Towards a development framework for Regulated Open Multi-agent Systems based on contracts

Emilia Garcia, Adriana Giret, and Vicente Botti

Universidad Politecnica de Valencia,
Camino de Vera S/N, Valencia, Spain
{mgarcia,agiret,vbotti}@dsic.upv.es

Abstract. In our work we deal with the problem of engineering Regulated Open Multiagent Systems. Our method integrates open multi-agent systems and service-oriented architectures. Moreover, e-contracting and regulation enforcement mechanisms are integrated in order to control the behaviour of the agents.

Keywords: Organizational Multiagent Systems, Service-oriented architectures, Contracts, Methodology

1 Introduction

In our work we deal with the problem of engineering Regulated Open Multiagent Systems (ROMAS). They are systems in which heterogeneous and autonomous agents may need to coexist in a complex social and legal framework that can evolve to address the different and often conflicting objectives of the many stakeholders involved.

Our architecture is based on Open Organizational Multiagent Systems [5, 1]. Figure 1 shows an overview of our architecture and its graphical notation. The concept of organization has become a key concept in Multiagent Systems (MAS) research, since its properties can provide significant advantages when developing agent-based software, allowing more complex system designs to be built with a reduced set of simple abstractions.Organizations impose limits on the actions that the agents can perform by means of *Norms* and *Contracts*. However agents maintain their autonomy, so they can choose the actions to do next and select with whom to perform them. In this way, organizations comprise both the integration of organizational and individual perspectives and the dynamic adaptation of models to organizational and environmental changes.

In our proposal, agents interact between them by means of *Services* which represent the functionality that agents offer to other entities. The *BulletinBoard* entity is a non-mandatory element of the architecture. It allows the publication of offers and demands of *Products* and *Services*. Any entity can consult, add or remove offers and demands in a standard way, so the *BulletinBoard* improves the interoperability between internal and external entities of the system.

Moreover, in our approach contracts are used to formalize the legal context of the organization, its commitments and interchanges (Figure 1: ©$_3$). Contracts

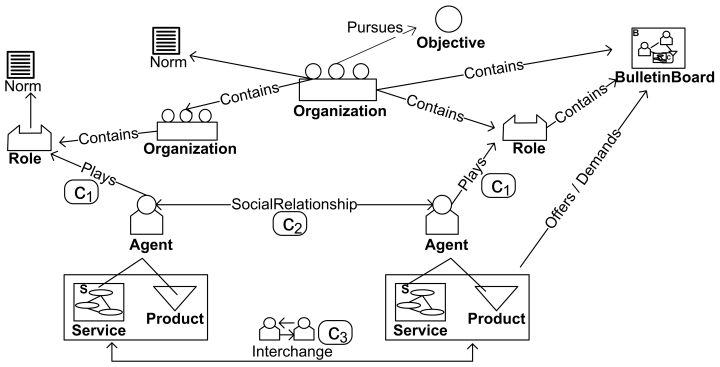

Fig. 1. Summarized ROMAS architecture

also allow to integrate the top-down specification of organizational structures with the autonomy of participating agents [4]. They can describe the rights and responsibilities that an agent for example, acquires when playing a specific role into an organization (Figure 1: ©$_1$). Besides, the structure of the organization can be detailed by means of the formal description of the social relationships between agents (Figure 1: ©$_2$). A detailed description of this architecture can be found in [6].

Developing systems of this kind is a very complex task because it includes the definition of the global behaviour of the system, the individual behaviour of each agent, the legal context of each entity, and the definition of social and contractual interactions. Our main objective is to provide a complete set of methods and tools that guide and help developers from the requirement analysis to the implementation and execution stage. Figure 2 presents an overview of our current and future work. Our first step was the analysis of the open challenges in this topic and the proposal of a new MAS architecture and metamodel that allow the complete definition of ROMAS (See [6]). At the moment, we are working in a methodology that will guide developers during the analysis and design of this kind of systems based on the previous defined metamodel. This methodology is specified using the template proposed by the FIPA Design Process Documentation and Fragmentation Working Group [2]. As future work, we plan to add a model checking module that validate the design of the system and an algorithm that validate the dynamic creation and activation of norms and contracts at runtime. Finally, all these methods and tools will be integrated in a case tool which will automatically translate models into code by means of a Model Driven Architecture technology (MDA).

2 Developing ROMAS

ROMAS methodology is focused on the analysis and design processes for developing organizational multiagent systems where agents interact by means of

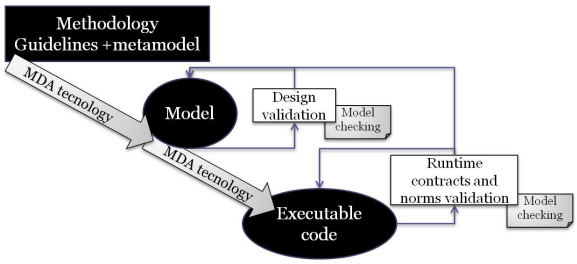

Fig. 2. Work project overview

services, and where social and contractual relationships are formalized using contracts. The analysis and design of a system are formalized by means of several diagrams that are instances of the ROMAS metamodel which is completely described in [6]. In order to facilitate the modeling tasks, this unified metamodel can be instantiated by means of four different views that analyze the model from different perspectives:

- **Organizational External view:** This view allows defining the global goals of the organizations, the functionality that organizations provide and require from their environment and their social structure.
- **Internal view:** This view allows defining the internal functionality, capabilities, believes and objectives of each entity (organizations, agents and roles) by means of different instances of this model.
- **ContractTemplate definition:** This view allows defining *Contract Templates* which are predefined restrictions that all final contract of a specific type must fulfill. Contracts are inherently defined at runtime, but contract templates are defined at design time and can be used at runtime as an initial point for the negotiation of contracts and to verify if the final contract is coherent with the legal context.
- **Interaction/Task view:** This view allows defining both interaction protocols and the sequence of activities in which a task is decomposed.

