AUTONOMOUS AGENTS IN ORGANIZED LOCALITIES: METAMODEL AND CONCEPTUAL ARCHITECTURE

Michaela Huhn, Aret Duraslan, Gianina Gănceanu, Jana Görmer, Jörg Hähner, Jörg P. Müller, Christian Müller-Schloer, Christopher Mumme, Christian Schulz

22. December 2010 NTH Computer Science Report 2010/02

This work was funded by the NTH Focused Research School for IT Ecosystems. NTH (Niedersächsische Technische Hochschule) is a joint university consisting of Technische Universität Braunschweig, Technische Universität Clausthal, and Leibniz Universität Hannover.

niedersächsische technische hochschule
Technische Universität Braunschweig
TU Clausthal
Clausthal University of Technology
Leibniz Universität Hannover

IMPRESSUM

Publisher

NTH Focused Research School for IT Ecosystems
Technische Universität Clausthal, Julius-Albert Str. 4, 38678 Clausthal-Zellerfeld, Germany

Editors of the series

Prof. Dr. Christian Müller-Schloer,
Prof. Dr. Andreas Rausch,
Prof. Dr. Lars Wolf

Technical editor

Dr. Dirk Niebuhr
Contact: dirk.niebuhr@tu-clausthal.de

ISBN 978-3-942216-14-2

AUTHORS

Aret Duraslan	Leibniz Universität Hannover
Gianina Gănceanu	TU Braunschweig
Jana Görmer	TU Clausthal
Jörg Hähner	Leibniz Universität Hannover
Michaela Huhn	TU Clausthal
Jörg P. Müller	TU Clausthal
Christopher Mumme	TU Clausthal
Christian Müller-Schloer	Leibniz Universität Hannover
Christian Schulz	Leibniz Universität Hannover

AUTONOMOUS AGENTS IN ORGANIZED LOCALITIES: METAMODEL AND CONCEPTUAL ARCHITECTURE

ABSTRACT

IT Ecosystems describe an upcoming type of complex socio-technical systems that consist of a large number of heterogeneous subsystems that execute partly autonomously and cooperate in order to meet individual or joint objectives. IT Ecosystems are planned as open systems; the subsystems are developed independently under different design rationale.

To meet the challenges of IT Ecosystems we propose the concept of Autonomous *Agents in Organized Localities* (A²OL). A locality provides the infrastructure in terms of domain ontologies, coordination protocols, and institutions; it builds a common basis for the interaction of autonomous agents. Within a locality, the agents may collaborate by building teams or by joining an organization. In addition, institutions are introduced to represent public authorities regulating to the behavior of individual agents by means of norms and associated normative mechanisms like monitoring norm compliance or sanctioning.

We provide a metamodel to structure the relevant concepts of A2OL and conceptual architectures for agents and institutions to facilitate the specification and modeling of this kind of systems. To evaluate our approach we consider an airport departure scenario as a first case study.

1 INTRODUCTION

IT ecosystems are large socio-technical systems which are composed of numerous subsystems. These subsystems can be networked IT-systems which are often developed and provided independently by different organizational bodies. This means that it is hardly possible to guarantee exhaustive testing or even verification of the functionality provided by the IT ecosystem as a whole. Additionally, humans are involved with these in various ways, for example, by developing, using, and modifying the system. In particular, these interactions are usually done in or with small parts of the whole.

In this document, we provide a meta-model and an architecture for a particular class of IT ecosystems and introduce the concept of autonomous agents in organized localities (A²OL). One of the major goals is to provide on the one hand a concrete enough foundation for analytical modeling and specification of such systems. On the other hand, the aim is that this foundation is broad enough to foster the development of a significantly large and relevant class of applications. The application domain addressed by A²OL is based on the idea that in many environments individuals and IT-Systems are situated in a given physical (or virtual) environment: the so-called locality. An example for such a locality is an airport. Interactions between the players in such a locality are manifold and very complex. For this reason, we provide the definition of an agent in a locality, which represents the basic participant of an IT ecosystem and can be either a technical subsystem or a human being in the given locality. Examples for agents in the airport scenario are passengers and worker as well as autonomous vehicles and the airline's check-in system. Looking only at the combinatorial complexity of possible interactions, these cannot be specified and verified in advance in most cases, especially because they may change over time due to changing demands. Thus, interactions need to be regulated at run-time. Therefore we provide the definition of institutions and organizations in the course of this document. In these definitions, institutions are entities which are responsible for the definition of mechanisms to define, regulate and evolve what a well-behaved type of agent is allowed to do in a given locality. An example institution for the airport is the aviation authority, which defines passengers' security at airports and derives specific rules for its implementation. Organizations in this context are moreover responsible for the coordination of (particular types of) agents in (particular sub-) localities. The main task of this coordination is to reach common goals provided by institutions using for example conflict resolution mechanisms or by checking the compliance of agents to particular constraints or procedures. The federal police implementing the security procedures provided by the aviation authority are an example for such an organization. The interaction between all players in a locality is multifaceted. Therefore, another important contribution of this document is the separation of their interactions in different dimensions, namely the domains of providing norms, execution, negotiation, and the homeostatic dimension.

In the remainder of this document we first provide definitions for basic concepts used for A²OL. These include the definition and discussion of localities, agents, organizations and institutions. Following this, the document structures the interaction dimensions before providing a particular architecture for A²OL. This architecture is then refined with respect to technical aspects and leads to a very concrete example, an airport transportation scenario, which is used to evaluate the previously defined model and architecture of A²OL.

2 BASIC CONCEPTS FOR AUTONOMOUS AGENTS IN ORGANIZED LOCALITIES

2.1 Locality

A locality has a purpose or a subject that attract agents to meet there. A locality provides a technical or physical infrastructure to be used by the agents to achieve their individual goals that are usually related to the subject of the locality. A locality can be understood as a physical or virtual place offering a number of opportunities. However, the locality does not require a unified objective or architecture for the agents to participate. In that, a locality comprises a system of systems. It has a scope defining a boundary, so agents may enter, leave, and return later to the locality. Examples of localities are an airport, a fair, or a social network. The infrastructure is part of a local ontology representing domain knowledge. A locality often sets the stage for a possibly hierarchical structured collection of scenes describing pre-defined interaction templates for particular coordinated activities or to achieve certain subgoals. Moreover, the locality may present the order in which agents may traverse the scenes. In an airport, for instance, we find an entry scene, a check-in, a security check and a boarding scene, all with a number of subscenes. The scenes have to be traversed in the listed order and during the transit from one scene to the next one certain constraints have to be fulfilled.

Secondly, the locality is associated with institutions and may provide organizations to foster coordination and regulate the interaction of autonomous, heterogeneous agents beyond physical and technical constraints. Institutions regulate the agent behavior in order to balance between different interests and to establish and sustain certain notions of stability. Organizations structure the grouping and collaboration of agents within the locality. Organizations and institutions are introduced in Section 2.3 and 2.4, respectively. In Figure 1 the major concepts associated with a locality are depicted.

2.2 Agents

2.2.1 Definition

An agent is an entity which observes and acts upon an environment that can be a computer program (e.g., for simulation) or the real world (e.g., to support humans). For this interaction, the agents have sensors to gather information about the world and interpret its actual state. On the basis of this information, agents create output to change to world's state. In Figure 2, the agents are illustrated as a black box *[Mül96]*, because there are different ways (architectures) in which agents perceive information about the environment and produce output.

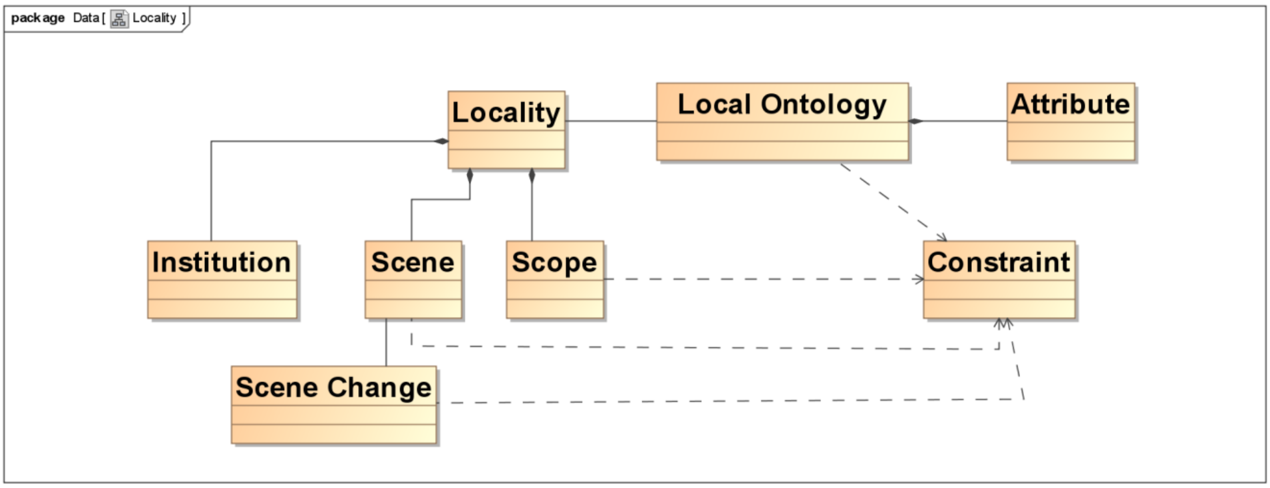

FIGURE 1. CONCEPTUAL MODEL OF A LOCALITY.

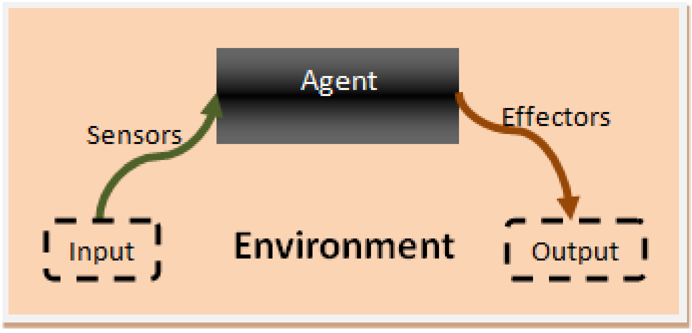

FIGURE 2. AGENT-ENVIRONMENT INTERACTION.

