MAS Modelling based on Organizations

Estefania Argente, Vicente Julian, and Vicente Botti *

Departamento de Sistemas Informaticos y Computacion,
Universidad Politecnica de Valencia,
C/Camino de Vera s/n, 46022, Valencia, SPAIN
{eargente,vinglada,vbotti}@dsic.upv.es

Abstract. An agent organization modelling is proposed based on four main concepts: organizational unit, service, environment and norm. Those concepts are integrated in ANEMONA meta-models, which are extended in order to include all entities needed for describing the structure, functionality, dynamicity, normativity and environment of an organization.

Key words: Meta-model, Organization

1 Introduction

Organizational models have been recently used in agent theory for modelling coordination in open systems and to ensure social order in MAS applications [1]. Agent Organizations rely on the notion of openness and heterogeneity and include the integration of organizational and individual perspectives and the dynamic adaptation of models to organizational and environmental changes [2].

Meta-modelling is a mechanism that allows defining languages of modelling in a formal way, establishing the primitives and syntactical-semantical properties of a model [3]. For example, INGENIAS [4] and ANEMONA[3] methodologies offer several meta-models for analysis and design of MAS, by means of thei component description (organizations, agents, roles); functionality (goals and tasks); environment (resources and applications); interactions and agent internal features, such as autonomy and mental state processing. INGENIAS follows an iterative development process based on *Rational Unified Process* (RUP). It is supported by powerful tools for modelling, design and code generation. ANEMONA, based on INGENIAS, is a MAS methodology for developing Holonic Manufacturing Systems. They both employ UML notation language for describing their meta-models, following GOPRR [5] restrictions. However, they lack of a specific normative description, a deeper analysis of the system dynamics and an open system perspective.

In our work, ANEMONA meta-models have been extended in order to mainly include the concepts of organizational unit, service and norm. Those concepts have been extracted from human organizational approaches [6–8] and also from multiagent system works [9, 10] and service oriented architecture[1]. They are

* Consolider Ingenio 2010 N.26706

[1] http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf

used for representing: (i) how entities are grouped and connected between them and their environment; (ii) which functionality they offer, and which services are used for dynamical entry/exit of agents in the organization; and (iii) which restrictions are needed for controlling entity behaviour inside the system.

The proposed MAS modelling employs six different meta-models: the *organization meta-model*, that describes which are the entities of the system (agents, organizational units, roles, norms, resources, applications) and how they are related (their social relationships; functionality needed or offered); the *activity meta-model*, that details the specific functionality of the system, based on services, tasks and objectives; the *interaction meta-model*, that defines system interactions, activated by means of objectives or service usage; the *environment meta-model*, that describes system applications and resources, agent perceptions and effects and also service invocation through its ports; the *agent meta-model*, that describes concrete agents, their responsibilities, objectives, services, tasks, played roles, known norms and reasoning mechanisms; and finally the *normative meta-model*, that details organizational norms and normative objectives that agents must follow, including sanctions and rewards.

A case-study example is used for a better comprehension of the meta-models, based on the travel domain. Hotel chains and flight companies offer information about their products (hotels, flights), booking facilities and advance payment. Their functionality is defined using services and controlled with norms that describe, for example, which are the minimal services that providers must register in the system in order to participate inside; how services are described (services profiles and processes); or in which order services must be served.

In this paper, the main extensions to ANEMONA meta-models are related. UML notation language is employed, following GOPPR restrictions. All relationships have a specific prefix that indicates: O for organization; GT for goals and tasks; WF for work flow; AGO for social relations; E for environment; N for norms and I for interactions. All meta-model extensions are graphically emphasized in dark colour. Due to lack of space, only those meta-models with more extensions are explained. More specifically, a description of the *organization meta-model* is detailed in section 2; how services are described using the *activity meta-model* is explained in section 3; extensions to the *environment meta-model* are shown in section 4; whereas section 5 describes how rules are modeled using a *normative meta-model*. Finally, conclusions and discussion are detailed in section 6.

