isolates instrumentation faults, and presents their propagation maps and their appropriate migrating actions to process operators. The third layer (i.e., process topology layer) represents the external process topology, where different process instrumentation objects are used from the instrumentation knowledge base module. Other knowledge bases can be added to represent other types of knowledge such as the enterprise business management module.

A G2 rule-based system maps out a multi-threaded path of execution, which is potentially different each time the rule is invoked. For this reason, rule-based systems are often more complex, harder to test, debug, and maintain, and less efficient than procedure-based systems based on methods. Thus, rules should be used for specific purposes such as general event detection and event detection based on data driven processing and forward chaining (Gen, 2005). Since the ICAM system knowledge is multi-faceted and complex, its knowledge processing structure should be also distributed and organized according to the class and/or module hierarchy of the supervisory agent. For example, generic rules for event detection of a specific reactive agent can be organized in the class associated with that reactive agent. Rules can also be categorized to achieve certain functionality. For example, the fault propagation and mitigation schemes (i.e., cases) can be implemented as a rule category. This would narrow the scope of rules, where rules are only applied to their specified level in the class hierarchy and/or the module hierarchy. Consequently, rules invocation by forward chaining will be less prone to errors. The distribution of knowledge representation and processing would meet most of the software requirements. This would pave the way for managing complex process plants by dividing them into sub-processes that can be managed by a separate ICAM system. A universal supervisor can then manage the whole hierarchy of sub-processes efficiently.

6.2 Communication requirements for the ICAM system

It is crucial to design the agent structure to achieve specific autonomy requirements in terms of an overlapping scheme for communication and computation along with ease of prototyping and deployment. The design of the system's middleware structure, which acts as an integration model showing the types of connectivity between the different agents, is also important for achieving autonomy. Middleware is connectivity software that consists of a set of enabling services that allow multiple processes running on one or more machines to interact across a network. Middleware can take on the following different models (Aldred et al., 2005; Bernstein, 1996; Emmerich, 2000; Fox et al., 2005; Pinus, 2004):

1. Transactional middleware, which permits client applications to request several services within a transaction from a server application.
2. Procedural middleware, which enables the logic of an application to be distributed across the network, and can be executed by Remote Procedure Calls (RPCs).
3. Message-Oriented Middleware (MOM), which has become an increasingly popular solution for interoperability of heterogeneous applications. It provides generic interfaces that send and receive messages between applications through a central message server that takes charge of routing the messages.
4. Object/component middleware (e.g., CORBA, Java RMI, and Microsoft COM/DCOM technologies), which is a set of useful abstractions for building distributed systems. The communication model for this platform is based on a request/reply pattern.

5. High Performance Computing and Communication (HPCC) middleware, which is oriented toward the development of parallel computing hardware and parallel algorithms. The Message Passing Interface (MPI) communication model meets the autonomy and high performance requirements
6. Web Service-Oriented middleware, in which XML-documents (i.e., messages) are exchanged between systems using the simple object access protocol (SOAP). A SOAP message may include, for example, all necessary information for its secure transmission.

Having reviewed the different middleware technologies, it is our opinion that the high performance computing and communication MPI-based middleware meets the ICAM system requirements. The MPI communication library offers many communication modes and protocols, giving system designers the freedom and flexibility to implement their communication specifications and protocols. The MPI library specifies synchronous, buffered, ready, and nonblocking communication modes. In the synchronous mode, communicating processes are blocked till a message transfer operation is completed. However, the non-blocking mode does not block the communicating processes, which allows more flexible implementation in terms of communication/computation overlap. Buffered mode gives designers more manageability over communication buffers, whereas the ready mode guarantees correct message sending operation if a matching receiving operation is posted.

Among pre-specified MPI protocols, designers can choose from several protocols such as the Eager, the Rendezvous, and the One-sided protocols for implementation. The Eager protocol can be used to send messages assuming that the destination can store them. This protocol has minimal startup overhead and is used to implement low latency message passing for smaller messages. The Eager protocol has advantages in terms of programming simplicity and reduction of synchronization delays. However, it requires significant buffering, additional buffer copies, and more CPU involvement at the destination. In contrast, the Rendezvous protocol negotiates the buffer availability at the receiver side before the message is actually transferred. This protocol is used for transferring large messages when the sender is not sure whether the receiver actually has the buffer space to hold the entire message. This protocol is safe and robust, and may save in memory. However, it requires more complex programming and may introduce synchronization delays. The One-sided protocol (i.e., remote memory access (RMA) protocol) moves data from one process to another with a single routine that specifies both where the data are coming from and where they are going. Communicating agents using this protocol must have a designated public memory (i.e., blackboard), which can be remotely accessed. This protocol has nearly the best performance compared to others in terms of synchronization delays; however, it requires a very careful synchronization planning process (Gropp et al., 1999).

Having described the communication design options available in the MPI library and according to the high performance MPI recommendations (Gropp & Lusk, n.d.), it is our opinion that the ICAM system communications should meet the following requirements:

- In order to avoid deadlocks, synchronization time, and serialization problems, the non-blocking communication mode should be used.
- To address the message size and scalability issues, the Rendezvous protocol would be the perfect candidate among the other MPI protocols.
- The problem of buffer contention and achieving fairness in message passing can be resolved by having large communication buffers.

- The one-sided protocol can also be implemented to augment ICAM system communication performance by enabling agents to have their own private blackboards, as was discussed in the previous section.

6.3 Reactive agent software implementation requirements

In order to reconcile efficient computation with ease of prototyping requirements, the ICAM system is deployed as a distributed interconnection of reactive MATLAB computational agents, which runs on a network of several Windows XP workstations. Distributed MATLAB sessions exchange messages by using our newly developed MPI communication protocol. Exchanged messages have two roles: a control role to achieve internal coordination with other agents, and a numerical data processing role to achieve the best interaction with the external environment (e.g.,, offshore oil processing rigs) (Sayda & Taylor, 2007b).

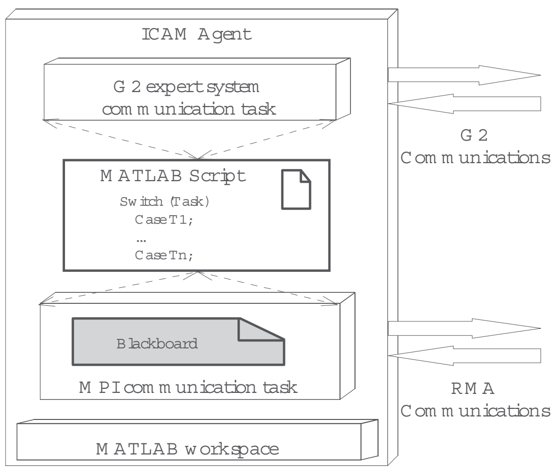


Fig. 6. ICAM system reactive agent deployment structure

Figure 6 shows the structure of a general reactive agent of the ICAM system. The general agent is implemented as a MATLAB m-script, which runs two communication tasks and a computational one. The computational task represents the agent's main functionality (e.g., model ID, fault detection and isolation, etc.). The first communication task is an MPI remote memory access (RMA) protocol, which provides the basic buffered messaging capabilities with minimum overhead (refer to the next section for more details). Furthermore, a public memory window is embedded in each reactive agent for remote access by other agents. The memory window will act as a blackboard for direct transfer of complex numerical data structures among agents. This design decision was made after investigating the advanced features of the newest MPI 2.0 library (Gropp et al., 1999), and to meet the blackboard functionality described in the behavioral model of the ICAM system.

The second communication task manages the connection with the main system supervisor (implemented as a G2 expert system). The general MATLAB template for reactive agents is built as a hierarchical finite state machine (FSM) module, which consists of two FSM layers. The first FSM is responsible for processing the ICAM system events received from the supervisory agent (i.e., the operating system of the ICAM system). The second FSM implements the specific computational functionality of the agent (e.g., FDI, model ID etc.). Further FSM layers can be added depending on the complexity of the reactive agent. Figure 7 illustrates the reactive agent implementation, which first starts its main MATLAB script and its associated graphical user interface (GUI). After the ICAM agent is instantiated and its buffers are initialized, the MPI communication environment and the G2 expert system link are initialized. The agent's specified computational task is started.

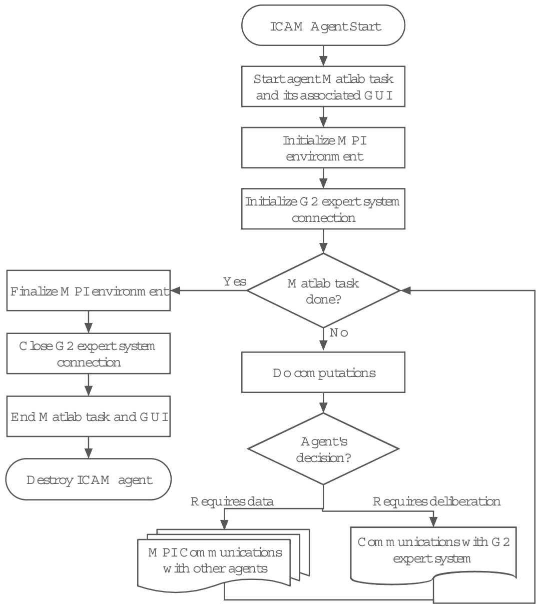


Fig. 7. Reactive ICAM agent implementation flow chart

Once the computations are done, the communication tasks are executed based on the agent's internal state and decisions. If the agent decides that it requires further deliberation about its internal state or its response to the external environment, then messages are exchanged with the ICAM system supervisor (i.e., the G2 expert system). On the other hand, if the agent requires more data for better awareness of the external environment, then it would exchange messages with other agents through its MPI link. if the computational task is done, the task is ended; if not, the computation loop continues to execute. The MPI environment is finalized, and the G2 expert system link is disconnected when the ICAM system shuts down. The proposed agent structure paves the way to design and to rapid prototype any complex multi-agent system for many applications. This definitely enables system designers to implement any communication protocol in addition to exploiting the full power of the MATLAB simulation, computation, and development environment.

7. ICAM System performance verification and validation

A prototype has been developed in order to have the ICAM system requirements deployed in a real-world system. Figure 8 portrays the simplified ICAM system prototype. Real-time data from the external plant or a simulation model are received by the statistical data monitoring agent, which preprocesses the data by removing undesired discrepancies such as outliers and missing data. Processed data are stored in a real-time database for logging and other purposes, and are then sent to the fault detection, isolation, and accommodation (FDIA) and model ID agents for further processing. When the data statistical preprocessor detects a change in the operating point or an abnormal change in data, it alerts the model ID and FDIA agents to further identify the nature of the data change. If the change is in the process operating point, the FDIA agent asks the model ID agent to update the process model parameters. If the change is a process fault (i.e., a sensor or actuator fault), the FDIA agent detects the nature of the fault and notifies the ICAM system supervisor for further processing. If the supervisor decides that a fault can be accommodated, it notifies the FDIA agent to do so. For every event that occurs, the supervisor is notified, which in turn monitors and assesses the logical behavior of the system. Processed data at every agent are sent to an operator interface, which allows operators to make the appropriate decision depending on the plant situation.

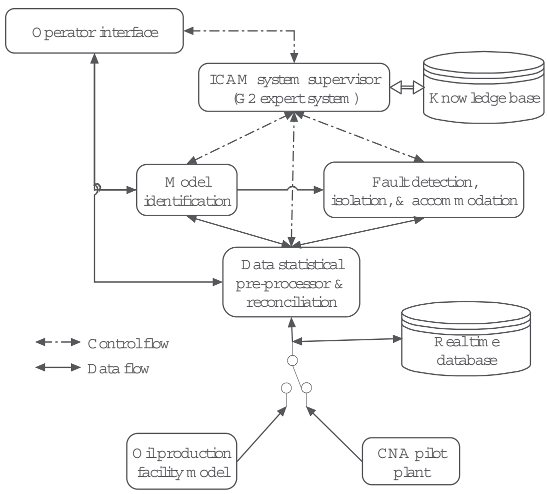


Fig. 8. ICAM system prototype

Real-time simulation experiments were designed to analyze the performance of the ICAM system prototype in terms of its logical behavior and its response to the external environment dynamics. Five different simulation scenarios were applied in real-time experiments. Three scenarios showed successful behavior by simulating three different plant faults, in which the ICAM system had an adequate knowledge about. The other two scenarios simulate situations, where the ICAM system has no knowledge whatsoever about. This would reveal The ICAM system limitation. Due to limited space, the simulation results are not discussed, however, the simulation experiments conclusions are discussed in the following section.

8. Learned lessons & future directions

Designing an intelligent multi-agent system is a very challenging task, as all agents are distributed and semi-autonomous. We faced several design challenges which resulted in limited system capabilities. Some of these design challenges and the future recommendations for solving them are suggested in the following points:

- Although we proposed the hierarchical colored petri nets approach to design the internal logic of the ICAM system reactive agents in our development plan (Sayda & Taylor, 2006), we did design the agents' internal logic in an *ad hoc* manner. We faced some difficulties during the design stage of the ICAM system prototype, as more functionalities were added. For example, the ICAM system crashed during early simulation runs due to communication deadlocks, in which two agents were trying to send messages to each other simultaneously. The problem was solved by imposing conditions on communicating agents to prevent such deadlocks. Future designs should use the colored petri net approach to verify the logical behavior of the ICAM system and its agents in different scenarios.