[Woo09] defines intelligent agents as capable of flexible autonomous action with the goal to meet their design objectives. By autonomous he means that agents are able to act without intervention from other systems (or humans). They therefore "have control both over their own internal state and over their behaviour" *[Woo09], p. 132*). In his definition, Wooldridge interprets "flexible" as denoting reactivity, pro-activeness and social ability:

- reactivity: Agents perceive the environment and create actions on basis of changes that occur in it in order to satisfy their design objectives (if-then rules). Figure 3 (left) shows the architecture of a reactive agent.
- pro-activeness: Agents are able to exhibit goal-directed behavior by thinking about alternatives for achieving goals and satisfying their design objectives. They thus take the initiative. Figure 3 (right) shows the IRMA (Intelligent Resource-Bounded Machine) Architecture *[PR90]* i.e., the architecture of a deliberative agent based on the BDI (Belief-Desire-Intention) Architecture *[RG91]*. The IRMA Architecture consists of a reasoning cycle for developing new plans and a deliberation process for updating the intention structure.
- social ability: Agents are able to interact with other agents or humans.

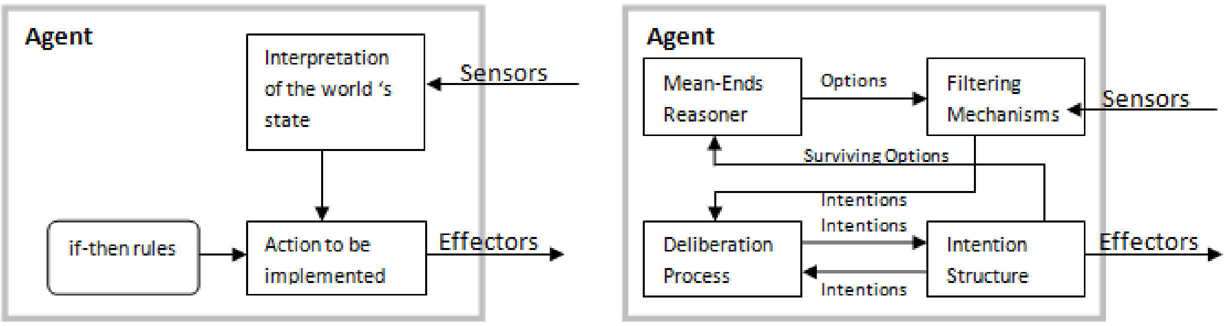

FIGURE 3. A SIMPLE REACTIVE AGENT ARCHITECTURE (LEFT) AND THE IRMA ARCHITECTURE BY [PR90] (RIGHT).

The two agent architectures illustrated in Figure 3 are part of the classification of agent architectures proposed by *[Mül96]*. He also described a hybrid architecture that is a combination of reactive and proactive architectures (e.g., the InterRap architecture *[Mül96]*) as well as social agent architectures.

We propose a concept of an agent-layered architecture which is supposed to be the blueprint for software agents and intelligent control systems, depicting the arrangement of components. The main goal is to be able to handle the full range of capabilities of an intelligent agent, from highly routine to extremely difficult open-ended problems. In order for that to happen, according to the view, it needs to be able to create representations and use appropriate forms of knowledge (such as procedural, declarative, episodic). The architectures implemented by intelligent agents are referred to as cognitive architectures. We decided for a layered architecture shown in Figure 4 in order to decompose complicated intelligent behavior into many "simple" behavior modules, which are in turn organized into layers. Each layer implements a particular goal of the agent, and higher layers are increasingly abstract. Each layer's goal subsumes that of the underlying layers, e.g., the decision to move forward by the action layer takes into account the decision of the lowest obstacle-avoidance layer. Our concept has three layers: **(1)** the execution layer, **(2)** the individual context layer and **(3)** the social context layer. The first two are the subjective context and represent the autonomous agent, the level 3 has the social context with its obligations. Our agents are based on the BDI-Model and interact within the layers bidirectional. The Belief-Desire-Intention (BDI) software model (usually referred to simply, but ambiguously, as BDI) is a software model developed for programming intelligent agents (cf. *[RG91]*). Superficially characterized by the implementation of an agent's beliefs, desires and intentions, it actually uses these concepts to solve a particular problem in agent programming. In essence, it provides a mechanism for separating the activity of selecting a plan (from a plan library) from the execution of currently active plans. Consequently, BDI agents are able to balance the time spent on deliberating about plans (choosing what to do) and executing those plans (doing it). A third activity, creating the plans in the first place (planning), is not within the scope of the model, and is left to the system designer and programmer. As opposed to more traditional AI approaches our architecture uses a bottom-up design. The interaction of agents is shown on the right hand side of Figure 4. Each agent is allocated to an institution which regulates the organized locality which imposes social and deontic norms. At present we think of a service where new agents coming into the system sign in. Interacting agents can be viewed vertically or horizontally where the balance between local goals and norms is figured out. One of our advantages is the modularity and the possibility to adopt to new designs as well as contexts. We are aware of finding solutions for the following disadvantages:

- the inability to have many layers, since the goals begin interfering with each other,
- the difficulty of designing action selection through highly distributed system of inhibition and suppression, and
- the consequent rather low flexibility at runtime.

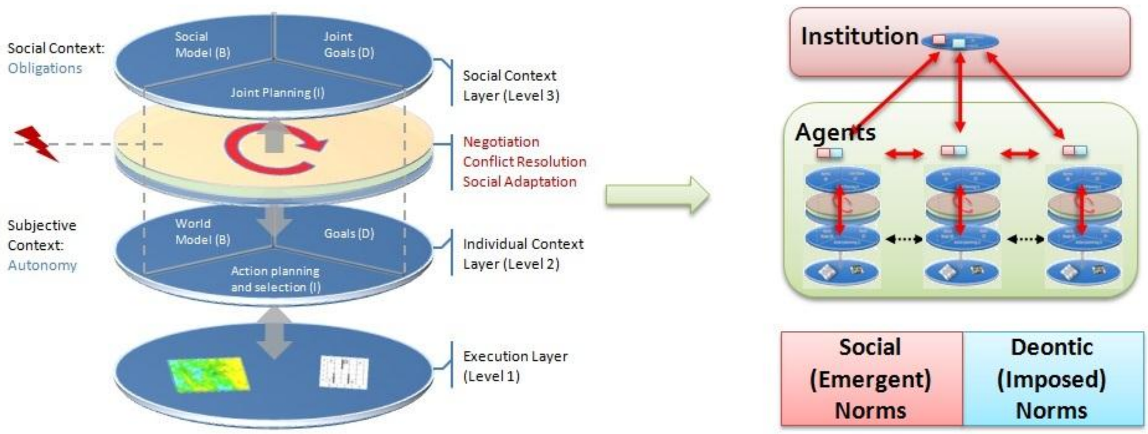

FIGURE 4. CONCEPT OF LAYERED AGENT ARCHITECTURE.

Our architecture can be characterized by certain properties or goals like learning. Some early theories such as SOAR and ACT-R originally focused only on the 'internal' information processing of an intelligent agent, including tasks like reasoning, planning, solving problems, learning concepts. More recently many architectures (including SOAR, ACT-R, PreAct, ICARUS, CLARION) have expanded to include perception, action and also affective states and processes including motivation, attitudes, and emotions. Our architecture is composed of different kinds of sub-architectures (described as 'layers' or 'levels') where the layers may be distinguished by types of function, types of mechanism and representation used, types of information manipulated, or possibly evolutionary origin. This is a hybrid architecture (like CLARION). We want our architecture to grow, e.g. by acquiring new subsystems or new links between subsystems.

2.2.2 Multiagent-system

A multiagent-system (MAS) is a system consisting of multiple agents having a large number of structural variations (heterogeneous) or multiple identical agents (homogeneous) that interact with each other in an environment. Interaction often implies communication, and therefore the role of communication is very important in an agent-based system. The model of communication is based on three features:

- Agents are able to make a decision regarding their actions. Communication between agents differs from communication between two objects in so far that agents can decline a request. Agents can say 'no'.
- Communication is a type of action (speech act theory). Agents can make plans by including communication at the same level of actions.
- Communication carries a semantic meaning that has to be understood by agents. Hence, it is necessary to define a standard (e.g., KQML or FIPA ACL).

2.2.3 Metamodel view

FIGURE 5 depicts the agents' aspect of the metamodel according to *[Fis09]*. An agent has an architecture (e.g. BDI) that describes its handling with world states (cf. chapter 2.2.1) An agent has access to a set of resources (information, knowledge, or ontologies) from its surrounding environment, i.e., the locality. Furthermore, the agent has goals. It can perform particular "DomainRoles" (to act in accordance to a plan) and behaviors, which are represented by the agents' capabilities. By acting the agent receives rewards (positive or negative). Additionally *[Fis09]* uses the concept of "Instances" that can be considered as run-time objects of an agent that defines the corresponding type. In our metamodel view the type of an agent is described by its architecture, capabilities and behaviors.

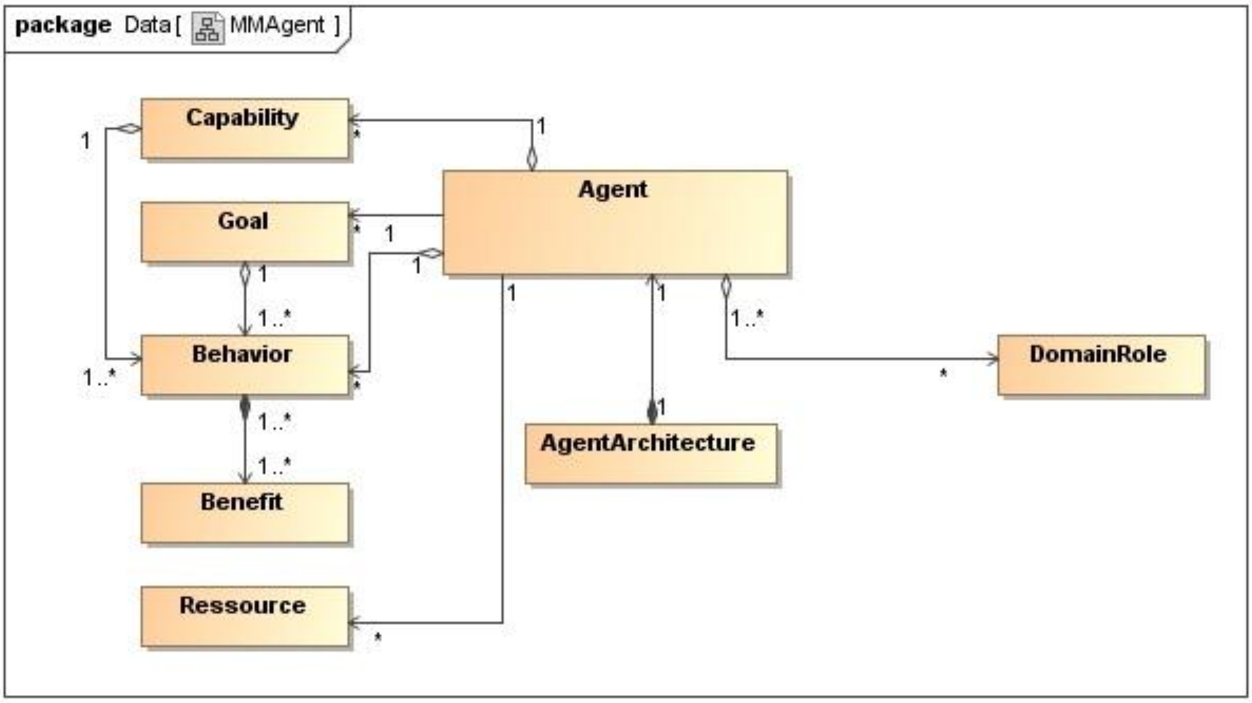

FIGURE 5. CONCEPTUAL MODEL OF AN AGENT (CF. [FIS09]).

