Identification of Roles and Protocols in the Analysis and Design of Multi-Agent Systems using the Business Process Concept

Pablo Villarreal[1], Maximiliano Alesso[1], Sebastián Rocco[1], María Rosa Galli[1,2], Omar Chiotti[1,2]

[1]GIDSATD – Universidad Tecnológica Nacional – Facultad Regional Santa Fe,
Lavaisse 610 – 3000 SANTA FE - Argentina

[2]INGAR-CONICET, Avellaneda 3657 – 3000 SANTA FE – Argentina
pvillarr@frsf.utn.edu.ar, chiotti@ceride.gov.ar

Abstract. Most organization-oriented methodologies for the analysis and design of multi-agent systems start defining the organization structure. This structure is defined in terms of roles that the agents have to play in the organization. However, there are cases where the identification of roles and protocols is not easy, so it is not convenient to start the analysis defining the organization structure of the multi-agent systems. In this work, we discuss the use of the organizational concept of business process to identify roles and protocols. Then, we present an approach, which is based on the business process concept, to extend the Gaia methodology. This approach defines how to derive the roles and protocols of a system from the business processes. This approach is described following a real case.

Keywords: multi-agent systems; analysis and design; organizational concepts; business process.

1. Introduction

The extended diffusion that the agent technology has had in the last years has brought about the study of different aspects in the area of the agent-oriented software engineering [8]. One aspect is the development of methodologies and models for the analysis and design of multi-agent systems (MASs). In the recent years, different methodologies have been proposed, which have focused mainly in the inter-agent design aspects of the multi-agents systems instead of the intra-agent design aspects as agent models or architectures. These different proposals for the analysis and design of multi-agent systems can be classified in three types [2,9,11], according to the application of traditional modeling techniques or methodologies and the application of new approaches in the area of the software engineering. One type extends or adapts object oriented techniques and models for the analysis and design of multi-agent systems. Other type extends or adapts methodologies or modeling techniques derived from knowledge engineering for the analysis and design of multi-agent systems. The third type refers to those that model and design a multi-agent system as an organization.

Most organization-oriented methodologies [1,9], which model and design a MAS as an organization, start defining the organization structure. This structure is defined in terms of roles that agents have to play in the organization. However, there are cases where the identification of roles and protocols is not easy, so it is not convenient to start the analysis defining the organization structure of the multi-agent systems. We think that the same organizational principle is the source of other concepts that can be used in the analysis and design of systems as multi-agents organizations.

This paper focuses in the organizational concept of business process to identify roles and protocols in the analysis and design of multi-agent systems. Section 2 discusses briefly different types of proposals that have contributed to introduce concepts to guide the analysis and design of multi-agent systems. Section 3 describes the concepts used in the organization-oriented methodologies and introduces the business processes concept to identify roles and protocols. Section 4 describes more details of the Gaia methodology and present an approach, which is based on the business process concept, to extend the Gaia methodology.

2. Types of Methodologies for the Development of Multi-Agent Systems

One type of methodologies takes concepts of the object oriented development, extending or adapting object oriented techniques and models, to develop multi-agent systems. In spite of there are several similarities between agents and objects, there are also a number of important differences [8]. These differences bring about that the object oriented modeling techniques are poorly suited for multi-agent systems. However, there are several proposals that attempt to adapt or extend, object oriented modeling techniques and methodologies [4,5,7]. Some disadvantages of these proposals for the analysis and design of multi-agent systems are: they fail to capture the autonomous and proactive behavior of agents, as well as the richness of the interactions; although they can sometimes achieve a good modeling of the autonomous behavior of agents and of their interactions, they lack of conceptual abstractions for adequately dealing with multi-agents organizations [11].

Other types of methodologies extends or adapts methodologies or modeling techniques derived from knowledge engineering to apply them to develop multi-agent systems. The design of knowledge-based systems also has similarities to the design of multi-agent systems: knowledge acquisition, modeling and reuse [2]. However, these methodologies conceive a centralized view of knowledge-based systems and thus, they do not address the social aspects of agent in an organization [11]. One of these proposals is MAS-CommonKADS [3]. This is an extension of the method CommonKADS used for developing knowledge-based systems. Although this extension saves some disadvantages of knowledge-based methods for its applicability at the development of MASs, it does not take into account an organizational view of the MASs.