2 Modelling MAS Organizations

An agent organization is defined as a social entity composed of a specific number of members that accomplish several distinct tasks or functions and are structured following some specific topology and communication interrelationship in order to achieve the main aim of the organization [11]. Agent organizations assume the existence of global goals, outside the objectives of any individual agent, and they exist independently of agents [2].

ANEMONA meta-models offer the Abstract Agent (A-Agent) notion [3], that allows defining agent collections as unique entities of a high-level description considered as a complete and unique entity that acts as if it where an agent, which can be later refined and specified internally, defining all its components (simple agents or groups of agents). Thus, an A-Agent is defined in a recursive way, so it can be both an atomic entity or a multi-agent system (with unique entity) composed of A-Agent not necessarily equal.

This A-Agent entity has been extended with the **Organizational Unit** concept, that describes the existing groups of members of the organization. Those units have a specific internal structure; they also define several roles or positions that describe a set of functionality (services offered and required) and goals that represent organizational expectative for each position; they include resources and applications, that can be accessed by specific members of the organization; and they define some norms that control their member behaviour.

The proposed *organization meta-model* integrates this *Organizational Unit* concept and contains four views: structural, functional, social and dynamic. The first three ones are extensions of those employed in ANEMONA, whereas the new dynamic view is used for specifying which are the services that an organizational unit must offer for controlling and managing entry and exit of entities.

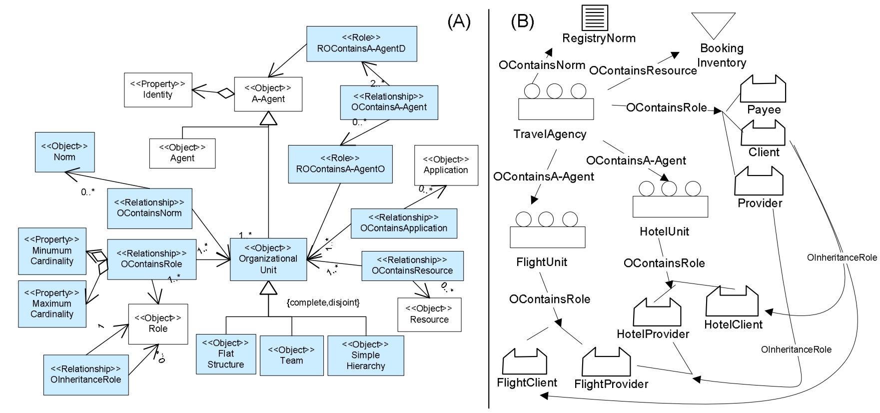

Fig. 1. (A) Organization Meta-model. Structural view; **(B)** Example of Organizational model diagram (structural view) for the travel agency case-study.

The **structural view** defines which are the "static" components of the organization, i.e. all those elements that are independent of the final executing entities (figure 1.A). Thus, the system is composed of *Organizational Units*, which sometimes might be considered as a global entity (acting as A-Agents) or as part or group entity of the organization. Moreover, in a recursive way, Organizational Units can also include other units. Internally, its members are related by means of a hierarchy, team or plain structure. The composition of these units facilitates designing more complex and elaborated structures, such as matrix, federation, coalitions or congregations [11]. The Organizational Unit acts as a group of agents (*OContainsA-Agent* relationship), but also as their environment. Thus, it contains both resources and applications that can be used by them (*OContainsResource* and *OContainsApplication* relationships). It also defines which are the roles inside the unit (*OContainsRole* relationship) and all norms that control their behaviour (*OContainsNorm* relationship).

In the travel case study (figure 1.B), the *TravelAgency* organizational unit represents the whole travel system. The *Client* role represents the final user that requests information about hotels, flights; orders booking rooms or flight seats; and even might pay in advance. The *Provider* role offers searching and booking service functionality. Finally, the *Payee* role is responsible of controlling the advance payment. As description and functionality for travel search and booking might be different for hotels and flights, two organizational units (*FlightUnit* and *HotelUnit*) have been defined, focused on their specific products.