2.3 Organizations

2.3.1 Definition

Interaction often implies that agents have to coordinate their activities e.g., to solve a problem or to reach common goals. In this case, agents have a relationship with each other as well as a form of organization *[FGM03]*. The main focus of an organization is on roles that agents take within the organization and that associate agents with one another. A role comprises the constraints (obligations, requirements, skills) that an agent will have to satisfy to obtain a role, the benefits (abilities, authorization, profits) that an agent will receive in playing that role, and the responsibilities associated with that role *[FGM03]*. So organizations can also be structured hierarchically e.g. by providing certain agents with more authority than other. Figure 6 shows a fully connected architecture of an organization including peer-to-peer and mass organization. A peer-to-peer is any distributed network architecture composed of agents that make a portion of their resources directly available to other agents, without the need for central coordination instances. Peers are both suppliers and consumers of resources, in contrast to the traditional client-server model where only servers supply, and clients consume. Mass organization can be seen as a corporative system of economic, political, or social organization or thought that views a community as a body based upon social and functional distinction and roles amongst individuals. The fully connected architecture has a general form of a chief director usually forming the single well-informed element so called "voice" of an organization to the outside, sub-division managers and the workers.

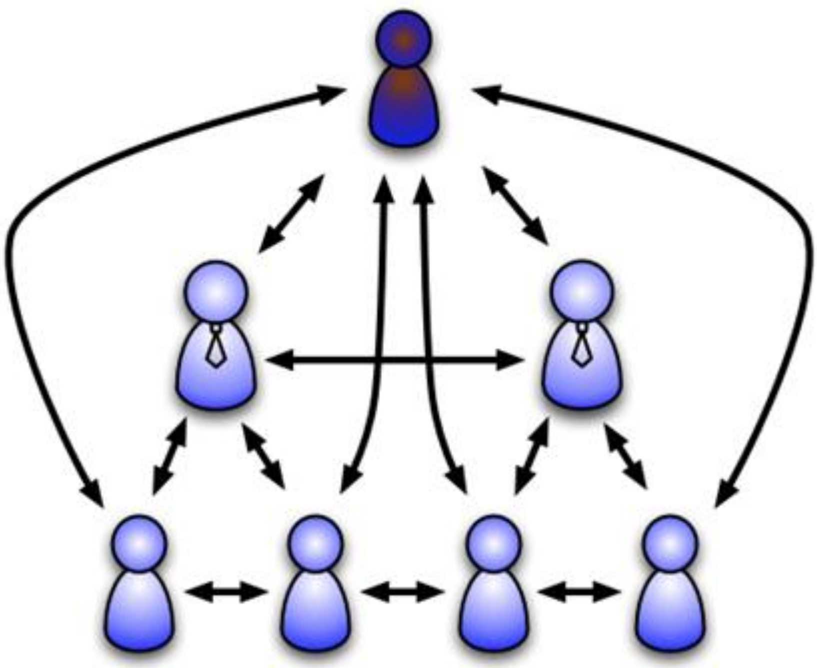

FIGURE 6. FULLY CONNECTED ARCHITECTURE OF AN ORGANIZATION.

External agents can see an organization as a single entity. An organization can therefore also be a participant of another organization with the result of a hierarchical structure of organizations and agents.

Figure 7 shows numerous agents that can interact amongst themselves (interaction lines) and/or have the same sphere of visibility and influence by participating in the same organizational relationship (dashed lines).

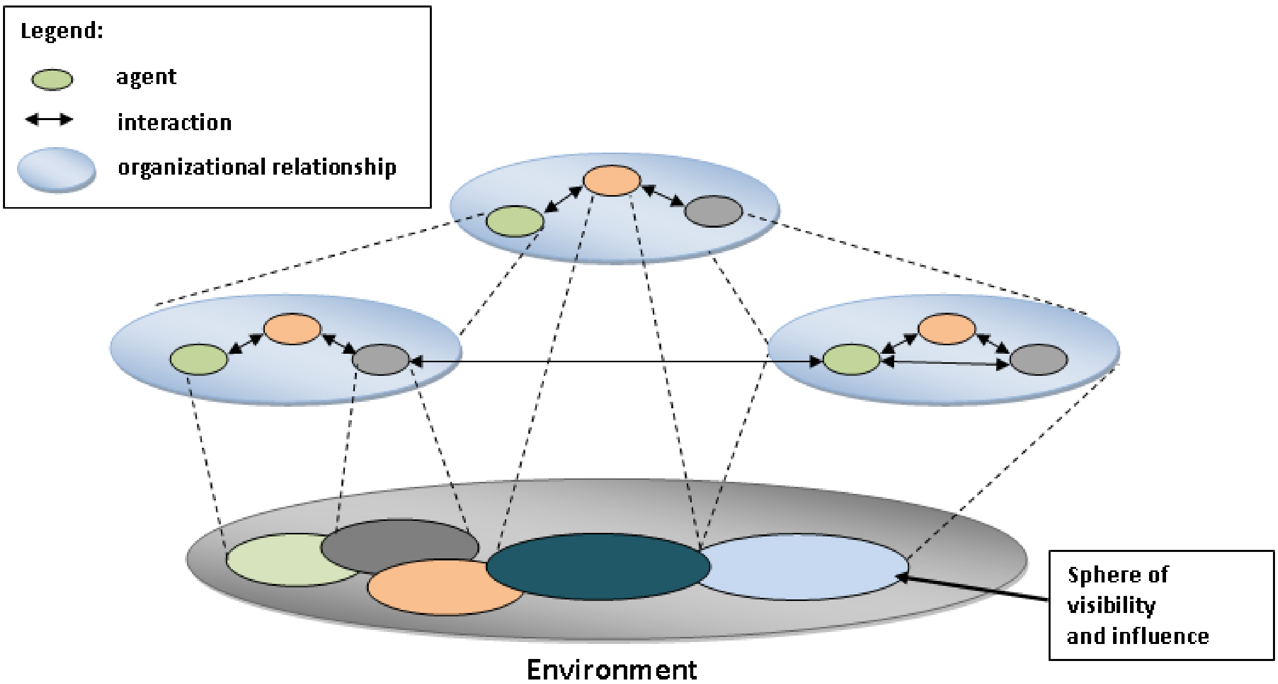

FIGURE 7. CANONICAL VIEW OF AN AGENT-BASED SYSTEM [JEN00].

2.3.2 Metamodel view

Figure 8 shows the metamodel of organizations that is an extension of *[Fis09]* organizational aspect. It includes the concepts Organization and its Structure, Team and its Context, Institution and Norm, Binding, InteractionUse, ActorBinding as well as Interaction and its Protocols for Communication and Coordination, DomainRole, Actor and Agent. An organization is derived from the agent aspect and it inherits characteristics of an agent, i.e. it has capabilities which can be performed by its members. The structure concept defines the pattern of the organization. It can bind agents or organizations to the "DomainRoles". Through the concept of interaction an organization has internal protocols that specify how its members communicate with each other and coordinate their activities. For interaction, "DomainRoles" are bound to actors (by "ActorBinding") that can be considered as representative entities within the corresponding interaction protocols. Thus an actor can be seen as an agent (or organization) with a role and task.
A team is a special kind of an organization. It is bounded by a context that limits the teams' existence in tasks and time. The concept of institution is defined in section 2.4.

The metamodel of the role aspect is depicted in Figure 9. It includes the concepts Role, Actor, and DomainRole as well as Capability and Resource (from the agent aspect). A role defines the behavior of an agent in a given context (e.g., an organization). It refers to a set of capabilities that define the behaviors it can perform and a set of resources it has access to. An actor can be considered as a generic concept as it either binds instances directly or through the concepts "DomainRole" and binding. The set of bound entities could be further specialized through the subactor (specialization of the superactor) reference that refers again to an actor (cf. *[Fis09]*).