Finally, the third type of methodologies refers to those methodologies that model and design a MAS from an organization-oriented point of view. These methodologies view a multi-agent system as an organization. Thus, they propose to develop multi-agent systems as a process of organizational design. Examples of complete and well-defined methodologies of this type are GAIA [9] and MaSE [1]. These methodologies take into account the autonomous nature and the proactive behavior of agents. This principle exhibits the convenience of thinking an information system as an organization in which agents play roles and participate of interactions among roles.

3. Multi-Agent Systems as Organizations

The idea of thinking a MAS as an organization suggests that it is better to design a MAS making a parallelism with the behaviors and structures of the human organizations [10]. This principle is derived from the autonomous nature and the proactive behavior of agents that constitute a system. In this way, each agent has a specific role in the organization, which determines its responsibilities in the organization and the interactions in which the agent role must participate to interact with other roles played by other agents. With this view, the organizational perspective has the purpose of making the development of MASs less complex and easier of managing with respect to other traditional methods used in the software engineering [10].

3.1. Concepts Used in Organization-oriented Methodologies for MASs

The concept of role is the main organizational concept used in the organization-oriented methodologies for the analysis and design of MASs. A role can be viewed as an abstract description of an expected function to be performed in the organization [9]. Thus, the roles model defines the behavior expected of the agents in the organization. A role not only defines the responsibilities of an agent in the organization but also its position within the organization. In these methodologies, the organization is modeled by a collection of roles that have relationships to one another through patterns of interactions with other roles. The roles define the structure of the organization in the analysis phase and then, in the design phase, what is the role that each agent must play in the organization is decided. In this way, these methodologies propose starting the analysis phase from the definition of the organization structure and they only use one organizational concept: the role concept.

One of the well-defined organization-oriented methodologies is Gaia [9]. Once defined the system requirements, Gaia proposes to start directly the identification of roles, then the definition of protocols and finally the complete definition of the roles. Thus, Gaia does not provide concepts nor a procedure to guide the identification of roles and protocols. A recent work [10] introduces three news organizational concepts: organizational rules, organizational structures and organizational patterns. These organizational concepts are used to extend Gaia methodology. These organizational concepts are introduced to save some limitations of the Gaia methodology for its application in open multi-agent systems. In these systems, the number of agents and their relationships change in run-time. Also, in open systems the agents can exhibit a competitive or self-interested behavior, which is not present in closed systems where the agents have a cooperative behavior. In this extension of Gaia, the analysis phase starts with the definition of the organizational rules, then the roles and protocols are identified and finally the complete roles model and the interactions models are defined based on the selected organizational structure.

Other well-defined organization-oriented methodology is MaSE [1]. This methodology is also suitable for closed MAS. But, in contrast with Gaia, the analysis phase starts using some techniques of requirement specification, like goals and use cases. Then, roles are identified based on sequence diagrams derived from the use cases. In the sequence diagrams the initial set of roles and communications paths are identified. The unique organizational concept used in MaSE is the role concept, which is derived from the use cases and sequence diagrams. However, the use cases with their sequences diagram do not capture all possible scenarios, and thus, all the required roles and interaction protocols of the system may not be identified.

In these organization-oriented methodologies, we think it is not easy to identify roles and protocols. In some cases, the identification of roles may be directly derived from the real-world organization roles that the MAS is designed to support. But it is not always true. At other times, there is not a similar counterpart of the MAS in the real-world organization, or the real-world organization is not well defined and structured, or the application of the MAS implicate changes in the real-world organization. In these cases, it is difficult the identification of roles and protocols, required by the system. Therefore, we believe that the organizational principle may be the source of other concepts that can be used for the identification of roles and protocols in the analysis and design of systems as multi-agents organizations.

3.2. The Organizational Concept of Business Process for the Development of MASs

Following the principle of designing a MAS as an organization, we believe that one important concept that can be used in the development of MASs is the business process concept. There are several definitions of business process, but we can define it as the set of interrelated activities carried out in an organization in order to contribute to reach the organization goals. Current management techniques point out that it is convenient to coordinate and structure the organization activities according to a set of business processes [6]. All organizations start defining their goal and the processes are useful to describe what must be done to reach that goal. Thus, the primary objective of the business processes is to reach the organization goal.

In this way, from a point of view of the system as an organization, the processes define several aspects of the MAS. Firstly, the process activities define the functions to support in the MAS. Secondly, the activities precedence relationships and the activities sequences of the processes define the way in which the activities (system functions) have to be coordinated in the organization (system). Thirdly, the inputs and outputs of the processes define the interactions with the external entities of the system, those users or information systems that have to interact with the MAS.