- Computation/communication coordination was another design problem, in which computation and communication code blocks were not ordered correctly in the agent code. For example, we combined the process model estimation (computation task) and sending the estimated model to other agents (communication task) into one task in the model ID agent, which proved to be a design flaw. Model estimation took a long time (i.e., over one minute), during which other agents were locked waiting for the estimated model due to synchronization failure. The problem was solved by separating the one functionality into two separate computation and communication functionalities (i.e., separate agent states) and modifying other agents accordingly. Although some design flaws had to be corrected, the ICAM system prototype acted as a set of distributed stochastic colored petri nets during real-time simulation. This implies that a careful agent design should be done along with a thorough system logical behavior analysis. Future design plans would take the stochastic nature of the system and time into account to guarantee robust performance.
- The industrial plant data characteristics also had a major impact on the ICAM system performance. For example, the ICAM system prototype is not robust against noisy data due to the design of the data differentiation-based steady state detection algorithm. Likewise, the FDIA algorithm is not robust to noise, which significantly affects the fault isolation task in moderate to high noisy data situation. We suggest embedding algorithms that are more robust to noise to cope with real-world industrial plants and their noisy measurements.
- Detection and isolation of fast dynamics faults (e.g., faulty gas pressure sensor) is another limitation of the ICAM system prototype. The outlier removal algorithm in the statistical processing agent treats fast dynamics faults as outliers, which changes the nature of processed data sent to the FDIA agent. Data filtering also may change the data characteristic, which may have an impact on the system performance. In addition, the system logical behavior was unpredictable and inconsistent in response to disturbances in process variables. So we suggest developing a better safety net, in which the knowledge of agents' limitations is embedded in the rule base of the supervisory agent. This allows the system to have a better reasoning ability and robust performance during undefined and unpredictable plant situations.
- The incorporation of domain knowledge would definitely improve the performance of the system. Such knowledge is represented by the topology of the industrial plant and its operation procedure in different situations such as startup, normal operation, and shutdown. This knowledge would be better utilized if a learning agent were embedded to deal with new situations in the plant and the internal behavior of the ICAM system itself.

As can be appreciated, those enhancements will require years of additional research and development.

9. References

Aldred, L., van der Aalst, W. M. P., Dumas, M. & ter Hofstede, A. H. M. (2005). On the notion of coupling in communication middleware, *International Symposium on Distributed Objects and Applications (DOA)*, Agia Napa, Cyprus.

Banks, S. B. & Lizza, C. S. (1991). Pilot's associate: a cooperative, knowledge-based system application, *IEEE Expert* 6(3): 18–29.

Bernstein, P. A. (1996). Middleware: A model for distributed services, *Communications of the ACM* 39(2): 86–97.

Cauvin, S. (2004a). CHEM-DSS : Advanced decision support system for chemical/petrochemical industry, *Fifteenth International Workshop on Principles of Diagnosis (DX'04)*, AAAI, Carcassonne, France.

Cauvin, S. (2004b). CHEM-DSS: Advanced decision support system for chemical/petrochemical manufacturing processes, *CHEM Project Annual Meeting*, http://www.chem-dss.org/, Lille, France.

Cochran, E. L., Miller, C. & Bullemer, P. (1996). Abnormal situation management in petrochemical plants: can a pilot's associate crack crude, *Proceedings of the 1996 IEEE National Aerospace and Electronics Conference, NAECON*, Vol. v2, IEEE, Piscataway, NJ, USA, Dayton, KY, USA, pp. 806–813.

Cochran, T., Bullemer, P. & Nimmo, I. (1997). Managing abnormal situations in the process industries parts 1, 2, 3, *NIST Proceedings of the Motor Vehicle Manufacturing Technology (MVMT) Workshop*, Ann Arbor, MI.

Derriso, M. (2005). Intelligent vehicle health management for air force space systems, *ISHM/NASA session of the IEEE Sensors for Industry Conference*, IEEE/ISA, Houston, TX, USA.

Durfee, E., Lesser, V. R. & Corkill, D. D. (1989). Trends in cooperative distributed problem solving, *IEEE Transactions on Knowledge and Data Engineering* 1(1): 63–83.

Durfee, E. & Montgomery, T. (1989). MICE: A flexible test bed for intelligent coordination experiments, *Proceedings of the 9th workshop on distributed AI*, Rosario, Washington.

Durfee, E. & Montgomery, T. (1991). Coordination as distributed search in a hierarchical behavior space, *IEEE Transactions on Systems, Man, and Cybernetics* 21(6): 1363–1378.

Emmerich, W. (2000). Software engineering and middleware: a roadmap, *Proc. of the Conference on The Future of Software Engineering*, IEEE Computer Society, Limerick, Ireland, pp. 117–129.

Figueroa, F. (2005). Integrated health with networked intelligent elements (IHNIE) prototype, *ISHM/NASA session of the IEEE Sensors for Industry Conference*, IEEE/ISA, Houston, TX, USA.

Figueroa, F., Holland, R. & Schmalzel, J. (2006). ISHM implementation for constellation systems, *42nd AIAA/ASME/SAE/ASEE Joing Propulsion Conference and Exhibit*, Sacramento, CA, USA.

Figueroa, F., Holland, R., Schmalzel, J. & Duncavage, D. (2006). Integrated system health management (ISHM): systematic capability implementation, *Proceedings of the 2006 IEEE Sensors Applications Symposium*, pp. 202–206.

Fox, G. C., Aktas, M. S., Aydin, G., Gadgil, H., Pallickara, S., Pierce, M. E. & Sayar, A. (2005). Algorithms and the grid, *Conference on Scientific Computing*, Vysoke Tatry, Podbanske.

Garcia-Beltran, C., Exel, M. & Gentil, S. (2003). An interactive toot for causal graph modeling for supervision purposes, *In Proc. IEEE International Symposium on Intelligent Control (ISIC)*, Huston, Texas, pp. 866– 871.

Garcia-Galan, C. (2005). Integrated system health management for exploration mission systems, *ISHM/NASA session of the IEEE Sensors for Industry Conference*, IEEE/ISA, Houston, TX, USA.

Gen (2005). G2 *for Application Developers Reference Manual*, 8.0 edn.

Gropp, W. & Lusk, E. (n.d.). Tuning MPI applications for peak performance. www.mcs.anl.gov/Projects/mpi/tutorials/perf, Argonne National Laborary.

Gropp, W., Lusk, E. & Thakur, R. (1999). *Using* MPI-2*: Advanced features of the message-passing interface*, Scientific and Engineering Computation, MIT Press, Cambridge, Massachusetts.

Jennings, N. R. & Mamdani, E. M. (1996). Using ARCHON to develop real–world DAI applications parts 1, 2, 3, *IEEE Expert* 11(6): 64–86.

Karsai, G., Biswas, G., Abdelwahed, S., Mahadevia, N., Keller, K. & Black, S. (2005). Intelligent component health management: An architecture for the integration of IVHM and adaptive control, *ISHM/NASA session of the IEEE Sensors for Industry Conference*, IEEE/ISA, Houston, TX, USA.

Kim, I. S. & Modarres, M. (1987). Application of goal tree-success tree model as the knowledge-base of operator advisory systems, *Nuclear Engineering and Design* 104: 67–81.

Köppen-Seliger, B., Marcu, T., Capobianco, M., Gentil, S., Albert, M. & Latzel, S. (2003). MAGIC: An integrated approach for diagnostic data management and operator support, *Proceedings of the 5th IFAC Symposium Fault Detection, Supervision and Safety of Technical Processes - SAFEPROCESS05*, Washington D.C.

Larimore, W. (2005). *Multivariable System Identification Workshop*, University of New Brunswick, Fredericton, New Brunswick.

Laylabadi, M. & Taylor, J. H. (2006). ANDDR with novel gross error detection and smart tracking system, *12th IFAC Symposium on Information Control Problems in Manufacturing*, IFAC, Saint-Etienne, France.

Liao, S. H. (2004). Expert systems: Methodologies and applications, a decade review from 1995 to 2004, *Expert systems with applications* pp. 1–11.

Liebowitz, J. (1998). *The Handbook Of Applied Expert Systems*, CRC Press, Boca Raton, FL.

Mark, W., Dukes-Schlossberg, J. & Kerber, R. (1995). *Towards very large knowledge bases*, IOS Press/Ohmsha, msterdam/Tokyo, chapter Ontological commitment and domain specific architectures: Experience with comet and cosmos.

Matania1, R. (2005). Interoperability and integration oil and gas science and technologyof industrial software tools, *Oil and Gas Science and Technology* 60(4): 617–627.

Maul, W. A., Park, H., Schwabacher, M., Watson, M., Mackey, R., Fijany, A., Trevino, L. & Weir, J. (2005). Intelligent elements for the ISHM testbed and prototypes (ITP) project, *ISHM/NASA session of the IEEE Sensors for Industry Conference*, IEEE/ISA, Houston, TX, USA.

Miller, C. A. & Hannen, M. D. (1999). Rotorcraft pilot's associate: Design and evaluation of an intelligent user interface for cockpit information management, *Knowledge-Based Systems* 12(8): 443–456.

Moore, R. L. & Kramer, M. (1986). Expert systems in online process control, *Proceedings of the 3rd international conference on chemical process control*, Asilomar, California.

Mylaraswamy, D. (1996). DKIT*: a blackboard-based, distributed, multi-expert environment for abnormal situation management*, PhD thesis, Purdue University.

Mylaraswamy, D. & Venkatasubramanian, V. (1997). A hybrid framework for large scale process fault diagnosis, *Computers and Chemical Engineering* 21: S935–S940.

Mylopoulos, J., Kramer, B., Wang, H., Benjamin, M., Chou, Q. B. & Mensah, S. (1992). Expert system applications in process control, *In Proc. of the International Syposium on Artificial Intelligence in Materials Processing Applications*, Edmonton, Alberta, Canada.

Newell, A. (1990). *Unified theories of cognition*, Harvard University Press, Cambridge, MA.

Ogden-Swift, A. (2005). Reducing the costs of abnormal situations ... the next profit opportunity, *IEEE Advanced Process Control Applications for Industry Workshop (APC2005)*, Vancouver, Canada.

Omana, M. & Taylor, J. H. (2005). Robust fault detection and isolation using a parity equation implementation of directional residuals, *IEEE Advanced Process Control Applications for Industry Workshop (APC2005)*, Vancouver, Canada.

Omana, M. & Taylor, J. H. (2006). Enhanced sensor/actuator resolution and robustness analysis for FDI using the extended generalized parity vector technique, *Proc. of American Control Conference*, IEEE, Minneapolis, Minn., pp. 2560–2566.

Omana, M. & Taylor, J. H. (2007). Fault detection and isolation using the generalized parity vector technique in the absence of a mathematical model, *IEEE Conference on Control Applications (CCA)*, Singapore.

Pinus, H. (2004). Middleware: Past and present a comparison.
URL: *http://www.research.umbc.edu/ dgorin1/451/middleware/middleware.pdf*.

Sayda, A. F. & Taylor, J. H. (2006). An implementation plan for integrated control and asset management of petroleum production facilities, *IEEE International Symposium on Intelligent Control ISIC06*, IEEE, Munich, Germany, pp. 1212–1219.

Sayda, A. F. & Taylor, J. H. (2007a). An intelligent multi agent system for integrated control and asset management of petroleum production facilities, *In Proc. of The 17th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM)*, Philadelphia, USA, pp. 851–858.

Sayda, A. F. & Taylor, J. H. (2007b). Modeling and control of three-phase gravity separators in oil production facilities, *the American Control Conference (ACC)*, New York, NY.

Sayda, A. F. & Taylor, J. H. (2007c). Toward a practical multi-agent system for integrated control and asset management of petroleum production facilities, *IEEE International Symposium on Intelligent Control (ISIC)*, Singapore.

Sayda, A. F. & Taylor, J. H. (2008a). A multi-agent system for integrated control and asset management of petroleum production facilities - part 1: Prototype design and development, *accepted for the IEEE International Symposium on Intelligent Control (ISIC)*, San Antonio, Texas, USA.

Sayda, A. F. & Taylor, J. H. (2008b). A multi-agent system for integrated control and asset management of petroleum production facilities - part 2: Prototype design verification, *accepted for the IEEE International Symposium on Intelligent Control (ISIC)*, San Antonio, Texas, USA.

Sayda, A. F. & Taylor, J. H. (2008c). A multi-agent system for integrated control and asset management of petroleum production facilities - part 3: Performance analysis and system limitations, *accepted for the IEEE International Symposium on Intelligent Control (ISIC)*, San Antonio, Texas, USA.

Schmalzel, J., Figueroa, F., Morris, J., Mandayam, S. & Polikar, R. (2005). An architecture for intelligent systems based on smart sensors, *IEEE Transactions on Instrumentation and Measurement* 54(4): 1612–1616.