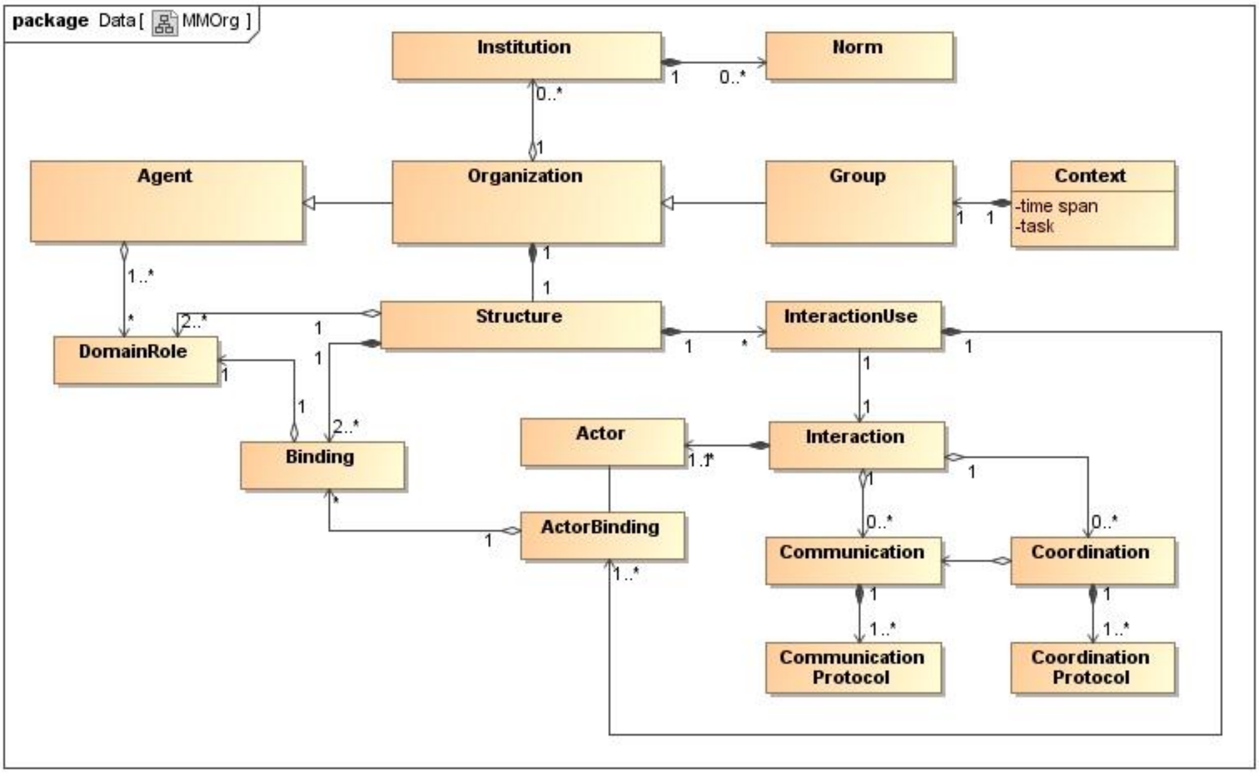

FIGURE 8. CONCEPTUAL MODEL OF AN ORGANIZATION (CF. [FIS09]).

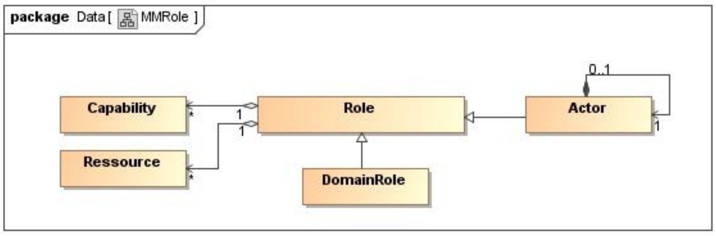

FIGURE 9. CONCEPTUAL MODEL OF A ROLE IN AN ORGANIZATION (CF. [FIS09]**).**

2.4 Institutions

A leading direction to cope with heterogeneity in open agent communities is to introduce superordinate guidance that sets the "rules of the game" *[AEN+05]* as an analogue to social structures and trusted public authorities in human societies. The concept of an electronic *institution* is established to subsume the notions and mechanisms to regulate the individual agent's behavior and to provide trusted services. However, the literature on institutions deviates significantly with respect on what and how normative regulations are established.

For now, we consider regulation and *norms* to be the main subject of an institution. Later it may be extended to also provide other kinds of trusted services. An institution is associated with a locality for which it provides all services that relate to normative regulations. It acts through an organization that executes institutional tasks. These tasks are twofold: **(1)** An institution provides mechanisms to ensure norm compliance of the agents; and **(2)** it may offer mechanisms for an adaptive norm establishment. With respect to norm compliance we view an institution mainly as an executive authority. Aspects of jurisdiction will become relevant in the context of complex norm structures or evolving scenes. The tasks contributing to norm compliance are:

- a *registry* administers the identities of agents currently present in the locality;
- *information services* provides the agents with knowledge about the current norms;
- a set of *monitors* monitors whether the agents behave according to the norms;
- *sanctioning* mechanisms assign positive or negative sanctions to the agents depending on their norm-relevant behavior;
- in case of obligations *norm enforcement* guarantees that control is imposed on the agent in such a way that the agent will behave norm-compliant.
- in case of ambiguities or conflicts in the context of norms, it may be necessary to provide *jurisdiction*, i.e. an institutional agent judging for the conflicting parties on the situation.

Norm establishment corresponds to legislative part of an institution. It covers the tasks of specifying new norms and adaptation of existing ones. Other important aspects are the placement of a norm in the norm structure and the refinement transformations to translate an abstract law like "don't imperil pedestrians" into operational rules that can be imposed on the agent behavior. Last but not least, the institution provides mechanisms to achieve an agreement on whether a new norm shall become effective. Examples for agreement strategies are voting, decision by an institutional leader or by external command. For the moment we will only consider static, elementary norm structures. So norm establishment boils down to reconfiguration of existing norms by command of an institutional authority like the airport traffic agency.

An overview on institutional mechanisms is given in Figure 10. The colored items depict those modules needed to establish normative regulations based on static norms. If one further aims at adapting norms or upgrading the norm system by legislative processes, more complex mechanisms for norm establishment and governance are needed which are out of the scope of the this report.

2.4.1 Metamodel view

Norms are an explicit description of the regulations that govern the behavior of an agent for the benefit of the community and itself as a member of it. The norm aspect of an institution is illustrated in Figure 11. A norm consists of the normative statement itself expressing the regulation imposed by the norm, a subject to which the normative statement applies, and the scope that may restrict its area of applicability to particular scenes or situations. Moreover, the norm has a specification of fulfillment and violation for monitoring compliance to this norm and the consequences of norm compliance in terms of possibly positive or negative sanctions are specified.

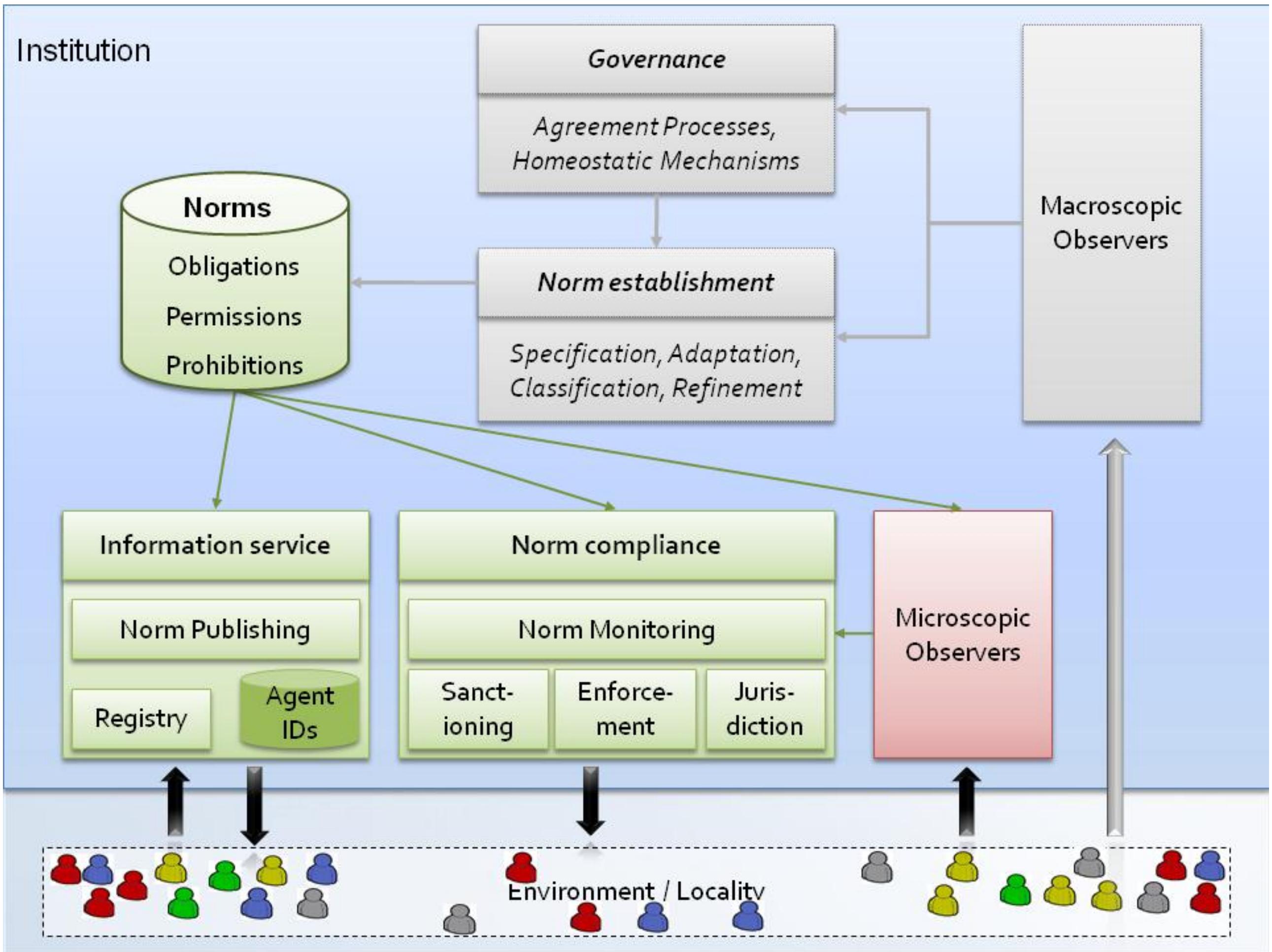

FIGURE 10. INSTITUTION ARCHITECTURE.

We distinguish different kinds of norms, namely obligations, prohibitions, and permissions. A set of norms is structured in a norm structure which classifies the norms with respect to abstraction levels, a priority scheme or context information.