2.1 Phases of the process

ROMAS methodology is composed of six phases (see Figure 3), which help developers to analyze and design the system from the highest level of abstraction to the definition of individual entities and implementation details. As the figure shows, this is not a linear process but an iterative one, in which the identification of a new element of functionality implies the revision of all the diagrams of the metamodel and the work products produced, so it requires to go back to the appropriate phase. For example, during the third phase (Roles description), part of the detected roles can be played by a group of agents that form another organization. In this case, it is necessary to go back to the first phase of the methodology to analyze the characteristics, global objectives and structure of this organization.

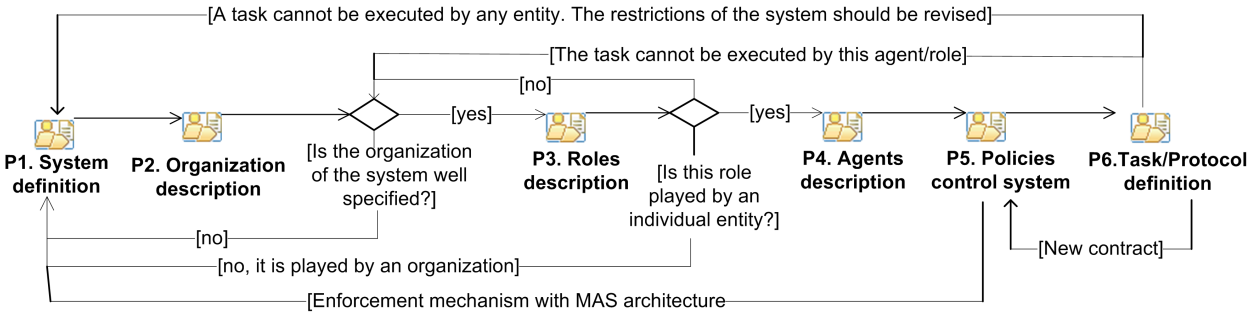

Fig. 3. ROMAS design process

- **Phase 1. System definition:** During this phase the analysis of the system requirements, the identification of use cases, stakeholders and global goals of the system are carried out.
- **Phase 2. Organization description:** During this phase the analysis of the goals of the organization and its structure is carried out. The global goals of the organization are refined into more specific goals, which represent both functional and non-functional requirements that should be accomplished by the organizational units of the system. Then the operational objectives are associated to a task and grouped by their functionality. A *Role* represents part of the functionality of the system, so one *Role* is associated to each group of objectives. When a group of objectives is developed by several roles, there is a relationship between these roles which can be formalized using a contract.
- **Phase 3. Roles description:** This phase describes each identified role by means of an instance of the *Internal view* metamodel.
- **Phase 4. Agents description:** This phase describes each identified agent by means of an instance of the *Internal view* metamodel.
- **Phase 5. Policies control system:** This phase focus on deciding which type of controlling mechanism for norms is going to be applied to each norm and contract. The second task of this phase is to formally specify norms and contracts. Developers can choose its own language to describe norms, although we strongly recommend a formal one like [3]. Each identified contract is formalized by means of an instance of the *ContractTemplate definition*.
- **Phase 6. Task/Protocol definition:** This phase describes each identified task and protocol by means of an instance of the *Interaction/Task model view* for each identified task and protocol.

3 Conclusions

Engineering complex systems composed of heterogeneous and autonomous entities which coexist in a complex social and legal framework is an open research

topic. In our view there are basically five issues to be solved in order to successfully design ROMAS: (1) Defining the social structure and coordination of the system; (2) defining formal regulation mechanisms that formalize the rights and duties of each entity, as well as, mechanism to control the fulfilment of these restrictions; (3) the validation and verification of the models, as well as, the consistency and coherence of norms and contracts.

The presented methodology tries to help developers to develop systems of this kind. The whole development process is guided by the main objectives of the organizations and it also takes into account the individual objectives of the autonomous entities that interact with the system. Moreover, the use of contracts to define the social and contractual relationships between entities allow the system to operate with expectations of the behaviour of others, but providing flexibility in how they fulfil their own obligations.

The proposed guideline allows being integrated into a complete development process, which may include the phases of analysis, design, implementation, validation, installation and maintenance of MAS. This software is part of our current and future work. The next step of our work is to develop an automatic transformation between our meta-model and a formal model checker language. This fact will allow us to validate the designs and to validate the creation and activation of norms and contracts.

References

1. E. Argente, V. Botti, C. Carrascosa, A. Giret, V. Julian, and M. Rebollo. An Abstract Architecture for Virtual Organizations: The THOMAS approach. *Knowledge and Information Systems*, pages 1–35, 2011.
2. M. Cossentino. Design process documentation template. *Technical report, FIPA*, 2010.
3. N. Criado, E. Argente, and V. Botti. A normative model for open agent organizations. In *ICAI*, volume 1, pages 101–107. CSREA Press, 2009.
4. V. Dignum, J. Meyer, F. Dignum, and H. Weigand. Formal Specification of Interaction in Agent Societies. *Formal Approaches to Agent-Based Systems (FAABS)*, 2699, 2003.
5. E. Garcia, E. Argente, and A. Giret. A modeling tool for service-oriented open multiagent systems. In *International Conference on Principles and Practice of Multi-Agent Systems (PRIMA)*, volume 5925 of *LNAI*, pages 345–360. Springer-Verlag, 2009.
6. E. Garcia, A. Giret, and V. Botti. Regulated Open multi-agent Systems based on contracts. In *The 19-th International Conference on Information Systems Development (ISD 2010)*, 2010.