In those aforementioned cases, where it is difficult to start defining the organization structure, we may suppose how should be that structure. Instead, if we consider the processes, we can know how the organization must behavior in terms of the activity relationships. Then, we can define the required roles to be carried out by each process activities and the required protocols to coordinate the process activities. Therefore, we propose to start the analysis phase of the development of MASs defining the organization processes. Thus, we can start defining the functional and coordination aspects of the organization and then, based on the processes definition, to derive the most suitable organization structure to reach the organization goal.

4. Extending Gaia Using the Organizational Concept of Business Process

This section presents an approach based on the concept of business processes to extend Gaia methodology. We think that this approach may be applied to others organization-oriented methodologies. Firstly, the models, the concepts and the development process of the Gaia are described. Secondly, the stages of the approach with their outputs are presented following a real case. Finally, a summary of the development process using this approach is described.

4.1. Gaia Methodology

The main concepts in Gaia are divided into two categories: abstract and concrete. The abstract entities are those used during the analysis and the concrete entities are those used during the design. The analysis phase has the purpose of defining the structure of the system in terms of the roles that have to be played in the agent organization and their interaction protocols. This is defined through the roles model and the interactions model, respectively. The roles model describes the attributes of the roles. A role is defined by four attributes: responsibilities, permissions, activities, and protocols. Responsibilities are represented by the safety and liveness properties. The liveness defines the potential execution trajectories through the activities and protocols associated to the roles. Safety properties define constraints that a role has to fulfill. Thus, responsibilities express the functionality of the role. Permissions define the rights that a role has when it accesses an information resource. Activities of a role correspond to a unit of action that a role performs without interacting with another role. The interactions model consists of a set of protocol definitions, one for each type of inter-role interaction. A protocol defines the way in which a role can interact with other roles. A protocol is defined by a number of attributes: its purpose, the role initiator, the role responder, its inputs and outputs and its processing. In the analysis phase, Gaia proposes three steps that can be performed iteratively. The first step is to identify the roles and define them informally. The second step is to identify the protocols associated to each of the roles. The third step is to elaborate the complete roles model and interactions model, defining the attributes of the roles and of the protocols.

The design phase has the purpose of defining the system in terms of the agent types that compose it, the services to be provided by each agent and their communication pathways. The design phase consists of three steps: to create the agent model, to develop the services model and to develop the acquaintance model. The agent model identifies the agent types that will make up the system, the agent instances that will carry out these agent types at run-time, and the mapping between roles and agent types. This mapping indicates the roles to be played by each agent type. The services model identifies the services that need to be carried out by each agent and specifies the main properties of these services. A service means a function of the agent. The services are derived from the protocols, activities, responsibilities and the liveness properties of a role. The acquaintance model documents the communication lines between the different agent types.

4.2. An Approach to Identify Roles and Protocols Extending Gaia

As we mentioned above, we propose an approach to extend Gaia methodology. This approach is based on the organizational concept of business process to allow developers to identify roles and protocols. We describe the approach following a real case in which it has been applied. The system we used as an example has the purpose of supporting the distributed tacit knowledge management within an organization. This system is being developed by GIDSATD Research Group. Following, we describe each stage of the approach and their output models.

Stage 1: Define the System Goal.

Every information system pursuits a certain purpose for which it was conceived and put into operation. Starting from an organizational abstraction of the MAS, it is necessary to take into account that every organization must define a goal and direct all efforts of every agent of the organization towards that goal. Then, the first step is to define the goal of the MAS.

Following the example, the purpose of the system is to offer a query mechanism to the decision makers of each enterprise domain that allows them to find people who have the knowledge to generate the information they require. Therefore, the goal of the system is “to process decision maker's queries in natural language and to find the possible information sources that allow obtaining answers to the queries requirements of the decision makers".

***Stage 2:* Define the Processes.**

Once the organization goal is defined, it is possible to define one or more processes whose primary objectives are to reach that goal. In this way, processes are useful to describe what must be done to reach the organization goal, in terms of the organizational activities and their relationships. Then, by considering the processes, we obtain the knowledge needed for the specification of the roles to be played in the organization and the protocols associated to the roles. From the definition of the business processes, we can obtain the following:

- The inputs needed and the outputs to be produced to the external entities of the organization (in the MAS the external entities can be the users or other information systems).
- All activities that the organization has to carry out and their precedence relationships.
- The flow of information among the activities.
- All possible activity sequences or control flows in the organization.

