EMFGormas: A CASE tool for developing Service-oriented Open MAS

Emilia Garcia
DSIC, Universidad Politecnica de Valencia
Camino de Vera S/N, Spain
mgarcia@dsic.upv.es

Estefania Argente
DSIC, Universidad Politecnica de Valencia
Camino de Vera S/N, Spain
eargente@dsic.upv.es

Adriana Giret
DSIC, Universidad Politecnica de Valencia
Camino de Vera S/N, Spain
agiret@dsic.upv.es

ABSTRACT

EMFGormas[1] is a new approach for modeling Service-oriented Open Multiagent Systems using the MDA Eclipse Technology. It offers a CASE tool based on an unified meta-model for engineering large-scale open systems in which the constituent entities interact among them by means of services.

Categories and Subject Descriptors

D.3.3 [**Software Engineering**]: Design Tools and Techniques

General Terms

Design

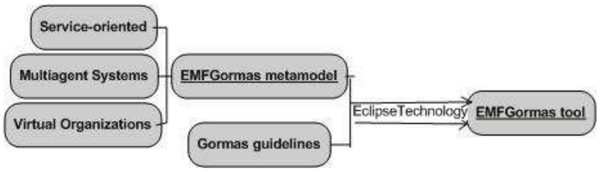
Keywords

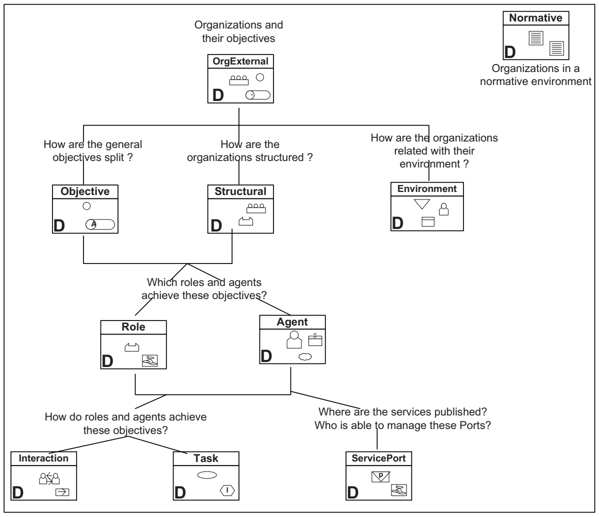
Service oriented architectures, Agent Organizations, Software Engineering, Tool Support

1. INTRODUCTION

Service-oriented Open technology is becoming more and more the enabler tool for today open enterprise information systems.Despite the number of research work in the field, there are still open issues such as: (1) Support for the integration of services and agents and organizations in a transparent way; (2) Dynamicity features for on-the-fly creation, modification and deletion of organizational structures, norms, services, etc; (3) Specific CASE tools for modeling SOMAS.

In our research work, we deal with the problem of engineering Service-oriented open MAS. To this end, we work on models and tools for engineering large-scale open systems in which the constituent entities interact among them by means of services (Figure 1). The solution domain for IS that we propose is built upon Virtual Organizations (VOs), which represent a set of individuals and institutions that need to coordinate resources and services across institutional boundaries [2]. We apply a Model Driven Architecture (MDA) mechanism to define a Service-Oriented Open MAS (SOMAS) meta-model that enhances the Virtual Organization Model [2]. In this work, we present a prototype of an engineering tool for SOMAS that employ Eclipse technology to implement the proposed meta-model and some methodology guidelines defined in [1].





Organizations and their objectives
OrgExternal
D
Normative
D
Organizations in a normative environment
How are the general objectives split ?
How are the organizations structured ?
How are the organizations related with their environment ?
Objective
D
Structural
D
Environment
D
Which roles and agents achieve these objectives?
Role
D
Agent
D
How do roles and agents achieve these objectives?
Where are the services published? Who is able to manage these Ports?
Interaction
D
Task
D
ServicePort
D

Figure 2: Unified meta-model for SOMAS

2. EMFGORMAS MODELING PROCESS

EMFGormas metamodel [3] allows modeling systems at a high level of abstraction. They include the integration of organizational and individual perspectives and also the dynamic adaptation of models to organizational and environmental changes. A complete definition of the metamodel and some case studies can be found in [2]. In order to simplify the modeling task several diagrams can be used. Figure 2 briefly summarizes our proposed modeling process.