Sloman, A. (2001). Varieties of affect and the COGAFF architecture schema, *proceedings of symposium on Emotions, Cognition, and Affective Computing at the AISB'01 convention*, York, UK.

Sloman, A. & Scheutz, M. (2002). Framework for comparing agent architectures, *Proceedings of the UK Workshop on Computational Intelligence*, Birmingham, UK.

Small, R. L. & Howard, C. W. (1991). A real-time approach to information management in a pilot's associate, *Proceedings of Digital Avionics Systems Conference*, IEEE/AIAA, pp. 440–445.

Smith, C., Gauthier, C. & Taylor, J. H. (2005). *Petroluem Applications of Wireless Sensors* (PAWS) *Workshop*, Cape Breton University, Sydney, Nova Scotia.

Taylor, J. H. (2004). Petroleum applications of wireless systems - UNB's control/information technology subproject, *submitted to Cape Breton University on 11 December 2003 and subsequently to ACOA on 21 September 2004*.

Taylor, J. H. & Laylabadi, M. (2006). A novel adaptive nonlinear dynamic data reconciliation and gross error detection method, *Proc. of IEEE Conference on Control Applications*, IEEE, Munich, Germany, pp. 1783–1788.

Taylor, J. H. & Omana, M. (2008). Fault detection, isolation and accommodation using the generalized parity vector technique, *submitted to the IFAC World Congress*, Seoul, Korea.

Taylor, J. H. & Sayda, A. F. (2005a). An intelligent architecture for integrated control and asset management for industrial processes, *Proc. IEEE International Symposium on Intelligent Control (ISIC05)*, Limassol, Cyprus, pp. 1397–1404.

Taylor, J. H. & Sayda, A. F. (2005b). Intelligent information, monitoring, and control technology for industrial process applications, *The 15th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM)*, Bilbao, Spain.

Taylor, J. H. & Sayda, A. F. (2008). Prototype design of a multi-agent system for integrated control and asset management of petroleum production facilities, *accepted for the American Control Conference (ACC)*, Seattle, Washington.

Vedam, H. (1999). OP-AIDE*: an intelligent operator decision support system for diagnosis and assessment of abnormal situations in process plants.*, PhD thesis, Purdue University.

Vedam, H., Dash, S. & Venkatasubramanian, V. (1999). An intelligent operator decision support system for abnormal situation management., *Computers and Chemical Engineering* 23: S577–S580.

Venkatasubramanian, V. (2005). Prognostic and diagnostic monitoring of complex systems for product lifecycle management: Challenges and opportunities., *Computers and Chemical Engineering* 29: 1253–1263.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N. & Yin, K. (2003). A review of process fault detection and diagnosis part 1, 2, 3, *Computer & Chemical Engineering* 27(3): 293–346.

Wang, H. & Wang, C. (1996). APACS: A multi-agent system with repository support, *Knowledge-Based Systems* 9(5): 329–337.

Wang, H. & Wang, C. (1997). Intelligent agents in the nuclear industry, *Computer* 30(11): 28–34.

Wilikens, M. & Burton, C. J. (1996). FORMENTOR: Real-time operator advisory system for loss control. application to a petro-chemical plant, *International Journal of Industrial Ergonomics* 17: 351–366.

Wooldridge, M. J. (2002). *An introduction to multiagent systems*, Wiley, Chichester, England.

Worn, H. (2004). DIAMOND: Distributed multi-agent architecture for monitoring and diagnosis, *Production Planning and Control* 15: 189–200.

24

Bio-Inspired Multi-Agent Technology for Industrial Applications

Prof. Petr Skobelev
Institute of Control of Complex Systems of Russian Academy of Science
"Knowledge Genesis" Group of Companies
Smart Solutions, Ltd
Russia

1. Introduction

Basic ideas of multi-agent technology (MAT) began to form in the latest decades of the 20^{th} century at the edge of artificial intelligence, object-oriented and parallel programming, and telecommunications [1-4].

The European Union Association of multi-agent systems (MAS) developers AgentLink [6] presents the Road-map of MAT up to 2020-2030 years with the slogan "Computing as Interactions". This slogan shows value of the technology not only for developing modern fully distributed systems which are now rapidly growing everywhere (what is called "ambient" and "ubiquitous" intelligence [5]) but also for solving complex problems which are difficult or even impossible to solve by classical mathematical methods or algorithms, for example, in scheduling and optimization, pattern recognition, text understanding, clustering, etc.

This relatively new area of using MAT for solving complex problems is also based on ideas of interactions. But it helps to turn complex systems from large centralized, monolithic and sequential programs with fixed hierarchical structure to distributed communities of small autonomous programs working asynchronously and in quasi-parallel with opportunity to form networking structures and interact, compete and cooperate for complex problems solving.

The value of such MAS for modern complex and rapidly changing world is difficult to overestimate. This is supported by the impressive statistics of scientific community increasing interest to these subjects. In the end of 80s MAS workshops gathered together 25-30 researchers and developers from 5-7 countries. Nowadays the situation has changed greatly. For example, in the last year World conference of Autonomous Agents and Multi-Agent Systems (AAMAS-2009) participated more than 600 delegates from 45 countries representing the results in the area of agent reasoning logic, knowledge presentation methods, platforms for multi-agent systems developments and application systems in the wide range of applications, from social processes modelling to robot control. However if the amount of scientific works in this area is rising rapidly the commercial projects and practical applications are not so well developed in spite of the fact that more than 25 commercial companies and 100 university projects in this area are world well-known nowadays.

The reason of this fact is the complexity and novelty of this new very attractive paradigm of software engineering which strongly demands new methods and tools for industrial

applications. But despite all difficulties there is a number of first successful industrial MAS projects all over the world from of network-centric logistics applications for military applications to optimization of energy consumption for cottages. The list of commercial companies that are actively developing MAS began rise distinctly at the turn of the century: LostWax (The Great Britain) – 1996 [7], Whitestein Technology / Living Systems (Switzerland) – 1999 [8], NuTech Solution (USA) – 1999 [9], etc.
According to the AgentLink Association [6] the list of these companies includes the British-Russian company Magenta Technology [10], co-founders are professor G.A. Rzhevsky from the Open university (The Great Britain) and the author of this paper. The company was founded in 1999 on the basis of the software engineering company "Knowledge Genesis" and gained valuable experiences of developing multi-agent systems for industrial applications, growing from the small group of enthusiasts in Samara in 2000 to 150 highly qualified programmers with central office in London in 2010.
The results achieved in many ways are based on the complex systems research which was held in the Institute of Control of Complex Systems of Russian Academy of Sciences under the leadership of Professor V.A. Vittih without whose strong personal support these works would not get a chance.
In this paper we will discuss main concepts, give overview of first generation of our MAT platform and MAT products for industrial applications, present new key ideas for next generation of our MAT platform and describe our current more complex and advanced MAT projects under development in "Knowledge Genesis" Group of Company and Software Engineering Company "Smart Solutions" fully specialized in real time scheduling.
We hope that presented results will stimulate new MAT developments in many new different industrial applications.

2. Theoretical background of MAT for solving complex problems: from swarm intelligence to emergent intelligence

The birthday of Samara's scientific school of multi-agent systems can be considered by June 15th, 1990. This day professor George Rzevski (Open University, London, UK) gave a series of lectures in the Institute for the Control of Complex Systems (ICCS) of Russian Academy of Sciences (previously known as the Samara's branch of Institute of Machines of Russian Academy of Sciences) about the new global real-time economy, modern theory of complexity and multi-agent systems as the new paradigm for solving complex problems. He was invited to Samara by professor V.A. Vittih the director of ICCS that time.
This series of exciting and innovative lectures became the starting point for long-term partnership of scientists from London and Samara and establishing in Samara the new R&D activities in MAS for real time logistics, text understanding and clustering and a number of other applications.
At this time e-commerce MAT applications for the Internet were the main direction of developments. It was considered that software agents will quickly become "virtual personalities" having developed mechanisms of perception, cognition, reasoning and learning based on Prolog-like inductive or deductive machines and similar tools.
At the same time well-known combinatory methods and algorithms (for example method of branches and borders) were dominating in multi-agent applications for solving complex problems, for example resource optimization.

Comparing with this approach our developments initially took completely different direction of R&D work that is called now “bio-inspired” or “swarm intelligence” approach [11]. This approach for complex problems solving is based on fundamental concepts of self-organization and evolution similar to living organisms, for example, as colonies of ants or swarms of bees.

The solution of any complex problem in such types of systems is being formed evolutionally in the process of ongoing competition and cooperation of hundred and thousands of simultaneously working very simple software agents organized as a small autonomous programs. Autonomy of agents means that agent can be invoked as state-less method in object-oriented programming but could be only asked by other agents to implement required task and can accept or reject the task because of previously agreed obligations to other agents.

For example, for solving complex optimization problems Ant Colony optimization method was developed, where the behavior of the getting food ants is modeled. The success of one ant in getting "food", i.e. taking some decisions, prompting other ants a correct direction, but after some time pheromone signs on this successful direction is "fade" requiring new trials of ants. In this case the solution of complex optimization problem can be found by interaction of a big number of relatively very simple agents continuously making trial and error attempts to get better results. But building such a solution in “swarm intelligence” approach can take rather long time while the result cannot be guaranteed and this approach is difficult to be applied in real-time.

But at this first stage “swarm intelligence” has proved that self-organization becomes an important alternative to the classical mathematics and also to traditional vision of “artificial intelligence” systems. In this approach “intelligence” should be considered not as a “mechanical” assembly of some intelligent “blocks” like “deduction”, “induction” and some other (as assembly of mechanical parts in car industry). In Swarm Intelligence the “intelligence” is not located in any of block – it is considered to be the emergent property of the system as the result of interaction of huge number of not-intelligent elements. The fact that intelligence of one ant or bee is relatively small but intelligence of colony of ants or swarm of bees is a powerful organization with high level of “adaptive intelligence” allowing to defense the nest, discover new territories, find food and solve many other tasks in continuously changing environments.

An important step in development and research of this area was done by Artur Koestler who presents the concept of holonic systems where “holons” representing properties of “parts” and “wholes” where considered as a new type of actively working building “blocks” for creation of self-organized systems [13].

The first implementation of this concept in PROSA system was done by Hendrik van Brussel, Paul Valkenaers and other authors from Christian university (Belgium) [14]. In this approach agents of “orders”, “products” and “resources” were introduced as well as “staff” agent which keeps all knowledge for decision making and advise other agents when required. This approach was successfully developed in multi-agent systems for first industrial projects for manufacturing by the team of professor Vladimir Marek (Prague Technical University, Check Republic) [15]. This approach was applied for creation of manufacturing systems for Skoda factory, control of submarine subsystems for Rockwell International, etc. This approach was advanced by the team of professor Paulo Leitao (Polytechnic Institute of Braganka, Portugal), for example, for developing ADACOR system for manufacturing [16]. These all works results in starting International Conference of Holonic and Multi-Agent Systems in Manufacturing (HoloMAS).

On the top of these developments we formed our vision of MAS considering and highlighting the following key features (Fig. 1).

Traditional systems

- Hierarchies of large programs
- Sequential execution of operations
- Instruction from top to bottom
- Centralized decision
- Data driven
- Predictability
- Stability
- Striving to reduce the complexity
- Total control

Multi-agent systems

- Large networks of small agents
- Parallel execution of operations
- Negotiations
- Distributed decisions
- Knowledge driven
- Self-organization
- Evolution
- Striving to thrive with the complexity
- Support for growth

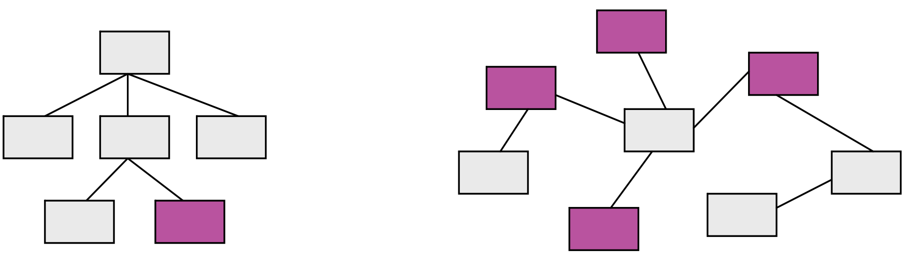


Fig. 1. Specific features of Multi-agent systems

We proposed to create multi-agent systems as a set of quasi-parallel autonomous agents designed as a co-programs with direct communication as well as with not direct communications. For not direct communication we introduced ontology-based specifications of scenes in common and shared memory of our systems which represents by semantic networks of instances of concepts and relations of domain ontology.

For these reasons the new own MAT platform was developed over Windows/Linux operation system.

In this platform agents have the following features:

- Work autonomously in platform environment, that is the agent cannot be called as a simple method but should have its own state and runs constantly. Agent cannot be forced to do something he can be only asked to do some task and he can either accept or decline the proposal depending on its goals, current state, active scenarios, etc.
- React on events and take or change decisions, including selection of scenarios and estimation of results. Agent should be able to terminate executing scenario, percept new information and react on new events.
- Communicate and negotiate with other agents what includes both direct and indirect negotiations. In the process of negotiations agents can ask each other questions, inform about the changes of their state, confirm agreement or disagreement with proposals of other agents, etc.