The concept of process activity is also quite important for our methodological proposal. Therefore, each activity of the process is defined in an activity template and characterized by the following attributes: activity objective, inputs, outputs, tasks involved in the activity, constraints and information resources (information used by the activity).

Fulfillment of a process activity allows reaching its objective, and the achievement of successive objectives involved in each activity allows the organization (the system) to reach its goal. In addition, the activity sequences or control flows of the processes determine how the activities (system functions) have to be coordinated in the system. In this way, the organization processes establish the form in which the roles to be played by agents in an organization, which carry out the process activities, have to be coordinated to reach the organization goal.

Following the example, the system requirements can be summarized in this way:

- There are people in the company that would like to ask for some information. It has not been foreseen that this information could be required. It already exists or can be generated in the company but it is not known who has it or can generate it.
- There are people that could receive the query and generate an answer since they have the requested information.
- The user that originated the query would like to consult the state of his query.
- The person that originated the query will then receive one or more answers from the different information sources.
- The person that originated the query could cancel it after the information search has started.

The users supported by the system will provide the inputs and receive the system outputs, so these users are the external entities of the system. Such users perform activities in geographically distributed places in the company, which are called Domains. It should be stressed that domains are autonomous and each of them has information systems to support the decision processes that are carried out inside them.

From this brief description of the system requirements, we gather that there are three main processes: (I) get the required information, (II) get the states of an issued query for information and (III) cancel a query. Processes have been designed following the process design methodology proposed in [6] and have been represented using diagram activity in UML. The defined inputs and outputs of the processes are shown in Figure 1.

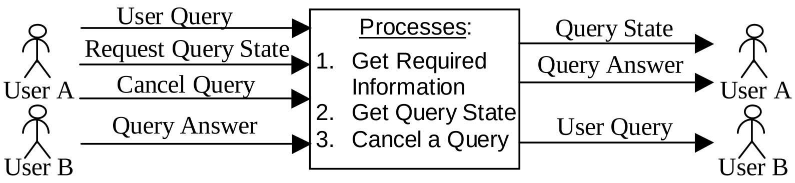

Figure 1. Inputs and outputs of the processes

For the purpose of showing our approach, we only describe the main process I, which is shown in figure 2. This process consists of seven activities and its goal is to obtain an answer to a user's query. The first activity of this process has the purpose of obtaining the user's query. The second activity has the purpose of coordinating all user's queries generated by the domains selecting the next query to be processed. The third activity has the purpose of determining the possible information sources (domains) for providing the required information to the user's query. The forth activity has the purpose of visiting the domains that the previous activity classified as possible providers of the required information. The fifth activity has the purpose of obtaining the answer to the query. The sixth activity has the purpose of delivering the answer to the user that issued the query, and the seventh activity has the purpose of learning from this search, thus improving the knowledge about domains as regards the information they can provide.

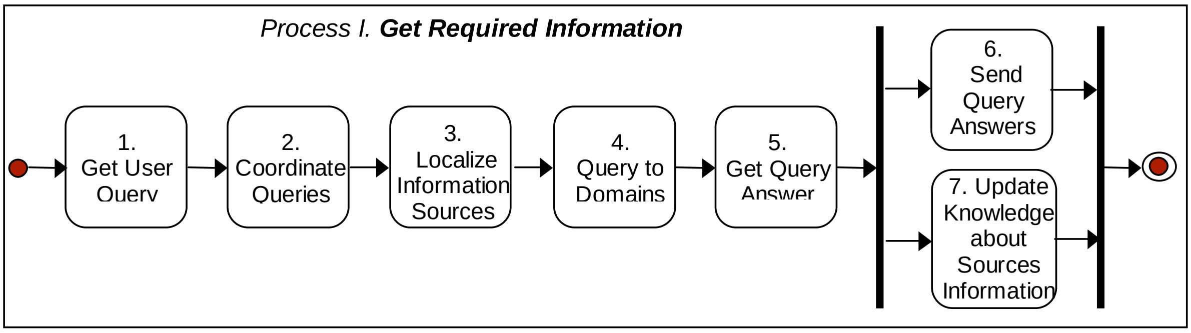
Process I. Get Required Information
1. Get User Query
2. Coordinate Queries
3. Localize Information Sources
4. Query to Domains
5. Get Query Answer
6. Send Query Answers
7. Update Knowledge about Sources Information

Figure 2. Process I: Get Required Information.