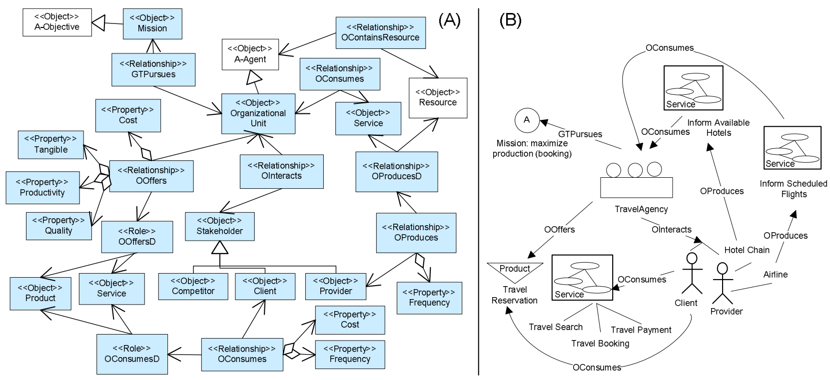

Fig. 2. **(A)** Organization Meta-model. Functional view. Mission; **(B)** Example of Organizational model diagram (mission) for the travel case-study.

The **functional view** describes the organizational mission; and how each organizational unit behaves, both externally and internally.

The *mission* (figure 2.A) defines global goals (*GTPursues mission*), who are the stakeholders that interact with the organization (*OInteracts*), which are their results (*OOffers* products or services), how they are consumed by clients (*OConsumes*) and what the organization needs from its providers (*OProduces* services or resources; *OConsumes* services; *OContainsResource*). In the case-study example (figure 2.B) the system (*TravelAgency* unit) offers the travel reservation product, consumed by its clients (tourists or businessmen). It also offers several services for travel searching, booking and payment. On the other hand, this system requires that some providers (hotel chains and airlines) supply all needed information about hotels and flights.

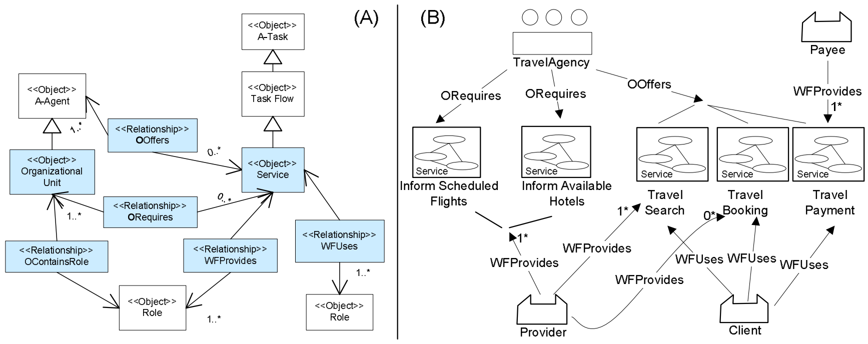

Fig. 3. (A) Organization Meta-model. Functional view. External functionality.; **(B)** Example of Organizational model diagram (external func.) for the travel case-study.

The *external functionality* of an A-Agent (figure 3.A) represents the set of services that this entity offers to other A-Agents (*OOffers* relationship), independently of the final agent that makes use of them. Moreover, a set of services required by Organizational Units can also be defined. Those services represent all functionality that needs to be "hired" to other A-Agents. The *ORequires* relationship is similar to "job offer advertising" of human organizations, in the sense that it represents a necessity of finding agents capable of providing those required services as members of the unit. This *ORequires* relation is related with the *OConsumes* relation of *mission* (fuctional view). All features, abilities and permissions of providers and clients of those services are modelled by means of roles, using *WFProvides* and *WFUses* relationships.

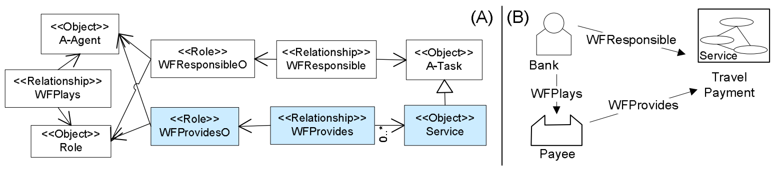

Fig. 4. (A) Organization Meta-model. Functional view. Internal functionality.; **(B)** Case-study example. The *Payee* rol is played by the *Bank* agent.