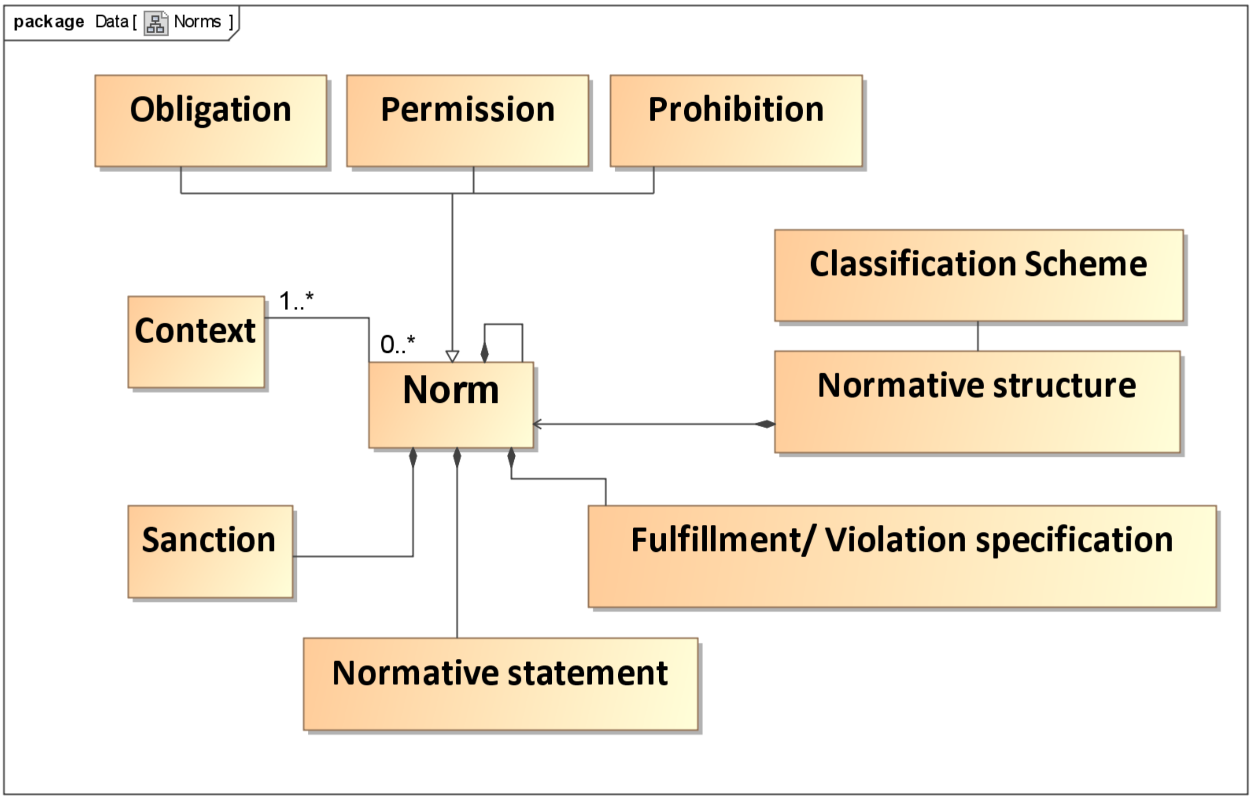

FIGURE 11. CONCEPTUAL MODEL OF NORMS.

3 DIMENSIONS OF INTERACTION

In *[DGG+09]* we considered four dimensions of interactions as relevant in A²OL (see Figure 12):

- *The task execution dimension* captures those concepts needed for the individual performing of production tasks in a subdomain of the locality. In the DemSy airport scenario such a productive subdomain maybe autonomous driving that is performed by an autonomous vehicle.
  The execution dimension focuses on (physical) activities that directly contribute to a productive of service task. Productive actions are usually the outcome of an individual, autonomous planning and action selection mechanism, nevertheless agents interfere via the effects of the action on the environment, e.g., driving on a particular lane with a certain speed affects all other vehicles sharing this street section.

- *The negotiation dimension* comprises the concepts and mechanisms for coordinating agents like various kinds of agreements and negotiation strategies. It allows the agents to organize themselves in order to execute joint tasks. Coordination is based on common ontologies and realized by communication actions and protocols. In the airport scenario, autonomous vehicles forming a transport service organization will use coordination concepts to distribute the passenger's request among them.

- *The normative dimension* deals with norms as a mean of regulating autonomous agents, norm establishment and the monitoring of norm compliance and sanctioning are considered here. Whereas the negotiation dimension addresses peer to peer coordination, normative regulations are established and enforced by an institution as a superordinate authority. An institution acts through its institutional agents that communicate with "productive agents" on normative issues. In the airport scenario traffic rules form a part of the norms.

- *The homeostatic dimension* provides concepts to balance between different interests and mechanisms to achieve the state of equilibria. Homeostatic mechanisms do not directly control the behavior of agents but interfluence the agents indirectly by adapting norms. In the airport scenario, the continuous flow of traffic may represent such an equilibrium that is achieved by adapting speed limits for particular roads due to the current traffic density.

Figure 13 indicates how the layered architectures of agents and institutions relate to these dimensions.

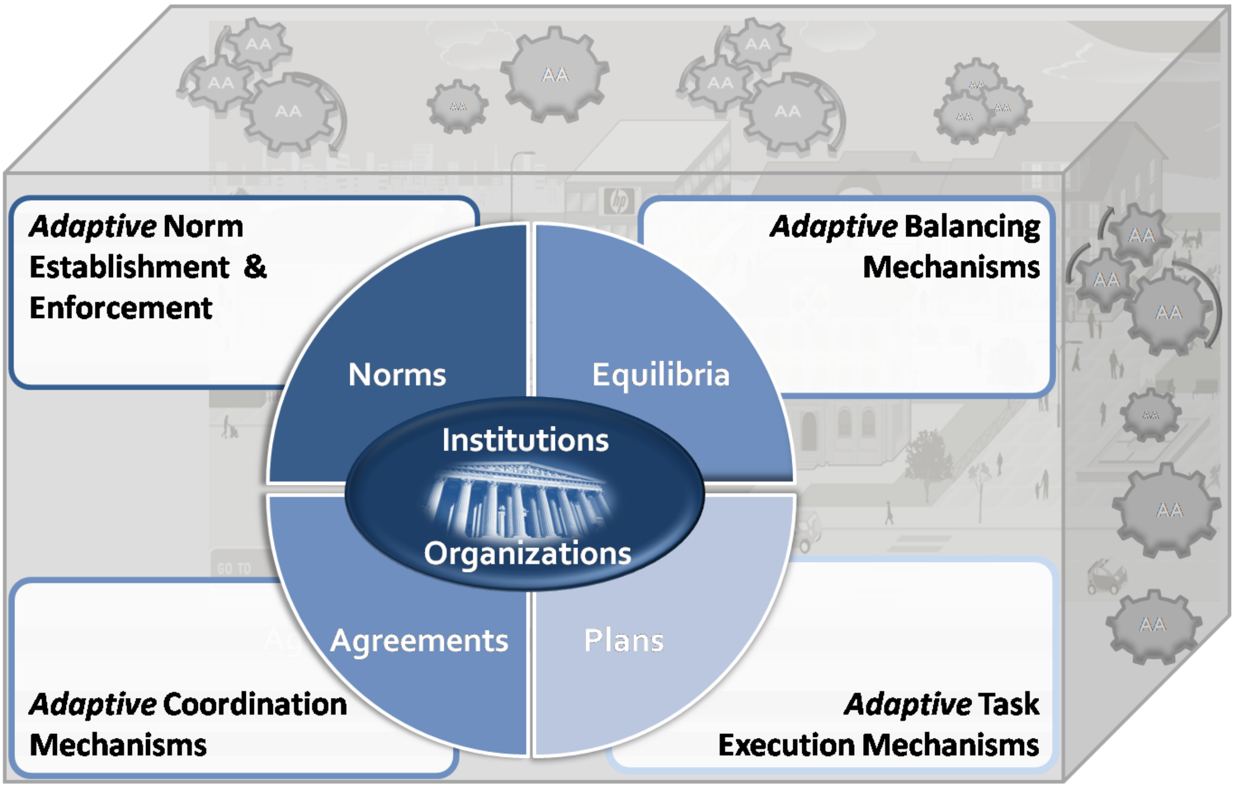

FIGURE 12. DIMENSIONS OF INTERACTION IN A²OL.

3.1 The Task Execution Dimension

The execution dimension comprises individual task and action execution and mainly concerns the lower layers of the agents' architecture, namely the robust execution layer and the mechatronic layer. These layers deal with the embodiment of agents and belong to its individual context. Nevertheless, interaction occurs as other agents are part of the physical environment and as such they have to be taken into account, e.g., because they form a sudden obstacle on the driving trajectory of an autonomous vehicle.

The mechatronic layer of an autonomous physical agent is built of quasi-continuously processing open- and closed-loop controllers and also the lower layers of its communication protocols. The mechatronic layer comprises the real-time control of the actuators. The robust execution layer considers actions on a level of abstraction that is adequate for planning and scheduling activities and in addition it controls and monitors the execution on the mechatronic layer. The robust execution layer contains modules for task scheduling and action selection. If various possibilities for the physical execution are possible in order to perform a specific *logical* action (e.g. "move from A to B"), the robust execution layer chooses one of them (e.g. "take the left junction") according to the current state of the system. In case of agents as (real) autonomous, mobile robots the first step is the reception of sensor data. The next step is a specific (application specific) preprocessing, i.e. filtering and rehashing of the received data. It can contain information about the "visible" environment, obstacles, other object but also about the current velocity of the robot, its battery status or the distance travelled so far. Depending on the type of agent (reactive/pro-active/hybrid), the information is compared to the current goals of the agent and the next actions and afterwards the explicit steering instruction that finally lead to the action (e.g. movement of the robot) are derived. These processes are running all the time the robot is active (not only in case of concrete decisions, like a junction). In addition to the locomotion of the robot, other forms of (mechanically) manipulations of the environment are possible. Especially in that kind of environments we consider in this paper, the interaction (e.g. the direct communication between each other as well as the communication with other components, like cameras or humans) plays a crucial role for the execution mechanisms.