As a result we combine swarm-based approach with more advanced team work of agents when they can coordinate their decisions by specific advanced protocols of negotiations on virtual market of demands and resources which we will consider in more details below.

The reviewed features differ the developed platform from such well-known platforms as JADE, Cougaar, Agent Builder, JACK and others which are using different separate threads of activity for work of every separate agent what leads to significant slowdown of the whole system.

The advantage of the proposed platform is the opportunity to solve complex problems in real-time, openness, high flexibility and performance of designed systems that can be specifically applied in different spheres.

But the main feature of designed systems is their ability to demonstrate very complex behavior generated by millions of transactions of very simple agents tat generate spontaneous unpredictable revision of taken decisions at unforeseen moments of time by different agents – showcasing such phenomena of complex adaptive systems based on un-linear thermodynamics as a chaos and order, catastrophes, oscillations, etc [17].

This type of intelligence that we call “Emergent Intelligence” became the main research topic in the Samara scientific school of MAS applications and the basis for developing first generation of really intelligent systems for industrial applications in different domains.

3. Resource and demand networks of agents

3.1 Basic classes of agents in RD networks

As a one of the first problem domains we considered real time logistics of mobile resources. The problem of real time allocation, scheduling and optimization of resources is one of the most important modern problems that is characterized with a high level of uncertainty and dynamics, requires individual approach to users with conflicting interests, etc.

In our approach [18-20] was proposed to advance resource - demand networks of agents (RDN), where we define the agents (roles) of resources and demands as an entities with opposite interests, that operate on the virtual market according their economic reasons and can compete or cooperate with each other.

In this case, RDN of any domain is formed by the needs (demands) and abilities (resources) of its elements. In the simplest case, orders and resources is constantly striving to find each other and establish links.

The demand role is to get "ideal" results and the resource role – to provide best possible options in "reality". Thus, each vehicle in multi-agent logistic system knows his route, point of destination, what cargo is loaded, etc. Receiving proposals from various trucks, order can decide which of them he is best suited. But, on the other hand, the truck itself may create a new "needs", specifying exactly what orders he needed at the current time to be fully loaded or get nearest fuel station or maintenance service, driver, etc.

In an increasingly complex world of freight transportation logistic an RDN model can take into account the needs and abilities of customers and orders, trucks and cargo, travel routes, stores and warehouses, truck drivers, repair shops, fuel stations, etc. In this case, the order is constantly looking for the best truck, and truck, from the opposite side, is looking for the best order, and also the best route and the driver, etc. As a result RDN model will become more and more close to the real world transport network. This model can be expanded by the introduction of new classes of agents representing the interests of new various physical or abstract entities and with increasing number and variety of classes of interaction protocols between these agents.

A number of new RDN-based methods and tools were developed for designing first generation of industrial multi-agent systems [21-28].

In this approach each agent can be formed by the swarm of other low level agents and join the community of such agents or at any time can leave it if he is not satisfied with conditions. For example, one agent of orders allocated for one of trucks, can decide to leave swarm of truck – but as a result the conditions for other agents will change and maybe another few agents of orders will also decide to leave the truck. As a result a group of agents may leave the truck in a snowfall transition process that will become "bankrupt" on virtual market of the system – that will system chance to change attractor and optimize resulting schedule.
This approach allows us to combine the "selfish" interests of individual virtual market agents with the interests of groups of agents using the common principles of self-organization and evolution.

3.2 The virtual market of RDN agents

The core part of our any MAS is a common virtual market on which agents can buy or sell their services according to their economic reasons (Fig. 2).

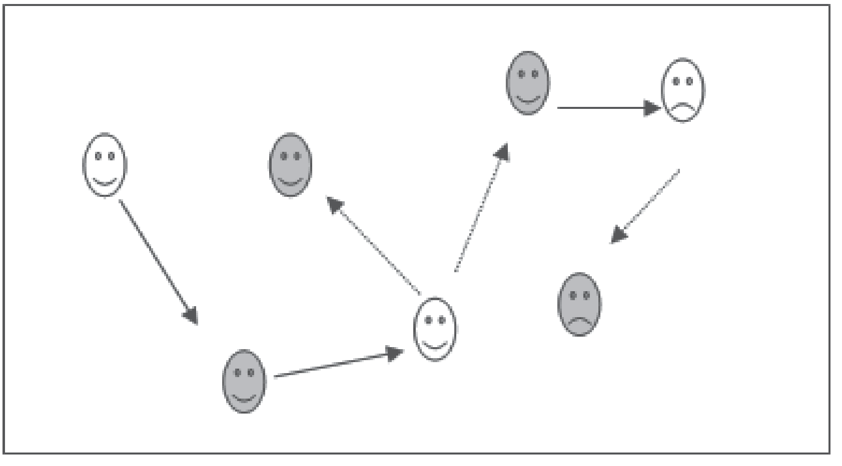


Fig. 2. The virtual market scene: demand agents (white) and resource agents (gray). Faces of agents mean satisfaction level. Different types of links shows different stages of negotiations (pre-booking, etc.)

The constant activity of all agents, from both the resource and demand sides, force multi-threads negotiations on the virtual market, going quasi-parallel [24]. In this case, each agent is designed as a state machine returns control to the dispatcher after each step of negotiations. Each agent is constantly trying to achieve its goals going into links defined on the scene with other agents (the order is booked on a truck, truck on the driver, etc.). These links could be changed by agents through recognizing and resolving conflicts, generated by events coming from outside or generated by the system itself. Agent's decisions cause a change in conditions for other agents and thereby trigger the process of self-organization in the system, leading to a reconstruction of the schedule in response to event.
Thus, the RDN always represent current solution of complex problem where agents never stop in the process of self-organization and provide adaptability of the solution, for example, form and adapt schedule in real time ("living schedule").
Rules for agents decision-making on the virtual market are determined by the microeconomic model of RD-networks, that define the virtual cost of such services, the

penalties and bonuses, rules for sharing their profits, what taxes and under what conditions should be paid, etc.
That is designed to give agents an opportunity to accumulate virtual money which plays the role of energy in the system and use them to create new or adapt fragments of existing solution.

3.3 The RDN-based MAS architecture

The RDN-based MAS architecture is presented on Fig. 3.
A key component of our MAS architecture is a virtual market which provide demand and resource role models, microeconomics, taxes, etc.
Ontology (knowledge base) for agents decisions, can be separated from the program code and updated by users. This system is provided with a special software tool to support and manage ontology and scenes [23]. In this case, every real situation could be described and stored in the system as a scene - instantiation of the concepts and relations from ontology, linking specific instances of objects (the name of the client, the driver's name, vehicle number, etc.).

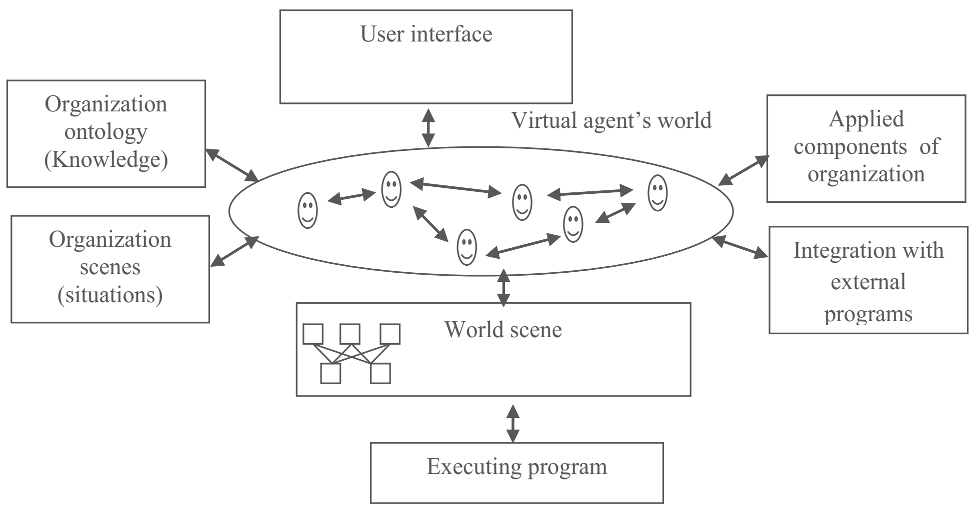


Fig. 3. MAS architecture for RD-networks implementation

The user can interact with system through web or desktop user interface. Important element of user interface is the queue of events.
When the new event coming (a new order e.g.) the system creates his demand agent, who on behalf of this order comes into interaction with the agents of resources to find the best match to place the order. If the best resource is busy, the system is fixing the conflict and making attempts to find a solution or resolution by shifts, swaps, etc. During this process, the resource (the busy one) may offer orders earlier placed on it to look for a new allocations. This process, like a chain reaction can influence and change other orders-resources allocations, forming a wave of changes (such as from the stone thrown into the water).
Similarly, if for any reason the selected resource (for example, vehicle) becomes unavailable (breakdown, accident, etc.), his agent has to find all orders scheduled on the truck and tell

inform them about the unavailability of the resource. Then the agents of the orders wake-up and activate and begin to look for other vacant trucks.
Output solutions (e.g. schedule of resources), as it was mentioned, is not considered as "static" data structure - result of a single algorithm application, but as a delicate balance (or "unstable equilibrium"), supported by the interaction of two main classes of agents of "demands" and "resources" playing the role of opposite entities of any kind like "yin" and "yang".
We consider the result is reached so the system can complete its work if there is no one chance for the agents to improve their condition.
Integration with 3rd systems is made through special integration modules.

3.4 The compensation method for balancing interests of RDN agents

To implement the developed approach a number of methods and tools uses the various swarm organizations of RDN were proposed.
The main idea of developed method of compensation [25] is that new agent needs to compensate losses for other agents which change their allocations under the request from new agent. This can made the "wave" of negotiations, in a simplified form shown in Fig. 4 - 6.

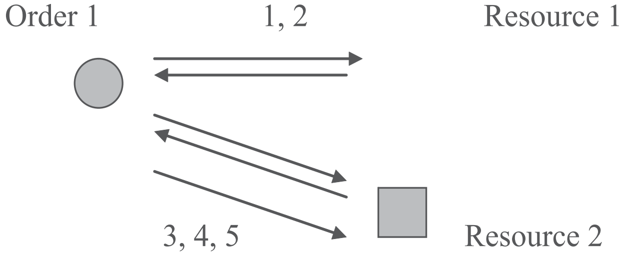


Fig. 4. The agent of the order detects two resources available and books the second one: 1 - request to the first resource, 2 - answer and suggestion from the first resource, 3 - request to the second resource, 4 - answer and a suggestion from the second resource, 5 - the second resource selected and booked

The developed approach utilizes many of the modern ideas of optimization, presented in meta-heuristics, creating an environment of competing and cooperating algorithms (agents). Thus, the agents can remember and avoid bad decisions by using their memory, they can inform each other about intermediate options, stop searching according time constraints for decision-making, etc.
An important advantage of this resource optimization technology is the ability to build adaptive plan when it is not constructed anew each time when an event occurred (as it is done in the classical optimization techniques) but adjusted in real time with the appearance of events.
This adaptation is carried out continuously by identifying conflicts in schedules, negotiating and compromising between the agents of orders and resources. This allows the system to operate in real time.
This adaptation is carried out continuously by identifying conflicts in schedules, negotiating and compromising between the agents of orders and resources and this allows the system to operate in real time.

It is interesting to note that this approach was successfully applied not only to solve the logistic tasks, but also in texts understanding, clustering, web-marketing and several others.

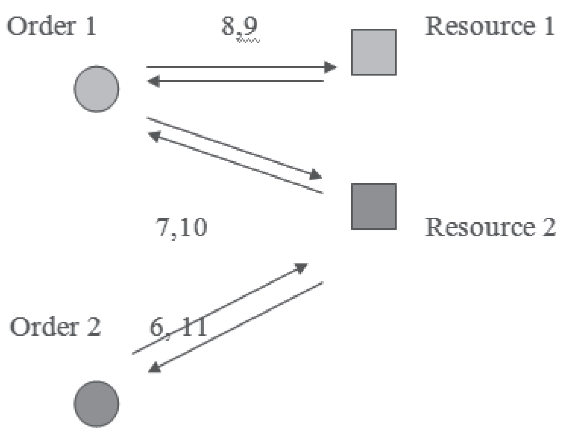


Fig. 5. The second order arrival and rebooking the second resource to the second order, with compensation: 6 - request to the second resource, 7 - divert suggestion to the first order from the second resource, 8 - request to the first resource, 9 - positive response and the offer, 10 – allowance for the second resource re-booking, 11 - agreement with the sum of compensation to the first order

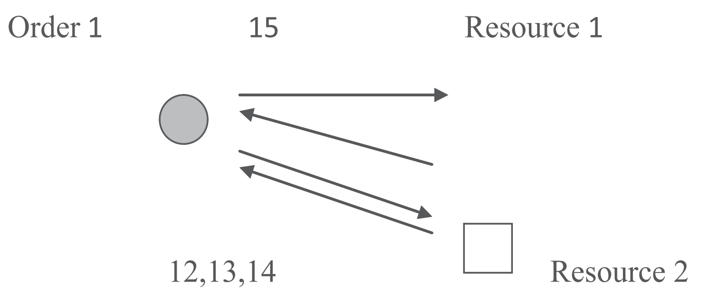


Fig. 6. The free resource appearing in the system revise the situation and reschedule booked resources, even during the execution of the order
12 - free resource initiate the running order, 13 – rebooking to the second resource is beneficial for the first order, 14 - confirmation and the second resource booking, 15 - the first resource notified about the route changing and cargo reloading to the second resource

4. The first MAS platform and the first applications (1999 - 2003)

On the basis of software engineering company “Knowledge Genesis” the Magenta Technology company was established in London in 2000 for developing first generation of multi-agent systems for solving complex problems (http://www.magenta-technology.com). One of the main catalysts of new company and its development centre in Samara was prof. G.A. Rzhevsky, whose forward-thinking vision and great knowledge and experience in complexity science made big impact on all developments. From that moment an intensive creative work on the development of multi-agent systems has begun in Samara. It was largely on the enthusiasm, overcoming any boundaries of the working time team that created real "collective intelligence" of the company. The jointed international team constantly generated new ideas many of that were implemented and grown in industrial applications.