In the travel case-study example (figure 3.B), the *TravelAgency* unit offers *TravelSearch*, *TravelBooking* and *TravelPayment* services to agents playing the *client* role. Moreover, *provider* agents must supply at least an information service, invoked in the *TravelSearch*. Thus, any agent willing to play a provider role has to be capable of providing a service of this kind. However, the *TravelBooking* service is not compulsory, so providers can freely decide whether to offer it or not. The *TravelPayment* service is assigned to the *Payee* role.

Finally, the *internal functionality* of an A-Agent (figure 4.A) is defined by its tasks, which are delimited by the roles that the entity plays and the services provided by this role. For example, the *Bank* agent (figure 4.B) plays the *Payee* role in the travel case-study, implementing the *TravelPayment* service functionality.

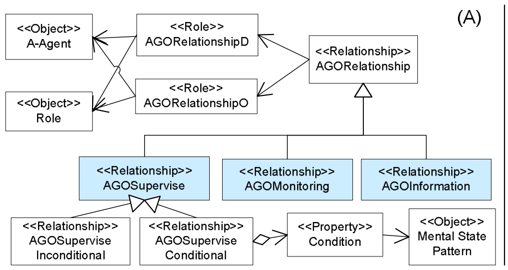

Fig. 5. (A) Organization Meta-model. Social view; **(B)** Example of Organizational Model diagram (social view) for the travel agency case-study.

The **social view** (figure 5.A) describes roles and A-Agent social relationships, divided into three types: supervision, monitoring and information.

The *AGOInformation* relationship describes how information or knowledge links are established inside the organization. If two A-Agents are connected with this type of link, then they are entitled to know each other and communicate relevant information. The *AGOMonitoring* relationship implies a monitoring process of agent activity, so the monitor agent is responsible of controlling tasks of its monitorized agents. Finally, the *AGOSupervision* relationship implies that a (supervisor) agent transfers or delegates one or more objectives to its subordinate agent, which is obliged to include those objectives as own and pursue them. This social view extends ANEMONA one and is also based on [12] and [13] works.

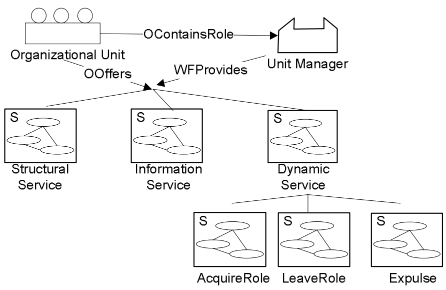

Fig. 6. Pattern Design for the dynamic view (Organization Model).

In the travel case-study example (figure 5.B), the *FlightUnit* has been modelled using a hierarchical structure in which there is a supervisor (*manager* role) that receives all flight requests from clients; and invokes *FlightProvider* services, also controlling their behaviour.

The **dynamic view** (figure 6) defines the pattern designs for organizational unit services, that enable managing all its structural and dynamic components. Those services are divided into structural, informative and dynamic services. The *structural services* are focused on adding or deleting norms, roles or organizational units. The *informative services* provide information about the structure of the organization. And the *dynamic services* manage the inclusion and exit of agents into the unit and the role adoption. Those last services need to be published in an open system for allowing external agents to participate inside.

3 Modelling MAS Services

Services represent some functionality that agents offer to other entities, independently of the concrete agent that makes use of it. The main features of services are: (i) synchronization, that implies interaction between entities that offer the service and those ones that require and use it; (ii) publishing, so the service is registered in a service directory and other entities can find it; (iii) participation, so then entities that consume the service can differ through time; (iv) entity standardization, as service consumers and providers are related to specific roles, for which restrictions are defined through norms, resource access permissions, etc.; (v) functionality standardization, as services are described in terms of inputs, outputs, preconditions and postconditions, making easier the description of the service functionality; (vi) tangibility, as services usually produce tangible products which can be employed for evaluating both quality, service efficiency and client satisfaction; and (vii) cost, so service production and consumption imply several costs and/or benefits.