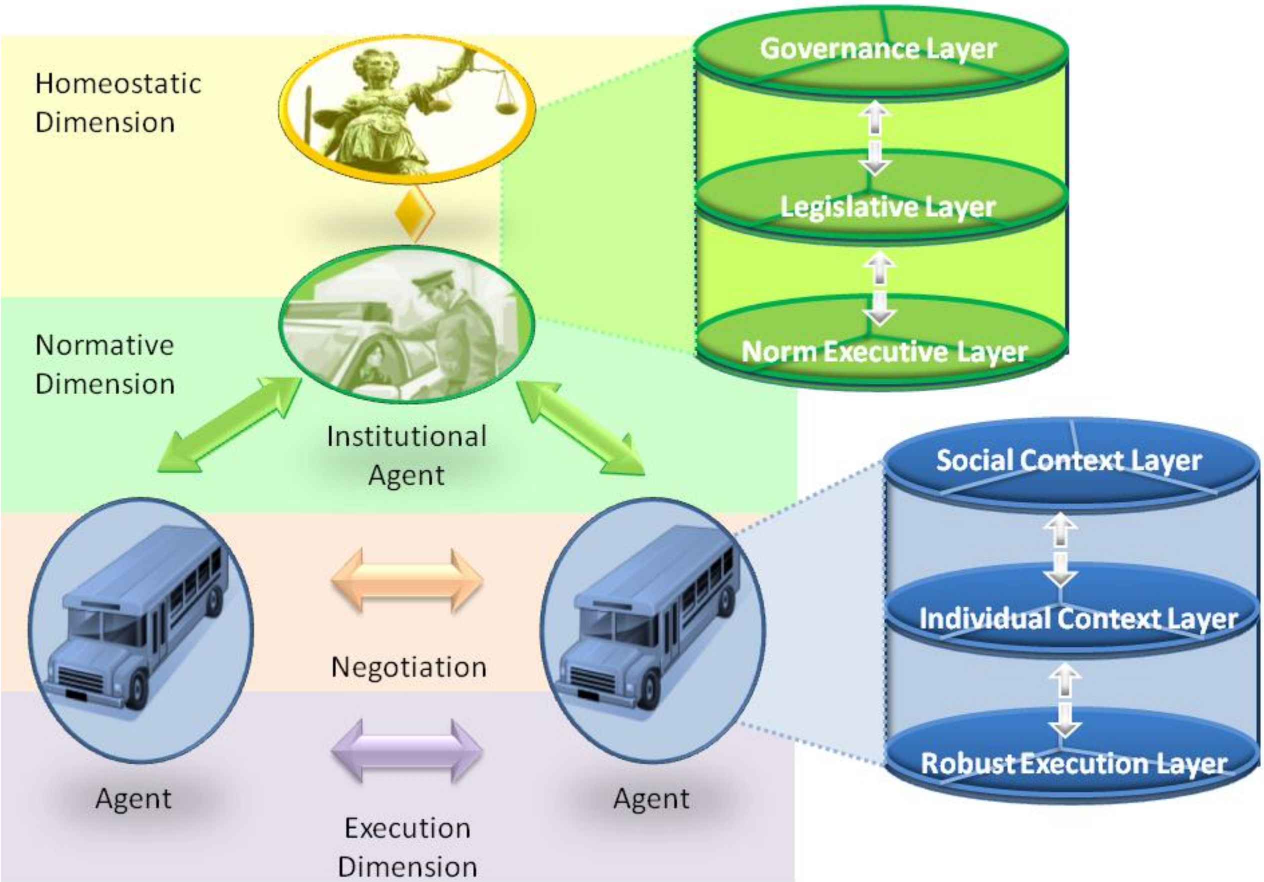

FIGURE 13. AGENTS WITHIN THE DIMENSIONS OF INTERACTION.

3.2 The Negotiation Dimension

The negotiation dimension captures concepts and mechanisms for coordinating agents, including various kinds of agreements and negotiation strategies.
Coordination is the capacity of agents to interact with each other so as to avoid ineffectual behavior by reducing competition for resources *[Wei99]* and to deal with conflicts concerning resources. Coordination entails the mutual (and, where appropriate, temporal) "harmonization" of activities. It takes place independent of individual aims of the agents and comprises control mechanisms for working together. There are two types of coordination:

- Cooperation: Non-antagonistic agents pursue a common aim.
- Competition: Agents pursue different, partly opposing aims.

A conflict exists if intentions and action plans of two or more persons are not compatible with each other and/or are not executable at the same time *[Rüt77, Tho74]*. Conflicts concerning resources (physical and mental) thus exist between several persons *[TCM01]*. For MAS the term conflict is more formally defined by *[PD00]*: Agents possess propositional attitudes (e.g., beliefs, desires, intentions, fears, wishes, etc.) that represent the agents' context at a given time. A conflict arises if an agent must give up a subset (at least one) of his propositional attitudes. This is caused by the behavior of other agents, and a conflict is therefore always a conflict between two goals *[Cas00]*.

Conflicts are perceived as unpleasant and lead to an imbalance in the environment. Nevertheless, conflicts are decisive for the evolution (Among other things, they cause long-term changes) of humans, teams and organizations and must be accepted as part of the development (evolution) of, e.g., a system or a society. When handling conflicts, conflict avoidance is thus not always the best choice. Conflict resolution and conflict management (negotiation) can also be required *[TCM01]*. Distributed conflict handling in MAS can be divided into two classes: On the one hand there is off-line solving for solving combinatorial problem (e.g., constraint satisfaction problems). On the other hand there is real time conflict solving that arises from interacting machines or autonomous systems *[TCM01]*.

3.3 The Normative Dimension

An IT-Ecosystem is seen as a complex system of systems, composed of a large number of autonomous subsystems (agents) which should work together. This implies that the ecosystem is capable of maintaining an functional balance among its components while assuring safety. A particular approach to an IT-Ecosystem is a locality as introduced in section 2.1. The fulfillment of the requirements stated previously requires the existence of scenes where the freedom of decision of an agent is restricted. The description of the expected behavior of those agents is realized by means of the specification of norms.

A norm is an explicit limitation of the autonomous behavior of the agents to an expected regulated behavior. Based on the fact that norms can emerge either top-down or bottom-up, we have split the norms in two categories: hard deontic norms and respective, social norms as illustrated in Figure 14.

Hard deontic norms are specified by a public authority called institution which also provides mechanisms of norm establishment and norm compliance as presented in section 2.4. Considering the deontic like concepts and how crucial is that in certain scenes the norms are respected by the agents, the hard norms have been furthered refined in three classes: permissions, prohibitions and obligations *[VSAD05, ML09]*. Permissions are those norms that specify which actions are allowed to be performed in which state. Prohibitions and obligations are the types of norms that restrict the behavior of an agent. In the case of prohibitions, the agent considers them as guidelines which it may decide to violate. As part of a prohibition, the institution specifies the sanction that will be received by the agent which violates it. The sanctioning could be a notification/warning/punishment from which the agent can learn and re-establish the constraints of its decision making process. Obligations are the type of norms that must never be violated. Therefore, they are imposed on the agent, enforcing to behave according to the norm. The enforcement process is task of an institution in order to ensure norm compliance. As it can be observed, the trade-off between the degree of the autonomy of an agent and the norm compliance is determined by the type of the norm.

In general, the norms that are relevant in a particular situation have to be selected from a collection of norms which are organized in the so-called norm structure. The selection process is controlled by classification schemes that allow for prioritizing between norms and norm refinement. In this way, an agent is supported in its decision which out of a conflicting set of norms is to be violated in a certain situation.

The social norms are regulations on the behavior that have been established bottom-up, i.e., they emerge from social interactions (see section 3.4) *[FF04, TTD09]*.

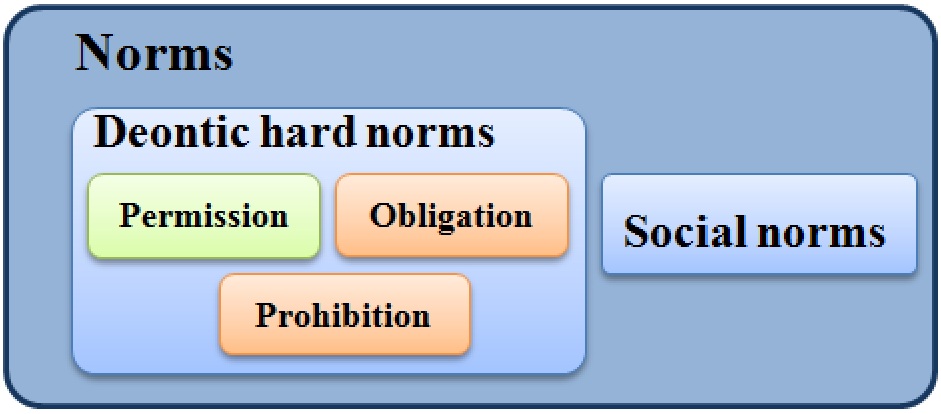

FIGURE 14. TYPES OF NORMS.

Norms may evolve based on complex macroscopic analysis processes of behavioral patterns within a community of agents in a period of time, considering also the changes in the local context or general environment of the locality. The confirmation of a new/adapted or evolved norm is realized through agreements either central, i.e. top-down between public authorities as public organizations or institutions, or decentral, i.e. bottom-up between all agents or participants in the analysis process. If a new norm is introduced by an institution then a specific set of roles has to become aware of its specification in order to adapt their behavior. Based on this, we distinguish between the agents that can recognize norms, sanctions and those that cannot *[SCPP09]*. The capability of norm recognition helps the agent to adapt its behavior successfully and avoid negative sanctioning.

3.4 The Homeostatic Dimension

An ecosystem, as defined in biology and ecology, is "a dynamic complex of plant, animal and micro-organism communities and their non-living environment interacting as a functional unit" *[UN92]*. One of the most important characteristics of an ecosystem is the ability to maintain a functional balance amongst its components, which enables all components to work as a functioning unit. The status of functional balance is also called equilibrium. An equilibrium can correspond to one or more global system states and represents a dynamic balance between interdependent components of an ecosystem. Ecosystems are known to tend towards equilibria, where they are fairly robust against disturbances. Behind this lies a phenomenon called *homeostasis*, which is a mechanism that balances various influences and effects, and opposes changes, so that a stable equilibrium state is maintained. To achieve this, homeostatic mechanisms in nature are mainly understood as feedback loops, which are caused by strong interactions among various components.

In analogy to the concept of the ecosystem in biology and ecology, a working IT ecosystem is based on a functional balance between its technical components. The overall goal in an IT Ecosystem is to influence the components of the system such that a desired functional balance (equilibrium) is achieved. Consequently, we look for a homeostatic mechanism in our IT Ecosystem, in order to achieve and maintain a stable equilibrium state. The homeostatic dimension is responsible for the realization of such a homeostatic mechanism. This mechanism based on feedback loops between different systems components, which must be influenced/controlled in an adequate way so that the whole system stays in a desired equilibrium state. Hence, we have a control problem to solve. In control theory the inputs of a system are manipulated to achieve a desired effect on the output of the system. Similarly, in an IT ecosystem, the output of the system is the global system state, and the inputs are all intervention possibilities, which influence the component behavior. The difference in IT ecosystems is that we cannot use standard control engineering approaches to solve this problem, since we do not have a clear mathematical definition of the input and output description, system dynamics, etc. in an IT Ecosystem *[And06]*.