To complete the processes definition, the attributes for each activity are defined. As an example, Figure 3 presents the activity *LocalizeInformationSources*, which correspond to the main process.

Activity	*LocalizeInformationSources*
Object	Determine the domains that are potentially capable of providing the required information.
Inputs	User query.
Outputs	List of domains to be queried.
Tasks	1. Receive a user's query to identify possible information sources. 2. Match a user's query in natural language with the knowledge about domains information. 3. Generate a list of domains that are potentially capable of answering the user's query.
Constraints	Maximum number of queries to be received. Maximum number of queries to be simultaneously answered.
Resources	Knowledge Base about information managed by domains.

Figure 3. Attributes of activity *LocalizeInformationSources*

Stage 3. Identify Roles and Protocols.

Following the processes definition, each activity in the aforementioned processes should be assigned to a role, which has to be played in the organization to achieve the objective of that activity. The interactions in which the role has to participate are also defined. Thus, the roles and protocols of the whole system are obtained. In this definition, some points should be taken into account for the identification of roles and protocols from the processes:

– Process activities define the roles to be played to carry out these activities in the organization.

– If some activities are functionally cohesive or use the same resources, they can be carried out by a unique role.

– Performing a process activity in the context of a process implies different interactions with other process activities through the defined precedence relationships. Such interactions are expressed by the activity inputs and outputs. These activity interactions define protocols.

Following the example, we have identified the roles of the system by analyzing the processes. From process I, the defined roles with the process activities assigned to each role are shown in the figure 4.

Role of the system	Process activity assigned to roles
DomainRepresentative	GetQueryUser
QueryCoordinatorsAdministrators	CoordinateQueries
InformationSourceLocator	LocalizeInformationSource, UpdateKnowledgeAboutInformationSources
QueryCoordinator	QueryToDomains
ResponseSupplier	SendQueryResponse

Figure 4. Roles of the system and the process activities assigned to the roles.

Once the roles have been identified, the protocols are identified following the activities assigned to each role and the activity precedence relationships in process I. Figure 5 presents the roles and protocols derived from process I. The process activities assigned to each role are shown by a tuple (Process_Number; Activity_Number). Protocols that settle interactions among roles are shown with arcs among roles, and the activity attribute of the roles is indicated with a loop.

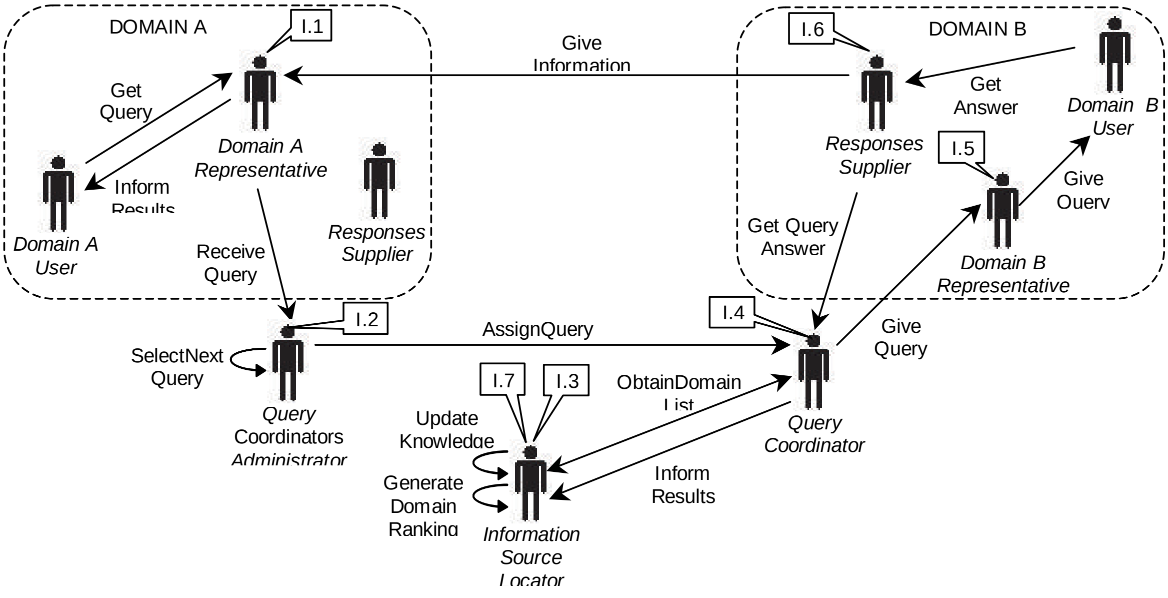

Figure 5. The Roles and Protocols from the main process *Get the Required Information*.