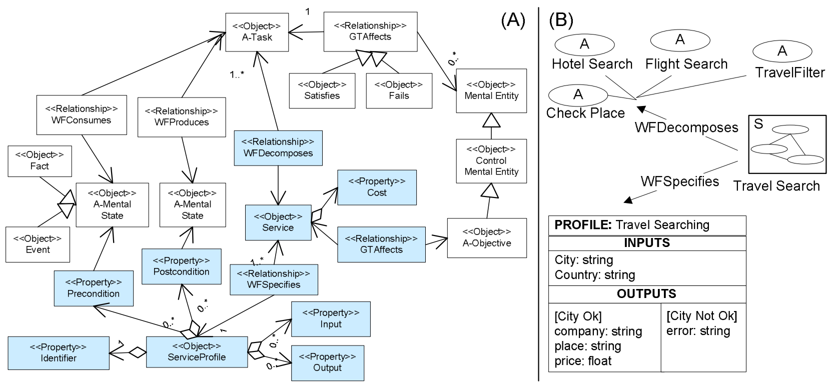

Fig. 7. (A) Activity Meta-model. Service description.; **(B)** Activity model diagram for *TravelSearch* service of travel case-study example.

In the **activity meta-model** (figure 7.A), service system functionality is described by means of their profiles and A-Tasks in which they are decomposed. The *ServiceProfile* concept describes activation conditions of the service, its input, output parameters and also its effects over the environment. It can lately be used in an OWL-S service description. The *A-Task* concept (figure 8.B) describes the service functionality or processes. It represents both concrete tasks, task-flows or service composition (*WFInvokes* relationship). A task-flow description (figure 8.A) relates connections between tasks and their environment.

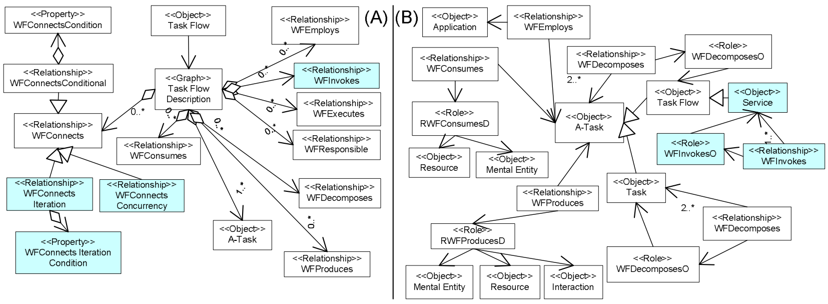

Fig. 8. (A) Activity Meta-model. Task Flow Description.; **(B)** Activity Meta-model. Task description.

The activity model diagram for *TravelSearch* service of the case-study example is shown in figure 7.B. This service is described using the "Travel Searching" profile and contains four tasks: *CheckPlace*, that checks inputs (country and city); *FlightSearch* and *HotelSearch* (concurrent tasks that invoke *InformScheduledFlights* and *InformAvailableHotels* services, respectively); and *TravelFilter*, that selects best hotels and flights.

4 Modelling MAS Environment

Based on human organizations [14, 8] environment should be modelled with two different perspectives: structural and functional. The structural perspective describes which are the components of the system (agents, objects, resources); how they are related (i.e. agent groups, behavioural norms, resource access); and how those elements are conceptually represented, by means of an ontology. The functional perspective determines which are the activities related with the environment, i.e., how agent communication is produced (direct or indirect messages, using specific environment elements...); how agents can perceive and act over the environment and how agents are connected with other types of entities such as objects, applications or resources.

The proposed *environment meta-model* (figure 9.A) focuses on the description of its elements (resources, applications and mental entities); perceptions and actions over the environment; and permission accesses for using those elements.