Furthermore, IT ecosystems are composed of a large number of autonomous, distributed, heterogeneous subsystems (agents), which are coupled in various ways. Due to the high variety of interaction between the subsystems, one cannot treat individual systems in isolation. Instead, they should be seen as part of an IT ecosystem. As a result, interactions play a crucial role in terms of controlling the system, because the global system state emerges as a side effect of the interactions of subsystems. Therefore, if we want the IT ecosystem to achieve a stable equilibrium state, we also have to influence these interactions using a homeostatic control mechanism. Such a mechanism can be realized using the notion of norms and institutions to restrict the autonomy of agent and influence system behavior (section 3.3). Generally ***[DD09]***, norms can be implemented using **(1)** a top-down approach where higher-level authorities (institutions) impose norms (deontic norms) to agents and enforces them towards the desired behavior or **(2)** a bottom-up approach where norms (social norms) emerge from the interactions between agents ***[MSA08]*** or **(3)** a combination of top-down and bottom up approaches.

We can now define equilibrium as a state where all norms are satisfied and a functional balance between the system components is achieved. Changes in the environment, addition or removal of agents, behavioral modifications of the agents or new high-level goals will trigger the homeostatic mechanism to find a new equilibrium or to return to the old one.

4 EVALUATION OF THE METAMODEL AND ARCHITECTURE ON A SCENARIO

4.1 An Airport Scenario

Description of the scenario:

The transportation service in an airport area is supposed to be done by an amount of autonomous transport vehicles (i.e. robots). In general there is a departure and an arrival part of the scenario, for now we just apply the departure section. Typical departure elements in an airport area are

- the entrances, where the people arrive at the airport
- the check-in terminals
- some gates
- several areas in the apron of the airport, where the airplanes are located

The transport robots are equipped with batteries, so they have to recharge themselves after some time in one of the recharging stations, which are also represented in the scenario. In addition to that, we have several two-lane roads, which connect the above mentioned elements. To avoid congestion and blockages, there are some side roads which can be used instead of the main roads (with a reduced vehicle speed though). In a real scenario we have both, a transportation of passengers as well as a transportation of baggage, though in this evaluation we focus on the former. Thus, the transportation service of the transport vehicles contains the following steps:

- passenger transport from an airport entrance to one of the five check-in terminals
- passenger transport from a check-in terminals to one of the four gates
- passenger transport from a gate to the correct parking position of the respective plane

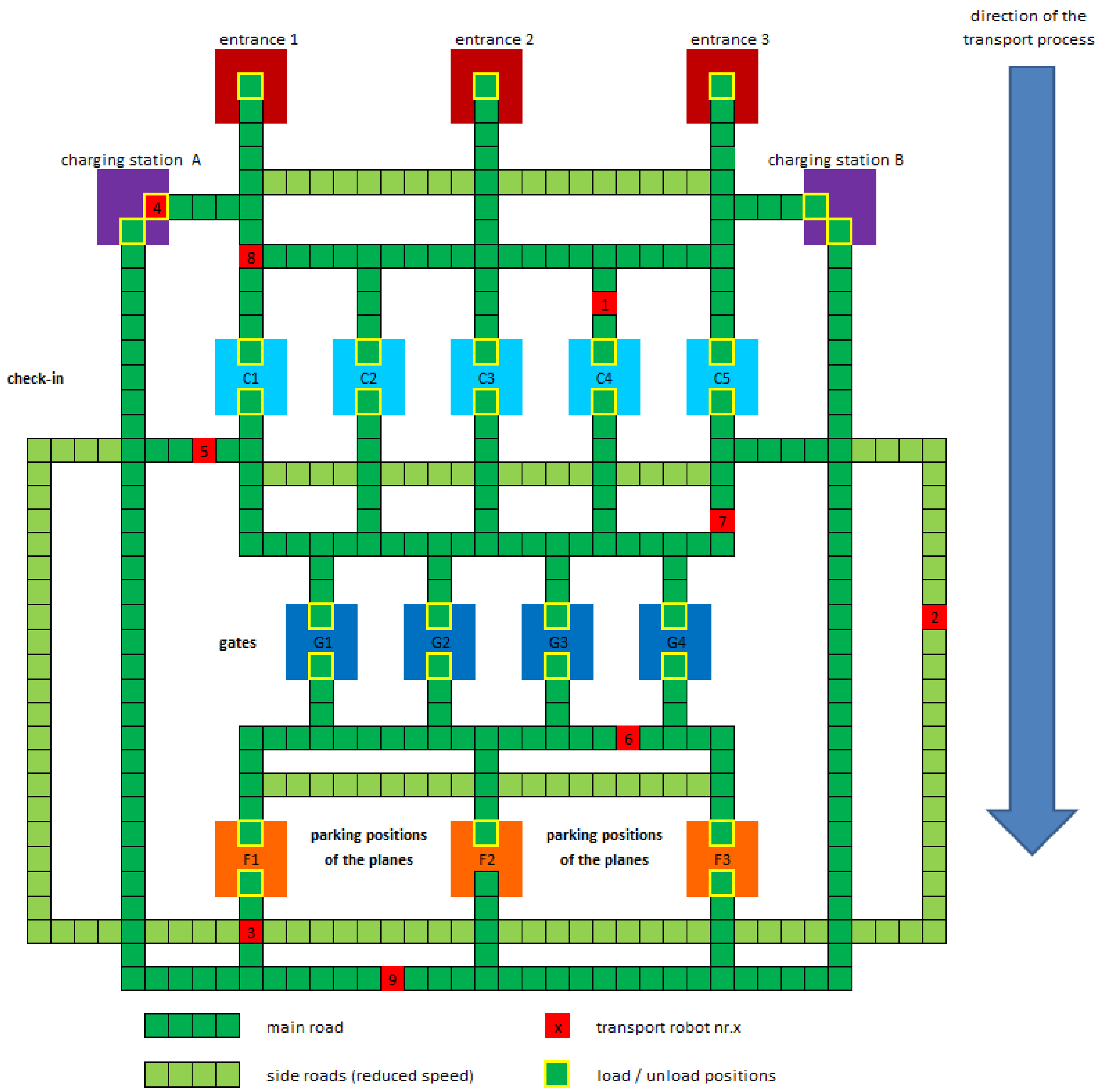

FIGURE 15. AIRPORT SCENARIO (DEPARTURE).

In order to apply the concepts of the AIM metamodel we suppose, that the transport robots belong to three different competing transport providers. To control the behavior and the interactions between the robots, there are some institutions, which are responsible for the generation and compliance of norms, as well as they care about norm violations and impose sanctions on violating robots. Examples for regulating norms are speed limitations or an intermittent one-way demand for a certain road. The appliance of these concepts is specified in the following section.

An exemplary infrastructure for airport transportation that we use in a simulation framework combining Repast Simphony *[Rep10]* and Jade *[Jad10]* is shown in Figure 15. The implementation of the airport scenario in the simulation framework strictly makes use of the A²OL concepts as they were introduced before. Hence, it can serve as a first evaluation whether the concepts are appropriate to build an IT-ecosytem upon them. First observations from the scenario implementation are reported in the following paragraphs:

4.2 Evaluation of the Metamodel on the Airport Scenario

4.2.1 Norms

The projection of the locality in this scenario is represented by the airport itself. In this location, we assume the existence of several autonomous agents that should interact and work together. To ensure that the interaction is successful, we have to verify first their compatibility (e.g. proof of the communication protocol between two agents). An example of autonomous agents are transport vehicles which are in charge with carrying persons and luggages from the entrance through all the via points required for the embarkation (e.g. check-in, gate, airplane location). Their autonomy property can be seen also as a degree of selfishness, which can lead to unsafe situations. In order to mitigate and avoid safety-critical situations arising from their autonomy, we have introduced norms as a mean to regulate the behavior of the agents in the airport. As stated in the section 3.3, there are norms introduced by an institution as obligations, permissions and prohibitions and norms that emerge bottom-up based on agreements between agents. For the moment, we have considered formalizing the first category extending the work of Vázquez-Salceda by introducing new operators in the specification of the norm condition. In order to find an answer to the raise questions as "how agents' behavior is affected by norms?" and "how to enforce a norm?" which have been also considered in *[VSAD05]*, we have started with obligation. The reason for this choice is the fact that obligation must be imposed and therefore they bring the most weight on the limitation of agent behavior. More concrete, we have formalized the obligations concerning agents performing a certain sequence of actions (with controlling automata). In this way we ensure that the resulted behavior of the agent after enforcement is compliant with the norm. For the first steps in the implementation of the scenario we have assumed that the agents are aware of the norms and also their compliance has been hard coded in the agent behavior. Therefore, we propose for the next steps to generalize not only the specification of norms but also the enforcement process which should consider verifying norm compliance based on the vocabulary of the norms and agent behavior described with the associated ontologies.

4.2.2 Agreements

In the scenario we assume the existence of a variable number of autonomous agents (transport vehicles) and transfer orders (persons and luggage). At the airport entrance, the agents receive orders that have to be completed in a certain time. With success/failure they receive a positive/negative reward. Loaded agents drive slower than non-loaded agents. Furthermore, agents lose energy when driving and have to use charging stations for recharging. In the first step we investigate traffic jams (e.g., how do they arise?) on the basis of these assumptions. Changing the variables "number of agents" and "number of orders" will permit conclusions regarding their impact on such traffic jams.