***Stage 4.* Define the Role’s Schemas and the Protocol’s Attributes.**

At this stage, roles model and interactions model are completed, defining the role’s schemas and the protocol’s attributes, as Gaia proposes. The attributes of the roles are defined in the role’s schema. Some defined elements in the processes can be taken into account to define some role’s attributes:

– A process activity can be represented by an atomic tasks sequence that has to be carried out as a transaction. The task concept is the same as the role activity concept in Gaia. Thus, activity tasks define the activities of a role in the role’s schema.

– Constraints of the process activity generally define safety properties of the role that carries out that activity.

– The resources identified in a process activity help to define the resources to be accessed by the role. The role permissions are defined on these resources to be used to perform the role.

– The activities sequences and the activity precedence relationships in the processes help to define the liveness property because it defines the potential execution trajectories of the role through its activities and associated protocols. These potential execution trajectories can be derived from the activities sequences and the precedence relationships among activities.

As an example, figure 6 shows a schema of the role *InformationSourceLocator* from the system of the example.

Role Schema: *InformationSourceLocator*

Protocols and Activities:
ReceiveQuery, MatchQuery, GenerateDomainsList, GiveDomainsList, GetQueryAnswer, UpdateKnowledge

Permissions:
reads *UserQueries*
QueryAnswer
changes *BaseKnowledge*

Responsibilities
Liveness:
InformationSourceLocator = ReceiveQuery.MatchQuery.GenerateDomainsList.
GiveDomainsList | |GetQueryAnswer.UpdateKnowledge
Safety:
UserQueries < n // number of user queries attended simultaneously
QueryAnswers < m // number of query answers attended simultaneously

Figure 6. Schema of the role *InformationSourceLocator.*

Stage 5. Define the Agents, the Services and the Acquaintance Models.

This stage corresponds to the design phase of Gaia methodology. Therefore, we follow the steps proposed by Gaia in which the agent, the services and the acquaintance models are defined.

Following the example, we start with the definition of the agent model. Figure 7 shows the four agents types that constitute the multi-agent system. These agents are the Domain Representative Agent (DRA), the Information Source Locator Agent (ISLA), the QueryCoordinatorsAdministratorAgent (QCAA) and the Query Coordinator Agent (QCA). In this model, the roles that each agent has to perform in the system organization can be seen. The number of DRA’s instances depends on the number of domains to be supported by the system, since there is a DRA for each domain. Once the agent model was defined we have to define the services model, which is not shown due to space limitations.

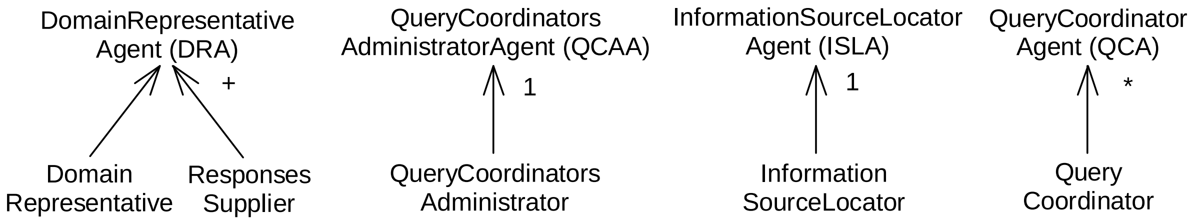
DomainRepresentative Agent (DRA)
+
Domain Representative
Responses Supplier
QueryCoordinators AdministratorAgent (QCAA)
1
QueryCoordinators Administrator
InformationSourceLocator Agent (ISLA)
1
Information SourceLocator
QueryCoordinator Agent (QCA)
*
Query Coordinator

Figure 7. The agent model

Figure 8 shows the acquaintance model. It shows the communication links between agent types. The DRA has a loop that points out a communication link between agents of some type.