Agent perceptions and actions are described using the *EnvironmentPort* concept, which is a specialization of *Port* entity. This concept has been extracted from AML language modelling [10], in which a port represents an interaction point between an entity and other model elements. Two kinds of ports have been defined: environment and service ports. The environment port allows lecture and/or write access to resources or applications. The *Perceptor* port establishes how agents can obtain information from resources and applications. The *Effector* port allows agents to modify resource data. The *EContainsPort* relationship indicates who manages and controls the environment port access. The *WFEmploysPort* relationship represents which roles are allowed to use the port and in which way (*WFEmploysReadPort* for obtaining information; *WFEmploysWritePort* for creating or modifying environment information).

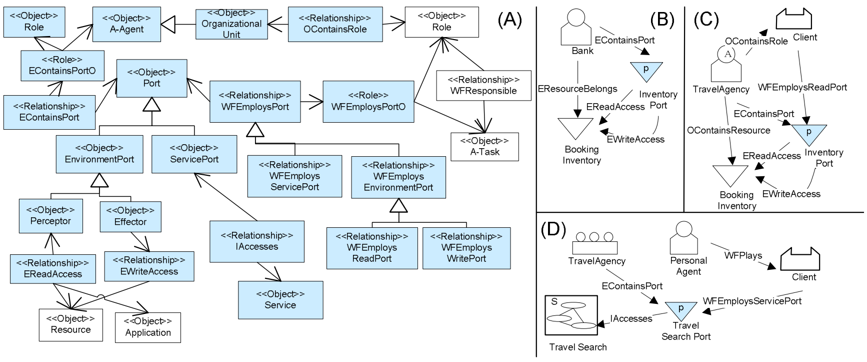

Fig. 9. (A) Environment meta-model. Port access.; **(B)** Case-study example: the *bank* agent contains the *Booking Inventory* and manages its access; **(C)**The *TravelAgency* unit contains the *Booking Inventory* and manages its access; **(D)** The *TravelAgency* unit publishes *TravelSearch* service, which is used by agents playing *client* role.

The *ServicePort* concept represents the publishing feature of the service, i.e. the contact point or grounding mechanism for service access. The entity in charge of publishing it (in a service directory, for example) is represented with the *EContainsPort* relationship.

For the travel case-study, an example of a resource belonging to a specific agent is shown in figure 9.B, in which the *Bank* agent controls access to the *Booking Inventory* by means of the *Inventory Port*. However, in many problems the resource does not belong to a specific agent, but to the environment of a group of agents. In this case, the organizational unit that represents this group contains this resource and manages its access through a resource port. For example, in figure 9.C, the *TravelAgency* unit contains the *Booking Inventory* resource, which can be read or modified; but the *client* role defined in this unit is only empowered to read access. Finally, an example of a service port access is shown in figure 9.D, in which *PersonalAgent* playing the *client* role is allowed to make use of the *TravelSearch* service. The *TravelAgency* unit is in charge of publishing this service (represented by the *EContainsPort* relationship).

5 Modelling MAS Norms

Norms have been widely used as mechanisms to limit human autonomy inside societies, in order to solve problems of coordination, specially when total and direct social control cannot be exerted. In open multi-agent systems, norms have been considered as a key issue for managing the heterogeneity, autonomy and diversity of interests of agents [15].