At that period of time a few multi-agent platforms for building of multi-agent systems were developed among which the most well-known are:

- Jade (Java Agent Development Platform) – the most widely used multi-agent platform. It provides a middleware compatitable with FIPA standards to simplify the process of development of multi-agent systesm. It is promoted by Telecom Italia but supervisory board also includes Motorola, Whitestein Technologies AG, Profactor GmbH and France Telecom R&D.
- Cougar (Cognitive Agent Architecture) – also a Java-based platform for distributed multi-agent systems developed as the result of long-term project of famous military organisation DARPA in research projects ALP and Ultra*Log. It includes not only run-time engine but some visualization, data management and other tools.
- JACK Intelligent Agents – a Java-based multi-agent platform developed by commerce organisation Agent Oriented Software, Ltd. The platform of the third generation implements Procedural Reasoning Rules and supports Distributed Multi-Agent Reasoning System (dMARS). JACK uses BDI model of agents logics and built-in formal-logical means of agents work planning.

The platforms that have described previously can be all characterised by the following features:

- focused on R&D developments;
- provide very low-level programming tools;
- do not provide any specific methods and tools for dynamic scheduling and resource optimization in real-time;
- requires more efficient tools for industrial use.

The first experience of development of industrial systems has shown that for industrial applications it is required:

- to use agents world framework that encapsulates base classes of demand and resource agents and separate protocols, what enables flexible re-distribution of orders in time and among resource;
- to support large-scale schedules it is required to create a platform that supports a big number of agents that can be very simple but intelligence in work with schedule is provided as the result of their interaction and constant self-organisation and evolution;
- to create specific dispatcher and messaging system to support a big number of agents. It allows hundreds and thousands of agents to work in one thread of operational system.

In the short run the first generation of Magenta multi-agent platform was developed [26-27]. It was originally developed in Object Pascal for modelling applications and at the same time in C++ for Internet applications.

The developed platform included following components:

- *Run Time Multi-Agent Engine* - agents dispatcher designed as an operating system that can transfer control between a large number of agents (up to 100 thousand) and can provide fast transmission (up to 10 000 messages per second);
- *Ontology & Scene Constructor* - ontology constructor allows to create an ontology in the form of semantic networks for the specification of the problem domains, as well as scenes for specifying the situation and the results [28].

Key features of this platform was high performance and use of ontologies for domain knowledge specifications. On this basis a number of first prototypes of industrial systems for transport logistic were developed including the MAS solutions for tankers, taxi and trucks, couriers, train scheduling, the car windows supply and other supply chain management projects [29-36].

But also on the same platform several systems in completely different application areas were set up: smart Intranet system, real time clustering and text understanding systems, an airplane wing collective design, diet management, Internet campaigns optimizing, etc [37-48].

Later on developed multi-agent platforms were united into a one platform in the Java language for building J2EE applications, which has become de facto the industrial standard for many customers.

Let's consider the most interesting examples of the above systems.

4.1 Multi-agent clustering system

Many companies have accumulated large databases with data about customer orders, production and sale of goods, movements of goods in warehouses, hiring and firing employees, etc. In this case, gigabytes of data stored there are "dead weight" as a rule, although they contain a significant hidden knowledge about models of customer behaviour, typical characteristics of orders, patterns of decision-making, etc.

There are many software solutions designed for visualization and analysis of such data, including data mining (extracting knowledge from data), but none of them can run in real time. Besides that the task for hidden patterns revealing is still not resolved. The problem is that: generally these systems require at least prior knowledge and initial classification of the data, for example, studying the age of customers you should decide in advance what age groups (from 16 to 19 years, from 20 to 23 years and so on) will be taken into account. However, the structure of these groups by itself is a subject of research and cannot be fixed in advance. In addition, the structure of these data may vary in real time, for example, when new users come to an Internet store.

In this regard, we have developed the multi-agent approach for data clustering [38] in which the structure of the clusters are reviewed in the course of the new records coming, so we use data sets self-organization, similar as was shown above in adaptive planning.

Each cluster and each record gets his agent, acting on behalf of them. The new record can log into an existing cluster, create a new cluster, joining with the neighbours, or stay out of the cluster structure. For the record it may be better to enter a large or very dense cluster because managers start data analysis and make proposals, including rebates to its participants, from it. In some cases, managers are interested in very large clusters, but in the others - most rapidly growing or changing direction from growth to reduce, etc.

Decisions on such a virtual market every agent takes for himself, guided by the value rules given by the manager and virtual money. When it comes to sales, agent of record receives a percentage of the transaction and can spend virtual money to join one or more clusters. Similarly, if the cluster "lets in" the record (when to both of them this decision is profitable) the other records will be notified about this event. Moreover records have the right to reconsider their decision and leave this cluster, if the value for them has decreased (e.g. cluster expanded and its average density had fallen), exactly like in the truck's logistic, when an order may leave the truck, if the conditions for others have changed.

It is possible that leaving of one record from the cluster immediately can greatly worsen the situation for everyone, and then 4 more records can come out, and then another 16, etc. As a

result the cluster will "crash" and falls apart, but instead a set of other clusters, possibly having a completely different structure will form, that gives extremely interesting results for users.
This approach for data clustering was applied to analyze the Duty Free shops at the Heathrow Airport. Worked through a certain time interval the system found the fact (hidden patterns) that on Friday nights all the passengers of London-Amsterdam flights bought mostly perfume. To the store managers, this fact may indicate that at this time there is simply nothing else to buy, or on the other hand that people need this kind of products and corresponding changes in assortment can significantly increase the volume of sales and as a result the profit of the store with the same volume of rent space.
The developed system was advanced for un-structured data (texts) and applied to other number of tasks and challenges, in particular, to analyze documents of insurance company [42].
During a short period more than 20 thousand documents were analyzed, saved of about 4 man-years work for the company.

4.2 Multi-agent system of text understanding

One of the most complex modern problems, that is so far unsuccessfully solved by methods of traditional computational linguistics, is the problem of text understanding. The scope for such systems apply in the Internet is extremely broad: annotating and materials sorting, intelligent information retrieval (as opposed to the "key words" search), "smart" text editors that understands the edited text, automatic recognition of SMS messages of terrorists, etc.
The developed multi-agent approach offers an absolutely new solution of this problem, using the principles of self-organization and evolution [42]. In this approach, every word in a recognized sentence, and every semantic sense of the word in vocabulary gets his agent acting on its behalf. It should be noted that, for example, the English word "Table" in Oxford dictionary has 18 different meanings, corresponding to different contexts: Table – as a table for placing objects, Table - as the chart in Excel, etc. The purpose of agents words and agents meanings is to establish links between them in the right way, restoring the context of the scene, in this way to construct a semantic network of concepts and relationships (the scene), reflecting the situation presented in the text.
Developed approach gave a chance to create a number of industrial systems for the text understanding.
One of the system designed for automatic abstracting of articles in molecular biology to the world-known database Medline, filled up to 1 million articles on medicine every year. In the first version of the system ontology contains just 140 concepts and relations ("experiment", "organism", "belongs to", "a gene", "chromosome", etc.), but on a sample of 1000 abstracts the right search was up to 85%, although only 25% words in sentences in the abstracts were recognized by the system. That indicates a well-known redundancy of natural language and high sensitivity to the semantic meaning of phrases. This confirms the famous fact: for foreign text understanding in a narrow subject area, knowing the ontology of the world domains and understand only a small part of dictionary, is enough.
Other successful projects was for processing documents of the insurance company mentioned above [42], when it was necessary to process a large array of documents in the area of property, cars insurance, etc. In combination with clustering system, that provided the scenes construction obtained during the recognition of proposals, for few hours of work

the system found out exactly what concepts and relations are most popular in contracts and create samples of contracts that were most responding to the changing customer's requests. Approximately that requires a group of qualified experts and several years of manual work. This approach was also used to develop the system for fax recognition [43], data meta-search in the Internet [44] and other [46-47].

4.3 Multi-agent system of collective wing design

In this project the customer had an industrial system for design of mechanical objects of an airplane wing. The main problem was to check the compatibility of engineering solutions.

The matter of fact of the problem was the high level of dependence of each individual engineer changes. The specialist could make the changes in dimension or physical and chemical properties of the materials or others, which may dangerously affect the adjacent elements of the wing. To conduct inspections of geometric objects a special system for analyzing of the geometric intersection of objects (clash analysis) used, that reveals such conflicts about once every 2 weeks, requiring time-consuming processing.

To solve this problem the multi-agent approach was proposed, where the wing was represented as a semantic network of its elements [22]. In this case, the agent that have changed the wing (e.g. landing-gear), could identify itself to a unit of semantic network of wing and immediately get the information about his neighbours units of this assembly. Comparing with previous inefficient full checks, as it was previously done by the customer, we proposed checks only of the neighbouring elements of changed parts, that reduces processing time.

As a result, the time to verify of all airplane wing units decreased by few times, what shows the high efficiency of the proposed solution.

4.4 Multi-agent system for diet management

In this project, the client worked on a Web portal that offers its users an individual diet tuneable in real time. For example, if a person ate an "extra" cake during the lunch, then he should inform the system about that so the system could reschedule menu for dinner or even breakfast for the next day, by introducing additional constraints, or in the future, offering sports activity to burn the calories, such as an evening ride on the bike.

As a solution of the problem the multi-agent system was proposed [22]. All products in it, diets and users had their agents. Agents of food themselves evolved in combinations, based on user preferences, requirements and compatibility of diet foods (for example, an agent of meat could invite an agent of red wine), taking into account personal national, religious and other characteristics of each user.

Agent of selected diet, for example, with constrain in 2000 calories, checked the resulted menu and if the amount of calories extend beyond 2000, asked the most high-calorie dish to leave the menu. All invited by that dish products were followed. The vacant place was occupied by a smaller calorie dish that provided no conflicts with other dishes. If such conflicts arise, they were resolved on the basis of user preferences so the other dish left or the dish placed before. As in others systems developed leaving of one object caused a chain of changes of the variable for not foreseen length of such wave.

As a result, the menu on the day was built as a self-organizing system, easily changing with new events occurring and deviations from the plan.

4.5 Multi-agent system of optimizing Internet campaigns

The objective of this project was to find in real time the best way of putting company's banners on websites depending on how user clicks are going.

To solve this problem, a multi-agent system was developed, which has been used successfully in a number of marketing companies [48].

If the advertising campaign is successful, the campaign agents and sites are quite satisfied in this solution. If the banner of any site is "clicked" poorely by users, it is necessary to move it to a more suitable site. If the new site placement is occupied, it needs to reconsider the position banners stationed there. Again we have the wave of changing's that can give results by self-organization through the review of decisions of banners and sites during their negotiations.

In marketing agencies in the current time such work is done by a special manager, but in practice it is very difficult to keep track of hundreds or thousands of simultaneous going campaigns to take into account the relationship between sites and campaigns, to monitor all changes in real time, etc.

The developed system can significantly improve the effectiveness of advertising campaigns and reduce the complexity and cost of this process.

5. Industrial applications in transportation logistics (2004-2008)

5.1 Brief survey of existing scheduling methods and tools

In spite of significant progress regarding development of large-scale Enterprise Resource Planning (ERP) systems, opportunities of the enterprises on development of adaptive scheduling systems remain very limited.

Traditionally the ERP systems include subsystems of orders collection, large databases for orders and resources, accounting and reporting subsystems and a lot of other components. However in these systems batch or manual scheduling of orders is supported, that was already discussed above. The schedulers offered by such large companies, as SAP, Oracle, Manugistics (it was recently bought by JDA), i2, ILOG and others usually realize various versions of Constraint programming methods, based on combinatory search of options in depth, for example, a method of branches and borders [49].