In the next step turnouts are integrated along the paths, where slow agents can let other (faster) agents pass (cf. Figure 16). The question arises regarding what motivates the agents to pull over. Different coordination approaches are to be examined and evaluated (time/energy):

- Agents always let faster agents pass, if possible.
- Centralized planning
- History (Last time you let me pass, now I will let you pass)
- Institutional
- Priorities
- Sharing rewards (Clarke Grove)

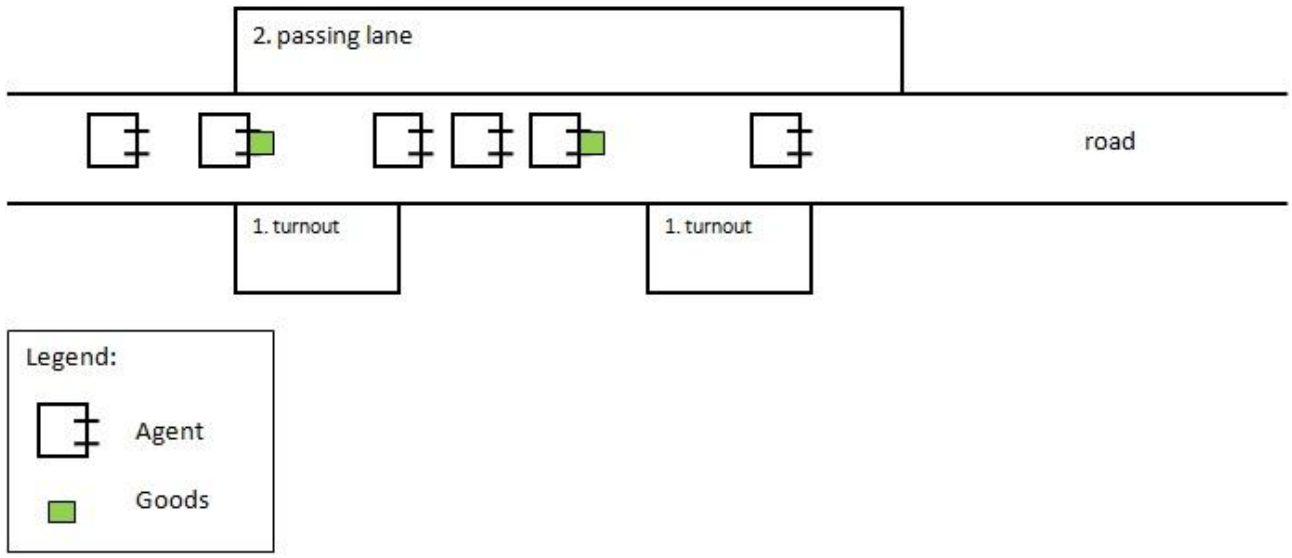

FIGURE 16. SCENARIO WITH PASSING LANES AND TURNOUTS.

In the third step, passing lanes are introduced (cf. Figure 16). Now agents can decide whether they pass other agents. It is assumed that agents need more energy when passing. The question arises, whether in this case it continues to be necessary that agents coordinate amongst themselves.

4.2.3 Planning and Task Execution

In the context of norm application, we focus on the decision making process at crossroads and junctions in the scenario section between check-in terminals and gates. Due to the symmetry of this section, the sensor data (information about the adjacent cells) of one robot is not sufficient for making the correct decision. Accordingly the missing information is supposed to be mediated by the use of different levels of interaction mechanisms (i.e. in this case communication). The intensity of these interactions is controlled by so called "interaction intensity institutions" and corresponding norms:

- exchange of the explicit path information
- fusion of sensor data
- exchange of information about already finished transport orders and chosen paths

5 CONCLUSION

We have introduced a conceptual metamodel to clarify the fundamental concepts of autonomous agents in organized localities. We classified the interactions in such systems in four dimensions, namely task execution and planning, negotiations, the normative, and the homeostatic dimension. The classification facilitates the separation of concerns and has lead to a first, still abstract version of layered architectures for agents and institutions. In the moment, the A²OL concepts are evaluated by realizing a simulation model for the airport departure scenario in a framework combining Repast Simphony ***[Rep10]*** and Jade ***[Jad10]***.

Organizations as well as institutions and norms as a mean to impose superordinate structure and behavioral regulations on agent communities are an aspiring research area in multi-agent systems ***[CSB05, BPD+06, Dig04, ROD05, GV06]***. Most aspects of organized and normative agents have already been addressed in different works in the literature. However, they are seldom considered in combination and a number of facets differentiate between our approach and others:

- We consider two dimensions of equilibria for the autonomous agents in organized localities: **(1)** the equilibrium between agents or agent organizations and the balancing between their conflicting interests. In addition we want to consider **(2)** the equilibrium between superordinate control and individual autonomy that is arises from the degree of regulative enforcement that is imposed by institutions on the agents. The extreme positions at this dimension are easily identified: On the one side we have total control of the institution which for instance can be realized by institutional agents observing the situation and controlling each single action of an agent. The other extreme may be an institution that just informs the agents about the preferred behavior but leaves it to them to decide. The challenge is to find a good choice in between that ensures that the agents respect the obligations of the locality, but also allows for a sufficient amount of autonomy.

- Although we are realizing our case study on the airport departure scenario in a simulation framework at first, the metamodel and the architecture is planned as a basis for embedded agents, e.g. for autonomous vehicles. Thus the productive execution layer is concerned with performing physical actions which is more complex than the activities of pure software agents. For physical agents, the productive execution layer is an embedded real-time layer that has to satisfy the classical attributes of dependability in the context of limited resources to provide means like fault prevention, fault tolerance and fault recovery.

- Humans are an integral part of an IT Ecosystem, they continuously interact with the agents by requesting a service from an agent or an agent organization but also on the execution layer when they enter a vehicle for instance. However, an order should specify *what* has to be achieved but not the detailed plan *how* to achieve it, to enable plan selection by the agents.

REFERENCES

[AEN+05] Josep Lluis Arcos, Marc Esteva, Pablo Noriega, Juan A. Rodríguez-Aguilar, and Carles Sierra. An integrated developing environment for elec-tronic institutions, pages 121 – 142. Birkhäuser Publisher, 2005.

[And06] Neculai Andrei. Modern control theory - a historical perspective. In Studies in Informatics and Control, volume 10, pages 51–62, March 2006.

[BPD+06] Olivier Boissier, Julian A. Padget, Virginia Dignum, Gabriela Lindemann, Eric T. Matson, Sascha Ossowski, Jaime Simão Sichman, and Javier Vázquez-Salceda, editors. Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems, volume 3913 of Lecture Notes in Computer Science. Springer, 2006.

[Cas00] C. Castelfranchi. Conflicts in Artificial Intelligence, chapter Conflicts Ontology, pages 21–40. Kluwer, Dordrecht, 2000.

[CSB05] L. Coutinhoa, J. Sichman, and O. Boissier. Modeling organization in MAS: a comparison of models . 1st. Workshop on Software Engineering for Agent-Oriented Systems (SEAS'05), October 2005.

[DD09] V. Dignum and F.P.M. Dignum. Emergence and enforcement of social behavior. In 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation, pages 2377–2383, 2009.

[DGG+09] A. Duraslan, G. Ganceanu, U. Goltz, B. Hammer, M. Huhn, C. Knieke, B.Mokbel, J.Müller, C.Müller-Schloer, C.Mumme, and C. Schulz. Adaptive Interaktionsmechanismen: Technical concepts, solution approaches, and methods (deliverable e-aim-1). Technical report, NTH School für IT-Ökosysteme, 2009.

[Dig04] Virginia Dignum. A Model for Organizational Interaction: Based on Agents, Founded in Logic. PhD thesis, 2004.

[FF04] Ernst Fehr and Urs Fischbacher. Social norms and human cooperation. Technical report, EconWPA, 2004.

[FGM03] Jacques Ferber, Olivier Gutknecht, and Fabien Michel. From agents to organizations: An organizational view of multi-agent systems. In AOSE, pages 214–230, 2003.

[Fis09] Christian Hahn; Cristián Madrigal Mora; Klaus Fischer. A platformindependent metamodel for multiagent systems. Autonomous Agents and Multi-Agent Systems, 18(2):239–266, 4 2009.

[GV06] Holger Giese and Alexander Vilbig. Separation of Non-Orthogonal Concerns in Software Architecture and Design . Software and SystemModeling (SoSyM), 5(2):136 – 169, 2006.

[Jad10] Jade. Java agent development framework. http://jade.tilab.com/, 2010.

[Jen00] Nicholas R. Jennings. On agent-based software engineering. Artificial Intelligence, 177(2):277–296, 2000.

[Mül96] Jörg P. Müller. The Design of Intelligent Agents: A Layered Approach. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996.

[ML09] Felipe Meneguzzi and Michael Luck. Norm-based behaviour modification in bdi agents. In AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, pages 177–184, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[MSA08] Partha Mukherjee, Sandip Sen, and Stéphane Airiau. Norm emergence under constrained interactions in diverse societies. In AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems, pages 779–786, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[PD00] P. Pasquier and F. Dehais. Approche générique du conflit. In ErgoIHM 2000, 2000.

[PR90] Martha E. Pollack and Marc Ringuette. Introducing the tileworld: Experimentally evaluating agent architectures. In In Proceedings of the Eighth National Conference on Artificial Intelligence, pages 183–189, 1990.

[Rep10] Repast Simphony: Recursive porous agent simulation toolkit. http://repast.sourceforge.net/, 2010.

[RG91] Anand S. Rao and Michael P. Georgeff. Modeling rational agents within a bdi-architecture. In James Allen, Richard Fikes, and Erik Sandewall, editors, Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91), pages 473–484. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1991.

[ROD05] Rossella Rubino, Andrea Omicini, and Enrico Denti. Computational Institutions for Modelling Norm-Regulated MAS: An Approach Based on Coordination Artifacts. In AAMAS Workshops, pages 127–141, 2005.

[Rüt77] B. Rüttinger, editor. Konflikt und Konfliktlösen. Rosenberger Fachverlag, München, Germany, 1977.

[SCPP09] B. T. R. Savarimuthu, S. Cranefield, M. Purvis, and M. Purvis. Internal agent architecture for norm identification. In Proc. of the international workshop on Coordination, Organization, Institutions and Norms in agent systems, pages 156 ˝ U–172, 2009.

[TCM01] C. Tessier, L. Chaudron, and H.J. Müller, editors. Conflicting agents: conflict management in multi-agent systems. Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[Tho74] H. Thomae, editor. Konflikt, Entscheidung, Verantwortung. Kohlhammer, Stuttgart, Germany, 1974.

[TTD09] Shingo Takahashi Takao Terano, Hajime Kita and Hiroshi Deguchi. Agent-Based Simulation of Learning Social Norms in Traffic Signal Systems. Springer, 2009.

[UN92] UN. Convention on biological diversity, 1992.

[VSAD05] Javier Vázquez-Salceda, Huib Aldewereld, and Frank Dignum. Norms in multiagent systems: from theory to practice. Comput. Syst. Sci. Eng., 20(4), 2005.

[Wei99] Gerhard Weiss, editor. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, 1999.

[Woo09] Michael Wooldridge. An Introduction to MultiAgent Systems. Wiley, 2nd edition, 2009.