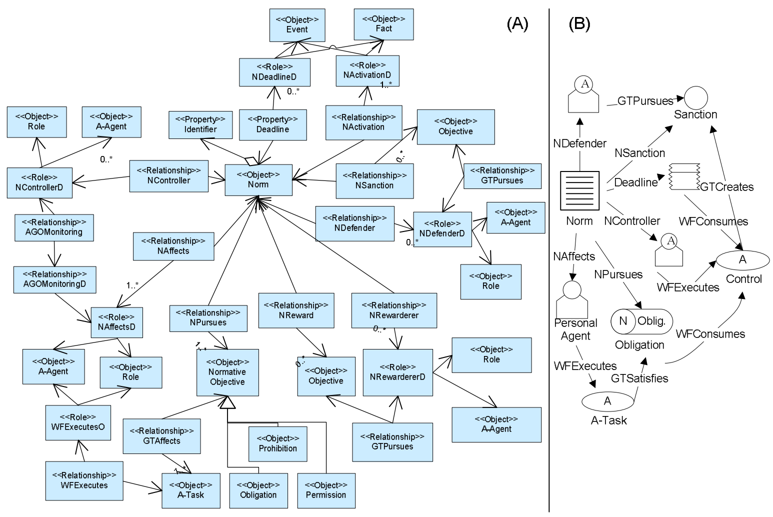

Fig. 10. (A) Normative meta-model; **(B)** Pattern design of an obligation.

The proposed **normative meta-model** (figure 10.A) describes the **Norm** concept, which represents a specific regulation, expressed by means of a *Normative Objective* (Obligation, Permission or Prohibition). This regulation affects A-Agents or Roles (*NAffects* relationship), whose actions are controlled by the normative objective (*WFExecutes* and *GTAffects* relationships). The norm also indicates who is in charge of monitoring that the norm is satisfied (*NController* relationship); and who is responsible of punishments (*NSanction* and *NDefender* relationships) and/or rewards (*NReward* and *NRewarderer* relationships). Finally, *NActivation* relationship specifies all facts and events of the environment that provoke the activation of the norm. Its deactivation is produced when the normative objective or the *deadline* are satisfied.

In figure 10.B, the pattern design for an obligation norm is shown. A sanction is created when deadline has reached and compulsory tasks have not been satisfactorily executed.

6 Discussion

An extension of ANEMONA meta-models has been proposed, in order to include concepts of organizational unit, service and norm. Those concepts have been extracted from human organizational approaches, from multiagent systems works and from service oriented architectures; being integrated in a framework for modelling organizations. In this way, the main features of an organization can be described: its structure, functionality, dynamics, environment and norms. Thus, the *organization meta-model* describes its components, relationships and connections with its environment. The *activity meta-model* details offered and required services, their tasks and objectives. The *environment meta-model* captures system resources and applications, agent perceptions and effects and port accesses permissions. Moreover, organization rules are expressed with the *normative meta-model*. Finally, the *agent meta-model* details concrete responsibilities of agents and their internal functions; and the *interaction meta-model* defines specific interactions between agents and service invocation (using service ports).

Regarding related work on MAS organizational modelling, AGR model [16] is based on agent, groups and role concepts. It was lately extended in the AGRE[**?**] work (E for environment). MOISE^{Inst} model [13] includes structural, functional and deontic views. Its structural view is related with our organization meta-model, detailing roles, groups and links or relationships. In our proposal, the environment is also modelled and the internal topology of groups is considered as well. MOISE^{Inst} functional view describes plans and missions to achieve goals, similarly to our A-Objectives. In our approach, services required and offered are modelled too, and agent interactions are deeply described in the interaction meta-model. Finally, MOISE^{Inst} deontic view describes permissions and obligations of roles, including sanctions. Our normative meta-model also incorporates rewards.

ODML [17] uses a basic underlying model of organizations for performance prediction of the multiagent organization. Their existing organizational models [9] have served as a basis for our topological analysis [11]. AML [10] extends UML with agent concepts, including resources, environment, organizational units and services, but it lacks of a normative modelling. Our proposal has adopted AML environment perspective, using ports for accessing services and resources. Moreover, our meta-models are integrated in an iterative process, such as in INGENIAS or ANEMONA methodologies.

OMNI [18] offers Normative, Organizational and Ontological Dimensions. The mission of the organization, its norms and rules, roles, groups and concrete ontological concepts are detailed. It is also based on contracts, used for acquiring roles and controlling agent interactions. In our proposal, those contract specifications can be used in the dynamic view of the organization meta-model for better defining the organizational services.

Finally, OCMAS [19] defines a meta-model for MAS that allows the system to design its own organization at runtime. It is based on agent capabilities (similar to our agent meta-model, in which tasks and services that an agent is responsible are defined); role assignments (described in our organization meta-model); and policies, which include behavioral and reorganization policies (defined in our normative meta-model) and assignment policies (described in our organization and environment meta-models using access restrictions on resources and services).