To reduce the number of options considered in combinatorial search new methods consider various heuristics and meta-heuristics (the term "heuristics" is usually understood as a set rules, defining what option is the best, and "meta-heuristics" means a rules to choose heuristics), allowing to provide good decisions for reasonable time and reducing search iterations [50].

Well-known heuristics in optimization are "greedy" methods. In such methods the decisions are taken by a choice of the best of options on each step, and once made decision is never reconsidered. Various other methods of local optimization are more complex, where initial solution which then is improving by local changes can be changed randomly or in some pre-defined way, if the good final solution is not reached, and the process repeats many times.

As one of the most known meta-heuristics we can consider Simple Local Search Based Meta-heuristics (SLSBM) – local optimization meta-heuristics. Here one of heuristics can implement casual choice of one candidate from the list of the best, another one - looking forward or randomizing of criteria, etc. One more meta-heuristics developing recently is Simulated Annealing which is based on modeling of process of cooling. This method represents an expansion of methods of local optimization in which many options could be formed on each step and it is possible to consider not only the best options, but also some worsening decisions with the probability calculated as function from some attribute, analogue of temperature.

The main idea of becoming more and more popular Tabu Search is the usage of history of decisions of local optimization when some investigated options are becoming prohibited (tabu) and consequently they are not considered on a following step.

One more new meta-heuristic is Ant Search, in which the behaviour of the ants, getting food is modelled. The success of one ant in getting of "food", i.e. taking of some decision, during some time prompts other ants a correct direction, but in due course signs on this successful direction "fade". In last period of time also many other meta-heuristics become more and more popular inheriting physical or biological concepts. Another example here is Adaptive Memory Programming method which inherits the use of common memory of decisions. In last developments researchers apply mixed miscellaneous meta-heuristics, in which several parallel algorithms are acting, and each of them suggest their own decision.

At the same time, even in view of considered methods and tools of local search of variants require greater expenses of memory and time for producing schedules. For example, producing of the optimum plan for the large transport company in one of available software packages takes about 8-10 hours. During this time the volume of orders can be essentially changed that will require to start planning all over again. At the same time the technology for planning in real time remain rather primitive, and an opportunity of flexible adaptation on the base of happening events refer mainly to an opportunity of manual plans updating. As a result, according to the estimations of transportation logistics experts, the created schedules are feasible only on 40 %, which compels many large transport companies still to contain staff of very skilled and expensive operators on planning and to carry out time-consuming manual or semi-automatic planning.

This, certainly, is promoted by both high complexity and labour intensity of planning, unpredictability of dynamics of a stream of events, by requirements of an individual approach to each order and resource, constant change of conditions of functioning of the enterprise forced by clients and competitors, and also necessity of the account of many other very specific features in each business. For example, the operator of trucks fleet should constantly keep in a head preferable time windows of loading/unloading of warehouses and shops, conditions of contracts with clients, rules of compatibility of cargoes, experience of the concrete driver and even such specific facts, that the certain road became impassable for greater wagons because of rank branches of trees.

As a result many of existing classic methods of planning and resource optimization have a number of very important limitations in practice:

- Do not consider complexities of the modern business operating in thousand of orders and resources, supporting interdependency between all operations, reflecting and balancing interests of many parties involved, having a lot of their own features;
- Do not provide opportunities for adaptive planning in real time which requires dynamic event-driven conflict solving in schedule;
- It's usually supposed that all orders and resources are identical but in practice they all have their own individual criteria, preferences and restrictions, which can change during the system work (service level, time of delivery, costs and profits, risks of delivery, etc);
- Do not give the tools for the acquiring knowledge which are specific to every enterprise, influencing quality of provided schedules;

- Do not allow an operator to explain and adjust decisions easily and in convenient way.

All this not only reduces productivity and efficiency of existing methods and tools but also in practice in many respects stops their use.

To provide opportunity to build adaptive schedulers on the top of existing ERP systems and eliminate the specified lacks in scheduling mobile objects multi-agent approach was offered which is based on the RDN concept [18-20].

It helps significantly to increase quality and efficiency of scheduling and make results more clear, understandable and adjustable for end-users and also to reduce delivery time.

5.2 Architecture of systems for adaptive scheduling

To implement the developed approach in scales of the large enterprises the architecture of system for adaptive scheduling is offered, it's presented on Fig. 7.

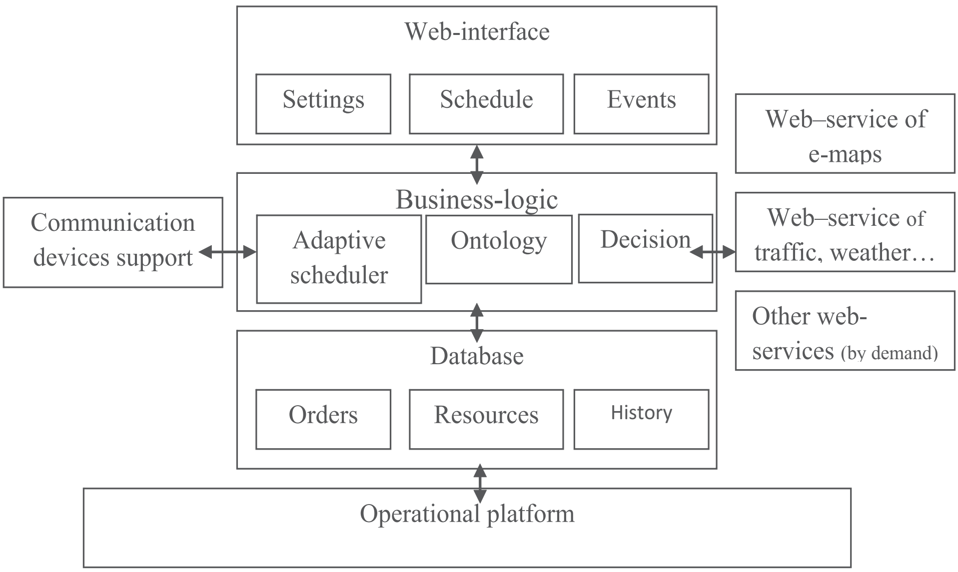


Fig. 7. Architecture of Systems for Adaptive Scheduling

Let's consider in detail the basic components of the given architecture.

This architecture implements a three-tier architecture including servers for web-interface, business-logic and databases, and also can get the operative information from external web-services and cooperate with communication devices of users (for example, drivers).

Web-interface layer of the system gives an opportunity to make settings and process orders and resources of the enterprise, etc. Through a web-interface the system operator can see the current schedule of system formed by the adaptive scheduler, in the form of Gantt chart (the schedule on each resource) or in a tabulated mode, from the side of both orders, and resources. At last, one more important component is for a display and processing of events of different type which can be transferred to scheduling manually or automatically. It is important to note, that internal and external events processing report is available for a user, that allows to explain the decision making logic of the system to an operator.

If necessary a user can be provided both with a desk-top interface for more convenient work at the local machine using web-start technology.

The layer of business-logic actually provides a reaction to events, adaptive scheduling and delivery of results. A basis of this part of the system is the adaptive scheduler constructed using the described above multi-agent approach. For each problem there can be developed a new scheduling engine, but at the same time there are certain opportunities of adaptation of existing "engine" according to new requirements by ontology configuration. The tools of ontology support allow to describe objects and attitudes of a problem domain, and also the scene describe current position of resources and orders in a transportation network at the moment of time. On this basis the rules of decision-making are formed, which can be switched on and off, be modified or adjusted by the user. The logic of decision-making is supported by the set of components, allowing to carry out calculations of distances or costs, which are specific for transport logistics, and other functions.

The database layer allows to save the information of concrete orders and resources, and also history of changes of the schedule.

The adaptive scheduling system can integrate itself with client platform or to use components of the offered platform including tools of a security control and management rights of users, provide visual reports, etc.

On the basis of given representation of architecture there can be developed the solutions on adaptive scheduling of resources for enterprises of various domains, considering the specific requirements and restrictions.

The examples of industrial application of the described approach and solution architecture are given below.

5.3 Tankers scheduling system

This system is used for management of large-capacity tankers, carrying out transcontinental transportations of oil. Everyday a company, carrying out up to 70 % of world transportation in a considered class of vessels, gets 10-15 inquiries about oil transportation [51]. Operators of the company should make the analysis of a situation in real time, sometimes even on the phone, to analyze situation, provide all economic calculations and make a decision, on what tanker it is necessary to execute the order.

At the same time it is necessary to keep constantly in a head arrangement and traffic schedules of own vessels, and also positions of competitors, count routes of traffic, consider features of passage of Suez canal if it is necessary (time for partial unloading of oil is required), consider, what ships can enter into what ports, where and when it is better to refuel the tanker, what are weather conditions, etc. To solve this problem the system of adaptive scheduling has been developed, which was integrated with a data management system. Due to small horizon of scheduling (the number of orders planned forwarding advance), in this system the arrival of a new event entails long chains of possible changes including up to 7 exchanges of orders between tankers. Thus, the new order can affect changes of a lot of tankers and even alteration of contracts with a number of clients.

Coordinator of the AgentLink European Multi-Agent Systems Roadmap Michael Luke from Southampton University (UK) explains [6] the benefits of agents in the developed system as: "By modeling each tanker as an individual agent is achieved the ability to see the options and ability to respond rapidly to emerging events in real time."

The system was developed for regular operations allowing to simulate orders allocations and schedule chosen orders more effectively. The cost of one day of idle time of such tanker is about $100,000. That allows to estimate economic benefit of the system implementation.

At the same time, the opportunity to take and formalize valuable domain-specific knowledge of operators which are necessary for decision-making turned out also very important for the customer for decreasing of human factor influence.

5.4 Corporate taxi scheduling system

This system allows the company [52], to schedule adaptively about 13 thousand orders a day at presence of several thousand cars with GPS, up to 800 from which are always on the road.
The company basically works through call centre in which 130 operators simultaneously accept calls. The company tries to provide an individual approach to each client, allocating only cars of the necessary class or a class above, with well-reputed driver, give on demand the car for disables, with the trailer, for smoking passengers, for transportation of animals, etc. The drivers work in the company as freelancers, deciding themselves what number of days and hours per week (with some restrictions) to work, renting cars at the company. At the same time they can come to work at any time. The drivers have handheld computers which allow the driver to appear on "radar" of the system when starting to work.
At occurrence of the new order the system automatically finds the best car and preliminary reserves the order. On the average the submission of the car takes about 9 minutes. From the moment reception of the urgent order, the system continues to redistribute orders for concrete time continuously in view of appearing of new resources, and does not make of the final decision till dynamically defined moment when it is necessary to send the car to client.
It is important to note that the system first tries to maximize profits. However, it also takes into account several other criteria that are important to the business. For example, when selecting from two options are roughly equal, the system gives the order to the driver, who have not received orders, thus avoiding claims to dispatchers, who previously could give a good order "their" driver. In addition, when a driver finishes, the system picks up his orders on the way home, which increases the earnings of drivers and reduces employee turnover.
The first month of implementation the number of sold orders increased by 7% with the same fleet, now 97% of all orders taxis are scheduled automatically, without the participation of managers; in 3,5 times (up to 2%) orders executed at the wrong time decreased, for 22.5% the taxi idle run, every taxi now serves on two additional trips per week at the same time and fuel costs, which is reflected in the increased profitability of each machine at 5%; taxi order is 40% faster, while the training of new operators decreased by 4 times, the website working more effectively where it comes from the 16% of orders the company.
This system has received a national award for best British innovative solution in the business in 2009, and was shown on Russian television in the "Times" program on Channel One.

5.5 Truck scheduling system

This system provides the truck scheduling for world famous networks of supermarkets. Among the transported goods there are food stuffs and drinks, including the frozen products, household electronics, clothes, etc.
The level of orders in corporate network – about 4 000 a day, the fleet of the company includes about 300 trucks of various volume, and a number of them is equipped by the additional equipment (refrigerators, etc.), the delivery network includes about 600 geographical locations all over the country.
The complexity of a problem in many respects is connected with presence of warehouses of intermediate storage, necessity of splitting of greater orders for some trips and, on the

contrary, consolidations of small orders of different volume, requirements of compatibility of cargoes, different opportunities of acceptance of trucks in different warehouses, etc.
For solving this problem the scheduling system was developed [53]. It automates all main steps of orders execution: from orders receiving and adaptive splitting and consolidation, routing and scheduling – to reports making. This system turned out to be the most difficult, where architecture of the virtual market includes a lot of agents acting together and proactively.
In particular, the orders are dynamically broken to sub-orders that are then consolidated in groups, and the trips also are formed dynamically from groups, and they, in turn, are planned for trucks. If the order has been splitted unsuccessfully, and it was not possible to plan good trips, it is made re-splitting and routing and scheduling begins anew. The big number of active agents (tens and hundred thousand) has led to necessity of application of more developed mechanisms of scheduling of agents, when only the most perspective agents get activity, competing with each other.
In present time the system is on implementation step, and decision making logic tuning is taking place. Before the deployment started the operators planned trips manually on the basis of numerous Excel tables. In this connection a lot of time was spent for adjustment of initial data in which there were many issues, including different versions of names of the same warehouse, etc.
It's expected with the system introduction we'll get not only significant economic benefit of more effective scheduling resources, but also the number of operators will be essentially reduced.