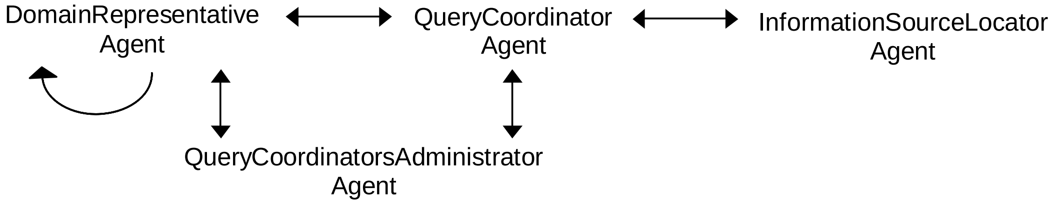
DomainRepresentative Agent
QueryCoordinator Agent
InformationSourceLocator Agent
QueryCoordinatorsAdministrator Agent

Figure 8. The Acquaintance Model

4.3. Summary of the Analysis and Design of MASs Following the Approach

Having described each stage of the approach to extend Gaia methodology for the identification of roles and protocols, the resulting methodology process with the stage outputs can be summarized as follows:

Stages of the Analysis Phase:

1. *Stage 1:* define the system goal. The output of this stage is the organization goal of the MAS.
2. *Stage 2:* define the necessary processes to achieve that goal. The output of this stage is a processes model in which all activities with their precedence relationships, the inputs and outputs, the flow of information and the possible activities sequences are defined. The model is completed with all activity templates in which the activity attributes are defined for each process activity.

Stages of the Design Phase:

3. *Stage 3:* identify roles and protocols in order to define the roles model and the interactions model. The output of this stage is a roles collection to be performed by the system with the process activities and the protocols associated to each role.
4. *Stage 4:* define the role's schemas and the protocol's attributes, in order to complete the roles model and the interactions model, based on the processes definitions. The output of this stage is a complete roles model and interactions model with the defined role attributes and protocols attributes.
5. Iterate from stage 3 to stage 4.
6. *Stage 5:* define the agent, the services and the acquaintance models. This stage should be carried out following the steps proposed in the design phase of Gaia methodology. The first step is to define the agent model, the second step is to define the services model and the third step is to define the acquaintance model.

The relationships among the models are shown in the figure 9.

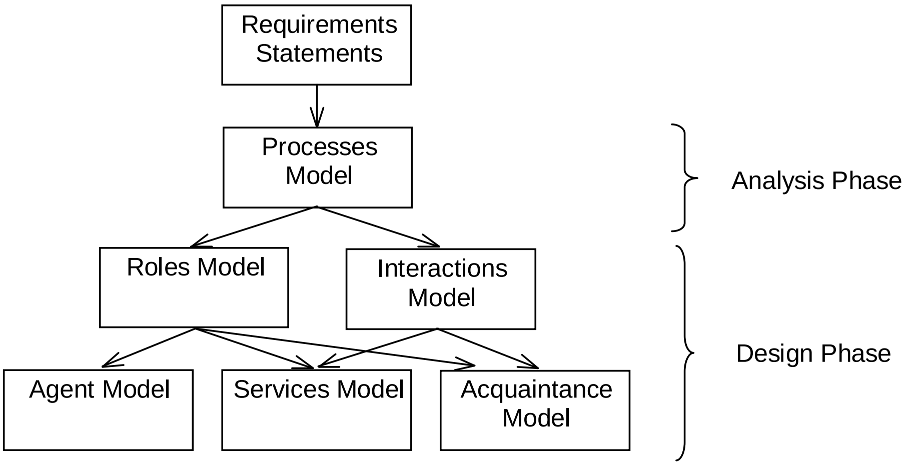

Figure 9. Relationships among the models

5. Conclusions

The organization-oriented methodologies for the analysis and design of multi-agent systems start defining the organization structure. This means to define, in the first step of the analysis phase, the roles and the protocols in which they participate. However, in some cases the identification of roles and protocols is difficult, as we have described in this work. Although there is a proposal to extend the Gaia methodology using other organizational concepts [10], this proposal is also based on types of organization structures to identify roles and protocols and does not provide a method to identify the preliminary roles and protocols.

Therefore, in this work, we have proposed to use the organizational concept of business process for the identification of roles and protocols. In this way, the analysis phase starts defining all the organizational activities, the activities precedence relationships and the possible activities sequences, through the business processes to be carried out in the organization to reach its goal. Thus, from the functional and coordination aspects of the organization defined in the business processes it is possible to derive the most suitable organization structure to reach the organization goal.