Our Organizational MAS modelling approach has been integrated in an iterative process of system development, in which several methodological guidelines are employed for describing the mission of the organization, its productive tasks and processes, its organizational dimensions and topological structure, its decision and information processes, its dynamics and normative behaviour and its reward system. Moreover, a BNF language for describing norms has been developed. It allows defining restrictions on service usage, registration and provision. Furthermore, a graphical development tool is currently being implemented, that helps designers with diagram model construction and automatic code generation.

References

1. Dignum, V., Meyer, J., Wiegand, H., and Dignum, F.: An organization-oriented model for agent societies. In Proc. RASTA-02 (2002)
2. Dignum, V., Dignum, F.: A landscape of agent systems for the real world. Tech. Report Inst. Information and Computer Sciences, Utrecht Univ. (2006)
3. Botti, V., Giret, A.: ANEMONA: A multi-agent methodology for Holonic Manufacturing Systems. Springer-Verlag. In Press, (2008)
4. Gomez, J., Fuentes, R., Pavon, J.: The INGENIAS Methodology and Tools. Agent-oriented Methodologies, Idea Publishing Group, 236–276. (2005)
5. Kelly, S., Lyytinen, K., Rossi, M.: MetaEdit+: A Fully Configurable Multi-User and Multi-Tool CASE Environment. In Proc. CAISE96, LNCS 1080:1–21. (1996)
6. Robbins, S.: Organizational Behavior. Pearson Prentice Hall (2007)
7. Moreno-Luzon, M., Peris, F., Gonzalez, T.: Gestión de la Calidad y Diseño de Organizaciones. Prentice Hall, Pearson Education (2001)
8. Hodge, B.J., Anthony, W., Gales, L.: Organization Theory: A Strategic Approach. Prentice Hall (2002)
9. Horling, B., Lesser, V.: A survey of multiagent organizational paradigms. The Knowledge Engineering Review, 19:281–316. (2004)
10. Cervenka, R., Trencansky, I.: AML. The Agent Modelling Language. Whitestein Series in Soft. Agent Tech. and Autonomic Computing. Birkhäuser Verlag (2007)
11. Argente, E., Palanca, J., Aranda, G., Julian, V., Botti, V., Garcia-Fornes, A., Espinosa, A.: Supporting Agent Organizations. In *Proc. CEEMAS07*, 236–245. (2007)
12. Grossi, D., Dignum, F., Dastani, M., Royakkers, L.: Fundations of organizational structures in multiagent systems. In Proc. AAMAS05, 690–697. (2005)
13. Gateau, B., Boissier, O., Khadraoui, D., Dubois, E.: Moise-inst: An organizational model for specifying rights and duties of autonomous agents. In Proc. CoORG (2005)
14. Weyns, D., Parunak, H., Michel, F., Holvoet, T., Ferber, J.: Environments for multiagent systems. In Proc. E4MAS, Springer-Verlag LNAI 3374: 1–47. (2004)
15. López, F., Luck, M., d'Inverno, M.: A normative framework for agent-based systems. Computational and Mathematical Organization Theory, 12:227–250. (2006)
16. Ferber, J., Gutkenecht, O., Michel, F.: From agents to organizations: an organizational view of multi-agent systems. In Proc. AAMAS03- AOSE. (2003)
17. Horling, B., Lesser, V.: Using ODML to model mulit-agent organizations. In Proc. IEEE/WIN/ACM INt. Conf. on Intelligent Agent Technology, 72–80. (2005)

18. Dignum, V., Vazquez, J., Dignum., F.: OMNI: Introd. social structure, norms and ontologies into agent organizations. In ProMAS, LNCS 3346: 181–198 (2004)
19. DeLoach, S., Oyenan, W., Matson, E.: A capabilities-based model for adpative organizations. Auton. Agent Multi-Agent Syst., 16:13–56 (2008)