5.6 Car rental scheduling system

Client has about 100 stations and each of them has on the average up to 150 cars of different classes. Customers can order the car by phone, directly come to station or book car via the Internet.
For convenience of clients it is possible to agree about delivery of the car during necessary time to the necessary place. But then it is required to send the car with the driver which can work at stations as in the certain days, and overtime. Also it's necessary to send drivers, to take away cars from clients, therefore in some cases it is necessary to send several drivers in one car, someone will bring the car to the client, and someone will take away the used cars.
For solving the problem the adaptive scheduling system was developed [54], that allows to re-schedule operatively the delivery of cars for new-coming orders and also in case of different kind of events and make schedule for drivers who bring or take cars. At the same time the system also addresses to an e-map and shows drivers recommended routes of traffic, and also sends them in real time all other necessary instructions.
Now development and testing of system on real data is finished and its expansion at first five stations is begun, up to the end of current year introduction in all other stations is expected.
The economic benefit of system consists in distributing cars in view of a situation for the whole country and to estimate more precisely, from what station it is necessary to give the car.
Reduction of the total number of cars in the network up to 10% and savings on fuel expenses and salaries of drivers are expected as the number of the superfluous trips, involved drivers and amount of overtime will be reduced.

6. New generation of MAT platform (2009-2010)

The results achieved help to generalize first experience and design second generation of platform to increase quality and efficiency of MAT solutions, cut down costs and time of development, minimise risks in industrial projects.
Key features of new platform are presented in Table 2.

№	Key Feature	Benefits	Usage examples
1.	Agent design supports full life cycle: perception, planning and execution.	Support of real time solutions and applications.	Dynamic dispatching, planning and optimization of mobile resources.
2.	Virtual market based on non-linear thermodynamics.	Speed up or slow down negotiations – to control self-organization.	Improvement of quality and efficiency of real-time scheduling.
3.	Adaptive p2p networks of real-time schedulers demonstrating co-evolution of self-organized systems.	Openness, high performance, scalability and reliability of enterprise-ready system.	Supply chain management, for example, factory and transport plans coordination.
4.	Dynamically formed bottom-up ontologies constantly renewed in the process of interaction with user.	Opportunity to fill the knowledge base on the fly without reprogramming.	In the process of interaction with a driver the system learns about such events as snowfall, road closure, etc.
5.	"Collective intelligence" support when every staff member actively participates in enterprise management.	Increase of business performance, efficiency, productivity and competitiveness.	Taxi driver using his mobile phone can signal about the groups of potential passengers.
6.	Sophisticated interaction with the system to find the problem solution.	Intellectualization of dialog with the user.	Allows the user to play with solution and adjust it on the fly accordingly.
7.	Support of work in case of uncertainty or errors in initial data.	Lack of data or errors is not a restriction for system to work.	Stability and reliability of results in case of incorrect data.
8.	Support of parallel processing.	Significant increase of performance	Planning and scheduling of large amount of resources.
9.	Combination of work in real-time with classical batch algorithms of resource planning and scheduling.	Improvement of planning quality when orders and resource are known in advance.	Strategic long-term factory scheduling.

Table 2. Key features of second generation of MAT platform

At present time we are developing a new version of such a multi-agent platform that includes the following components:

- Agents dispatcher providing control of agents states;
- Communication framework for message exchange and support of agents communication protocols;
- Framework for scene objects data models allowing to extract, make changes and save data efficiently;
- Agent design components including creation and deletion of agents, agents logic support, etc;
- Modules for dynamic scheduling, including creation of schedules, basic classes of agents, negotiation protocols, etc;
- Unser interface components including work with mobile devices;
- Basic toolset for ontology and scene management;
- Components for visualization of results (graphics and diagrams);
- Agents messages /decisions log.

For design objectives it is proposed to create the following components:

- Constructor of agents worlds, including agents classes, roles and their interaction protocols;
- Executing system for organization of parallel processing that supports multi-core processors, multi-processor systems and multi-service applications;
- Modelling system for debugging of agents worlds and methods of their interaction providing tools for generation of data, work control and showing results;
- Component of visualisation of multi-agent system behaviour;
- Agents world debugger;
- Commander as a subsystem for dialog with agents;
- Events generator for testing and playing scenarios;
- Real-time environment allowing to convert working world model into the final real-time application;
- Libraries of additional components for development specific applications.

New platform is under developments in J2EE and .Net environments.

7. Examples of new industrial applications (2009-2010)

7.1 Multi-agent inter-city transportations system

This project is created for the customer who has central office in Moscow and more than dozen of branches throughout the country. The company manages transportations by using of own fleet of 100+ trucks, equipped with GPS devices, and external carriers.

Monthly company receives hundreds and thousands of orders transport electronics, food, drinks and other products. To maximize effectiveness of truck utilization for trips from Moscow to regions and backwards it is necessary to find backhaul loads as well as to take into account contract details, minimize delivery time, possible risks and penalties, etc.

To solve this complex problem a multi-agent system that supports coordinated fleet scheduling by managers from central and branch offices was developed. While planning new order manager can simultaneously see new trucks in his region, find backhauls for them or can use these trucks to deliver his cargo. The system allocates orders to most suitable available trucks and if there no such trucks – it figures out conflicts with already

allocated cargos and tries to move orders or reallocate resources by shifting, dropping or swapping orders. The system reschedules interdependent operations in case if deviation was found between planned and actual states of resources.
The system is able to automatically monitor and control business processes of order receipt, cargo loading/unloading by contacting with driver through mobile phone, as that driver has to give signals of certain operation beginning and end.
The System is integrated with 1C software to prepare related financial documents and also generates required business reports of each division effectiveness for managers and directors in real time.

7.2 Multi-agent taxi management system

Basing on the requirement of Moscow taxi company a specific low-budget taxi management system has been developed which is able to use most popular Nokia-like models of mobile phones to communicate with drivers through special Java applet.
Previously in this taxi company the whole fleet was split into groups, managed by separate dispatchers. As a result there were many situations when the taxi, assigned to one dispatcher, had to go from the northern part of Moscow to the southern part, while taxi assigned to another dispatcher goes in opposite direction, what considerably reduced effectiveness.
Another important problem was frequent taxi delays without customer notification. It was especially essential for airport transfers. Furthermore taxi company was using portable radio set which was not convenient for fleet of 500 taxis. Drivers, in their turn, always suspect dispatchers in giving most profitable orders to "favourites". Finally, it was necessary to provide individual approach to every customer, including corporate customers.
The first version of the system was created which received practical approval on 50 cars within 3 months. Currently possibility of basic system version further functionality development and its implementation for the whole fleet of cars is being considered.
The most interesting directions for further development relating to taxi area are splitting by passengers, usage of drivers as an "intelligent sensors" and other very modern industry-specific possibilities for taxi business which become available by multi-agent technology.

7.3 Multi-agent system of airport ground-services management based on RFID technology

This project was implemented together with the University of Cologne for German Ministry of Economics and Technology within industrial consortium including Airbus, Fraunhofer institute, AutoID company RFID-tags manufacturer and BLW catering company.
The project was aimed at development of multi-agent system for modelling airport ground service operations such as food delivery on board, air stairs bringing, pickup service and luggage delivery, aircraft cleanup, defreasing and some other services.
The projects' feature was to discover the possibility of RFID tags integration into the process of ground service management which allows to find the location of any airport resource and provide adaptive planning in real time to increase the quality and reduce costs of airlines service, increase passengers' service level and reduce aircrafts idle time and etc.
According to the projects' results it was concluded that it is possible to improve passengers' service level, reduce cost and time of airlines service, reduce risks of flight delays and improve some other substantial indexes of aircraft logistics.

This solution was designed as a new generation of multi-agent systems development built as a network of cooperating schedulers each of which is responsible for its own service but at the same time coordinates tasks in close cooperation with other services.

7.4 Multi-agent real-time factory scheduling system

Developed system is made for factory workshop resource scheduling and optimization in real time including workers, equipment, materials and other.

This system was created for a large-scale airspace enterprise and can be applied for any works, that require individual approach to each production unit, nomenclature of which is constantly changing, have small production batches, require high workers qualification, have to deal with multiple unexpected situations and require high efficiency and flexibility in product realization.

To solve this problem a solution was created which allows to represent a schedule as a network of operations where agent of each operation knows who is on the right or left. The system allows easily to change the plan in case of events arising. At the same time it's possible to use different planning strategies from "just-in-time" to "as soon as possible" or "as cheap as possible", etc. It's expected that system implementation will allow to increase workshop efficiency by 15-20%.

Further system development includes adaptive network of workshop schedulers that, working separately and autonomously, will have an opportunity to compete and cooperate according to P2P scheme using enterprise service bus. In this case co-evolution and of self-organization of real time enterprise resource management systems will be demonstrated for the first time.

7.5 Multi-agent cargo planning system for International space station

This project is made by order of one of the biggest world-scale airspace corporations and is aimed at cargo transportation for International Space Station (ISS). User can build flights program, enter new launches of a spaceship, change type of spaceships and start-up time and enter other events that can change possible ways for cargo delivery.

Cosmonauts have their needs like need for water and air, fuel and food, equipment maintenance and repair, etc. As the result of system work cargo deliveries can be dynamically rescheduled, for example, amount of fuel and water, products for cosmonauts' live support and some other goods can be reallocated between spaceships flights.

At the moment this solution is developed and is at the stage of delivery.

7.6 Multi-agent system of satellites swarm management

Swarm of satellites management is one of the most leading projects based on multi-agent technologies being developed.

In this project company provides the platform for modeling cooperation in the group of orbital space satellites, fulfilling remote Earth sensing on behalf of EMERCOM and other parties. If one satellite looses an object or discovers its new features it should be investigated in more detail thus starting the cooperation with other satellites that changes their plans.

This development is focused on designing intelligence of new airspace satellites, operating as self-organized organisms and able to evolve in time due to their ability to share and reallocate tasks, collect knowledge and learn from experience.

7.7 Multi-agent scheduler of personal tasks for mobile users

This project is aimed at creation of personal tasks scheduler, which can operate or be accessible on mobile phone.

In this system on base of the ontology editor user can set up templates with sequences of actions, that then can be uploaded to the personal plan taking into consideration all semantic interdependencies between them, overlap with existing tasks, shift them, and, finally, sequence in the most convenient way for the user, constantly changing as new events occur.

With the help of ontologies used as scenario templates for tasks specification any operations chain can be managed for example for companies' business-processes control, government service delivery, taking a medicine and etc.

If necessary user can "download" to mobile device the templates that seem to be useful for business or private life situations, sport and other activities that will be planned in accordance with set up preferences and restrictions and adjusted in real time. For example if a user is at the meeting and it's time for him to go to the airport, his agent by analyzing user location with the help of GPS device, traffic jams and his current schedule finds out that user can be late and sends him a context–driven message containing offer to finish the meeting and order a taxi. Any events can overlap with the uploaded templates, causing an adaptive rescheduling of the user tasks by their shifts and drops in accordance with user preferences, interdependent tasks and etc.

In the first system version only single user tasks can be planned but since the main schedule is kept on the server later on it will be possible to collectively plan employees' work.

At the moment this project is at the stage of commercial system prototype development.

8. Conclusion

This paper gives an overview of research and application works of industrial multi-agent systems, designed and developed in Samara school of multi-agent technologies.

The main result of these studies is that the multi-agent technology allows to solve complex problems and create enterprise-ready systems based on the fundamental principles of self-organization and evolution. The developed system provides the quality and efficiency of results, reduce costs and risks and minimize dependence of human factors.

The developed methods and tools are now battle-proved, generic and applicable to a wide range of complex problems including clustering and text understanding, real time logistics, etc.

The results show wide perspectives for developing new systems for solving complex problems for many applications in real time.

9. References

[1] M. Wooldridge. An Introduction to Multiagent Systems. John Wiley and Sons Ltd, 2002, Chichester, England - 340 pp.

[2] Городецкий В.И., Грушинский М.С., Хабалов А.В. Многоагентные системы // *Новости искусств. интеллекта*. – 1998. - №2. - С. 64-116.

[3] Хорошевский В.Ф. Методы и средства проектирования и реализации мультиагентных систем // *Матер. семинара "Проблемы искусственного. интеллекта"*. - ИПУ РАН, 1999.