Furthermore, we have described an approach to extend the Gaia methodology, which is based on the organizational concept of business processes. In this approach, we have described how to derive roles and protocols from the business processes. As well, we have described how to derive the attributes of the roles and protocols, which are used in the Gaia methodology, from the processes and the process activities attributes. Thus, it may be possible to use a tool that allows developers to specify the processes model, to define the roles and to assign the process activities to the roles and then, based on this mapping, automatically to generate the attributes of the roles and protocols. The use of a development tool based on this approach may have other advantages: it may hold consistence among the processes model and the roles and the protocols models, it may take into account the best practices of the organization in terms of business processes, and it may lead the system architects who have familiarity with the business processes definition to the development of multi-agent systems.

An important aspect of the approach is the development process. In the development of a system, the analysis phase should describe what the system has to do and the design phase should describe how the system can best fit the requirements identified in the analysis phase. In the organization-oriented methodologies, the analysis phase has the purpose of producing a collection of roles, which describe what the system has to do to meet its requirements. But, once the roles are defined, the organization structure in terms of roles is already stated in the analysis phase. Instead, in our approach, the analysis phase has the purpose of producing a set of organizational processes. In this way, what the system has to do is described in terms of process activities with their precedence relationships. Then, the design phase of the approach has the purpose of defining which type of organization structure is the most suitable. Therefore, the role models and the protocols model should be defined in the design phase.

Finally, although we have used our approach in a real case and other examples, it should require some other experiences for its complete validation. In addition, the approach is more suitable for its application to closed systems where all the agents and their interactions are defined in design-time. Also, since the business processes have the objective of coordinating all the activities of the organization to reach the organizational goal, this approach is not suitable for systems in which the agents present a competitive and self-interested behavior. However, according to this work, we consider that the organizational concept of business processes is an important concept to use in the organizational-oriented methodologies for the analysis and design of MASs. So, the application of this concept to open systems should be studied.

References

1. DeLoach, S., Wood, M., Sparkman, C. *Multiagent Systems Engineering*. Int. Journal of Software Engineering and Knowledge Engineering. Vol. 11, no.3 (2001) 231-258.
2. Iglesias, C., Garijo, M., and Gonzales, J. *A survey of agent-oriented methodologies*. In A.S. Rao J.P. Muller, M.P. Singh, editor, Intelligent Agents V (ATAL 98). Springer-Verlag (1998).
3. Iglesias, C., Garijo, M., Gonzales, J., and Velasco, J. *Analysis and Design of multi-agent systems using MAS-CommonKADS*. In Intelligent Agents IV (ATAL 97), Springer-Verlag (1998).
4. Kendall, E. *Role Modeling for Agent System Analysis, Design, and Implementation*. First International Symposium on Agent Systems and Applications (ASA'99), Third International Symposium on Mobile Agents (MA'99), Palm Springs, October, 1999.
5. Kinny, D., Goergeff, M., and Rao, A. *A methodology and modelling technique for systems of BDI agents*. Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW’96. Lecture Notes in Artificial Intelligent, vol. 1038. Springer-Verlag (1996).
6. Klein, M. *Reengineering Methodologies and Tools*. IS Management (1994) 30-35.
7. Odell, J., Parunak, H.V.D, Bauer, B. *Extending UML for Agents*. Proc. of the Agent-Oriented Information Systems Workshop at the 17th National conference on Artificial Intelligence, Gerd Wagner, Yves Lesperance, and Eric Yu eds, Austin, TX, pp 3-17 (2000).
8. Wooldridge, M. and Ciancarini, P. *Agent-Oriented Software Engineering: The State of the Art*. In P. Ciancarini and M. Wooldridge, editors, Agent-Oriented Software Engineering. Springer-Verlag Lecture Notes in AI Volume 1957, (2001).
9. Woooldridge, M., Jennings, N., Kinny, D. *The Gaia Methodology for Agent-Oriented Analysis and Design*. Journal of Autonomous Agents and Multi-Agent Systems. Vol. 3(3), 2000.
10. Zambonelli, F., Jennings, N. R., Wooldridge, M. *Organizational rules as an abstraction for the analysis and design of multi-agent systems*. International Journal of Software Engineering and Knowledge Engineering. Vol. 11, no 3 (2001) 303-308.
11. Zambonelli, F., Jennings, N., Omicini, A, Wooldridge, M. *Agent-Oriented Software Engineering for Internet Applications*. In Coordination of Internet Agents (eds. A. Omicini, F. Zambonelli, M. Klusch and R. Tolksdorf) Springer Verlag, 326-346 (2001).