[4] Тарасов В.Б. Агенты, многоагентные системы, виртуальные сообщества: стратегическое направление в информатике и искусственном интеллекте // *Новости искусственного интеллекта*.–1998.- № 2.-С.5-63.
[5] Stefan Poslad. Ubiquitous Computing. Smart Devices, Environments and Interactions. – John Wiley and Sons Ltd, 2009 – 473 pp.
[6] Agent Technology: Computing as Interaction. A Roadmap for Agent Based Computing - http://www.agentlink.org/roadmap/index.html
[7] http://www.lostwax.com
[8] http://www.whitestein.com/
[9] http://nutechsolutions.com/index.asp
[10] http://www.magenta-technology.com
[11] Bonabeau E., Theraulaz G. Swarm Smarts. What computers are learning from them? // *Scientific American*. – 2000. - Vol. 282. - N 3. – P. 54-61.
[12] Thompson, *J. Ant Colony Optimisation* // School of Mathematics - Cardiff University, SWORDS, 2004.
[13] A. Koestler. The Ghost in theMachine. Arkana, 1990.
[14] Brussel H. van, Wyns J., Valckenaers P., Bongaerts L., Peeters P.: Reference architecture for holonic manufacturing systems: PROSA – Computers in Industry, 37 (3) (1998) 255 – 274.
[15] V.Marek. *Preface to Proceedings of 5-th International Conference on Industrial Applications of Holonic and Multi-Agent Systems for Manufacturing HoloMAS 2009*, Linz, Austria // Springer, 2009, pp. V-VII.
[16] P.Leitao. ADACOR:A Holonic Architecture for Agile and Adaptive Manufacturing Control. – *Computers in Industry* 2006, 57(2), 121-130.
[17] Rzevski G., Skobelev P. Emergent Intelligence in Large Scale Multi-Agent Systems // *Education and Information Technologies Journal* - Issue 2, Volume 1, 2007.
[18] Скобелев П.О. Открытые мультиагентные системы для оперативной обработки информации в процессах принятия решений. – *Автометрия*, 2002, №6, с.45-61.
[19] Виттих В.А., Скобелев П.О. Мультиагентные модели взаимодействия для построения сетей потребностей и возможностей в открытых системах. – *Автоматика и Телемеханика*, 2003, №1, с. 177-185.
[20] Виттих В.А., Скобелев П.О.. Метод сопряженных взаимодействий для управления распределением ресурсов в реальном масштабе времени // *Автометрия*, 2009, №2, стр. 78-87.
[21] Андреев В.А., Виттих В.А., Батищев С.В., Скобелев П.О. и др. Методы и средства создания открытых мультиагентных систем для поддержки процессов принятия решений. – *Известия Академии Наук. Теория и системы управления*, 2003, №1, с. 126-137.
[22] Rzevski G., Skobelev P., Andreev V. MagentaToolkit: A Set of Multi-Agent Tools for Developing Adaptive Real-Time Applications - *Proceedings of 4-th International Conference on Holonic Approach and Multi Agent Systems (HoloMAS 2007)*. – Germany, June 2007.
[23] Абрамов Д.В., Андреев С.В., Симонова Е.В., Скобелев П.О.. Разработка средств построения и использования онтологий для поддержки принятия решений // *Труды VII-ой международной конференции «Проблемы управления и моделирования в сложных системах»*, Самара 27 июня - 1 июля 2005. – СНЦ РАН, 2005, стр 435-440.

[24] Андреев В., Андреев М., Батищева Т., Олейников А., Скобелев П., Чевелёв А."Конструкция агента в системах оперативного планирования и принятия решений" // *Труды VII-ой международной конференции «Проблемы управления и моделирования в сложных системах»*, Самара 27 июня-01 июля 2005. – СНЦ РАН, 2005, стр 414-420.

[25] UK Patent Application No. 101516 – Managing a Virtual Environment – authors: G Rzevski, P Skobelev.

[26] Iwkushkin K., Minakov I., Rzevski G. and Skobelev P.: Magenta Multi-Agent Engine for Desktop Applications. *Proceedings of the 3rd Intern. Workshop on Computer Science and Information Technologies CSIT'2001*, Ufa, Russia, 21-26 September 2001. - Ufa State Aviation Technical University - Institute JurInfoR-MSU, Vol. 1: Regular Papers, 2001, 81-89 pp.

[27] Batishchev S., Iwkushkin K., Minakov I., Skobelev P. and Rzevski G.: Magenta Multi-Agent Systems: Engines, Ontologies and Applications. *Proc. of the 3rd Intern. Workshop on Computer Science and Information Technologies CSIT'2001*, Ufa, Russia, 21-26 September 2001. - Ufa State Aviation Technical University - Institute JurInfoR-MSU, Vol. 1: Regular Papers, 2001, 73-80 pp

[28] Andreev V., Iwkushkin K., Minakov I., Rzevski G., Skobelev P.: The Constructor of Ontologies for Multi-Agent Systems. In *3rd International Conference 'Complex Systems: Control and Modelling Problems'*, Samara, Russia, September 4-9 2001, 480 – 488.

[29] Иващенко А.В., Ивкушкин К.В., Никифоров В.А., Скобелев П.О. Мультиагентная система поддержки управления распределением ресурсов в проектах предприятий // *Труды VII-ой Международной конференции «Проблемы управления и моделирования в сложных системах»*, Самара Самарский научный центр РАН, 27 июня-01 июля 2005, с 428-434.

[30] Rzevski G, Skobelev P, Batishchev S, Orlov A.: A Framework for Multi-Agent Modelling of Virtual Organisations. In Camarinha-Matos, L M and Afsarmanesh, H (eds), *Processes and foundations for Virtual Organisations*, Kluwer Academic Publishers, 2003, pp. 253-260.

[31] Rzevski G., Skobelev P., Korablin M.: Multi-Agent Models of Networked Organisations. *Proc. of Intern. Workshop "New Models of Business: Managerial Aspects and Enabling Technology"*, School of Management of Saint Petersburg State University, Russia, June 28-29, 2001, pp. 46-49.

[32] Батищев С.В., Орлов А.Ю., Скобелев П.О. Метод push-pull взаимодействия при составлении и адаптации краткосрочного плана производства в условиях высокой динамики спроса. // *Труды VII Международной конференции «Проблемы управления и моделирования в сложных системах»* Самара СНЦ РАН, 27 июня-1 июля 2005 с.397-402.

[33] Andreev M., Rzevski G., Ivashchenko A., Skobelev P., Shveykin P., Tsarev A. Adaptive Planning for Supply Chain Networks - *Proceedings of 4-th International Conference on Holonic Approach and Multi Agent Systems (HoloMAS 2007)*. – Germany, June 2007.

[34] Basra R., Lü K., Rzevski G., Skobelev P., Resolving Scheduling Issues of the London Underground Using a Multi-Agent System. *Lecture Notes in Artificial Intelligence* (Subseries of Lecture Notes in Computer Science) 3593, 188-196, 2005.

[35] Rajveer Basra, Kevin Lü, George Rzevski, Petr Skobelev. Resolving Scheduling Issues of the London Underground Using a Multi-Agent System – *Journal of Engineering School* , London, 2003.
[36] Rzevski G., Himoff J., Skobelev P. Magenta Technology: A Family of Multi-Agent Intelligent Schedulers. *Proceedings of Workshop on Software Agents in Information Systems and Industrial Applications (SAISIA)*. - Fraunhofer IITB, Germany, February 2006.
[37] Андреев В.В., Абрамов Д.В., Карягин Д.В., Минаков И.А., Скобелев П.О. Разработка семантического Интранет компании на основе конструктора онтологий // *Труды 7-ой Международной конференции по проблемам управления и моделирования сложных систем*, Самара, - июня 2005. – Самара: СНЦ РАН, 2005. С.421-427.
[38] UK Patent Application No. 304995 – Data Mining – authors G Rzevski, I Minakov, P Skobelev.
[39] Вольман С.И., Карягин Д.В, Минаков И.А., Скобелев П.О. Разработка системы нахождения бизнес правил с использованием кластеризации на примере данных логистических компаний // *Труды VII Международной конференции « Проблемы управления и моделирования в сложных системах»*, Самара Самарский научный центр РАН, 22 июня-28 июня 2005, С.409-413.
[40] Rzevski G., Skobelev P., Minakov I., Volman S.. Dynamic Pattern Recognition Using Multi-Agent Technology. - *Proceedings of World Science and Engineering Academy and Society Conference (WSEAS 2007)* - Dallas, Texas, USA, March 2007.
[41] Minakov I., Rzevski G., Skobelev Petr, Volman S.. Creating Template Contract Documents Using Multi-Agent Text Understanding and Clustering in Car Insurance Domain. - HoloMAS 2007, pp.361-370.
[42] UK Patent Application No. 305634 – Automated Text Analysis – authors: G Rzevski, I Minakov, P Skobelev.
[43] Минаков И.А., Скобелев П.О., Томин М.С. Мультиагентная система интеллектуальной обработки факсов, используемых для обмена бизнес-данными. // *Труды VIII Международной конференции «Проблемы управления и моделирования в сложных системах»* Самара Самарский научный центр РАН, 24 июня- 28 июня 2006 с.510-515
[44] Kanteev M., Minakov I., Rzevski G., Skobelev P.. Multi-Agent Meta-Search Engine Based on Domain Ontology. - *International Workshop "Autonomous Intelligent Systems: Agent and Data Mining" (AIS-ADM 2007)* - St. Petersburg, Russia, June 3-5, 2007.
[45] Скобелев П.О. MyAgent Framework: мультиагентный подход к разработке социальных сетей с эмерджентным интеллектом. // *Труды IX Международной конференции «Проблемы управления и моделиро-вания в сложных системах»* Самара Самарский научный центр РАН, 22 - 28 июня 2007 с.511-527.
[46] Минаков И.А., Скобелев П.О. Набор инструментальных средств для интеллектуализации Интернет-порталов и социальных сетей // *Труды IX Международной конференции «Проблемы управления и моделиро-вания в сложных системах»* Самара Самарский научный центр РАН, 22 - 28 июня 2007 с.528-534.

[47] Вольман С.И., Кантеев М.К., Минаков И.А., Скобелев П.О. Разработка системы проактивных предложений релевантного клиента в режиме реального времени на основе анализа действий пользователя Интернет-порталов // *Труды IX Международной конференции «Проблемы управления и моделирования в сложных системах»* Самара Самарский научный центр РАН, 22 - 28 июня 2007 с.543-548.
[48] Вольман С.И., Минаков И.А., Скобелев П.О., Якушин А.В. Разработка мультиагентной системы поддержки принятия решений для оптимизации хода рекламных кампаний в сети Интернет. // *Труды IV Международной конференции « Проблемы управления (МКПУ-IV)* (Январь 26-30, 2009). – ИПУ РАН, 2009, стр. 1628 - 1638.
[49] Handbook of Scheduling: Algorithms, Models and Performance Analysis. Edited by J. Y-T. Leung // Chapman & Hall / CRC Computer and Information Science Series. – 2004.
[50] Stefan Vos. Meta-heuristics: The state of the Art. // Local Search for Planning and Scheduling. Edited by A. Nareyek // *ECAI 2000 Workshop*, Germany, August 21, 2000 // Springer-Verlag, Germany, 2001.
[51] Himoff J., Skobelev P., Wooldridge M. Magenta Technology: Multi-Agent Systems for Ocean Logistics – *Proc. of 4-th Intern. Conf. on Autonomous Agents and Multi Agent Systems (AAMAS 2005)*. – Holland, July 2005.
[52] Glaschenko Andrey, Ivaschenko Anton, Rzevski George, Skobelev Petr Multi-Agent Real Time Scheduling System for Taxi Companies // *Proceedings of AAMAS 2009 8th International Conference on Autonomous Agents and Multiagent Systems* 10–15 May, 2009, Budapest, Hungary – pp. 29 – 36.
[53] Vyacheslav Andreev, Andrey Glashchenko, Anton Ivashchenko, Sergey Inozemtsev, George Rzevski, Petr Skobelev and Petr Shveykin Magenta Multi-agent Systems for Dynamic Scheduling // *Proceeding of the First International Conference on Agents and Artificial Intelligence ICAART 2009* (Porto, Portugal 19-21 January, 2009) – pp. 489 – 496.
[54] Himoff J., Rzevski G, Skobelev P. Magenta Technology: Multi-Agent Logistics i-Scheduler for Road Transportation – *Proceedings of 5-th International Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2006)*. ¬– Japan, May 2006.
[55] Skobelev P., Glashenko A., Grachev I., Inozemtsev S.. MAGENTA Technology Case Studies of Magenta i-Scheduler for Road Transportation - *Proceedings of 7-th International Conference on Autonomous Agents and Multi Agent Systems AAMAS 2007* - US, Hawai, May 2007.
[56] Yankov I.A., Skobelev P.O, Shibanov S.V., Shashkov B.D., Multiagent approach for planning and resource allocation in dynamic heterogeneous networks // Dynamics of heterogeneous systems: conference proceedings / Penza, State University, 2009. Issue 1. pp. 22-37.
[57] Andeev S., Rzevski G.A., Shveikin P., Skobelev P.O., Yankov I.A. A Multi-Agent Scheduler for Rent-a-Car Companies Applications – *Proc. of Intern. Conf. of HoloMAS 2009* – Linz, Austria, September 2009. p. 305-314.

*Edited by Faisal Alkhateeb,
Eslam Al Maghayreh and Iyad Abu Doush*

A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents. Multi-agent systems can be used to solve problems which are difficult or impossible for an individual agent or monolithic system to solve. Agent systems are open and extensible systems that allow for the deployment of autonomous and proactive software components. Multi-agent systems have been brought up and used in several application domains.

ISBN 978-953-51-5991-9

Photo by Mike_Kiev / iStock

IntechOpen