Firstly, the analysis of the system requirements is carried

[1] This work is partially supported by the TIN2009-13839-C03-01, PROMETEO 2008/051 projects, CONSOLIDER-INGENIO 2010 under grant CSD2007-00022 and FPU grant AP2007-01276 awarded to Emilia Garcia.

Cite as: EMFGormas: A CASE tool for developing Service-oriented Open MAS , E. Garcia, E. Argente and A. Giret, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1623-1624

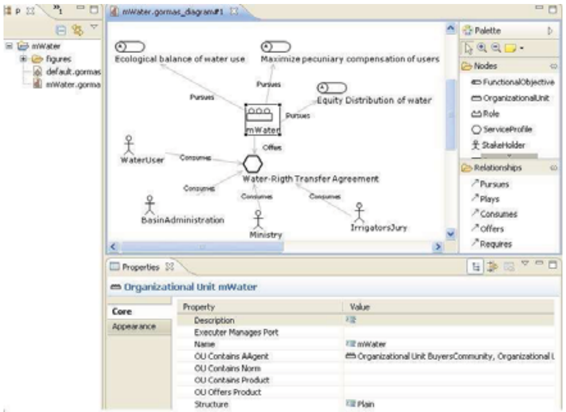
[2] http://www.dsic.upv.es/users/ia/sma/tools/EMFgormas

out, defining the global goals of the organization, the stakeholders and the functionality that the organization provides and requires from these stakeholders (**Organization external view** diagram). Secondly, the analysis of the goals of the organization is carried out in which the global goal of the organization is refined into more goals(**Objectives view** diagram). Next, the components of the organization are defined, i.e. the Organizational Units (OU), which represent groups of members of the organization; the roles defined inside each OU ; their social relationships; the products available by the OUs that can be accessible for their members; and the norms that control the global behavior of the OU members (**Structural view** diagram). Moreover, the products of the environment are analysed, divided into applications (functional interfaces) or resources (with consumable features). The permissions for accessing these elements of the enviroment are also defined (**Environment View** diagram). Thirdly, the internal functionality of the OUs is defined, by means of the Role and Agent Views. In the **Role View**, the roles are related with their responsibilities (tasks), capabilities (services) and objectives. The **Agent View** describes the concrete responsibilities of agents (tasks), the roles that they can play inside an OU, the services that offer to other entities, their mental states (believes, events and facts) and the goals that they pursue. The way in which the roles and agents achieve their goals is defined by means of the Interaction and Task Views. In the **Interaction View**, the participants of the interaction are identifed, as well as the sequence of activities (task and services) and performatives that are employed through the interaction. In the **Task View**, the specific functionality of the services and tasks is detailed, more specifically, the service description (*Service-Profile*); its specific implementation, by means of service or task composition; as well as the sequence of tasks that is needed. Finally, the **Service Port View** defines the way in which the services must be published so as to be discovered by any agent. Thus, the service publication points (service port) are identified, as well as the entities that control each port and give permissions for registring or accessing services. Throught the whole process, whenever a restriction on the behavior of the system entities is identified, it should be described in the **Normative View** diagram.

3. EMFGORMAS TOOL

This tool has been developed following the MDA standards by means of the Eclipse technology. EmfGormas tool consists of several Eclipse plugins that provides graphical capabilities to design Service-oriented Open MAS using the modeling language and process summarized in Section 2. The tool allows for editing multiple hierarchically structured views. The tool offers several graphical editors, one for each view of the model, that restricts the modeling task to the elements and relationships defined in the meta-model. The tool simplifies the interdependencies between diagrams and checks their completeness and coherence. Therefore, all the diagrams of the same model are connected and designers can navigate from one view to another clicking into the main entity of the diagram.

The tool generates an ecore description of the whole modeled system. Ecore is a standard format based on XML and it can be used to interoperate with any other Eclipse plugins and with other tools. Moreover, the model that represent a system can be seen as a source model to automatically generate code for any agent platform. For example, using the Mofscript plugin is possible to translate between ecore model to any target code in a semi-automatically way.



Organizational Unit mWater

Property	Value
Description	
Executer Manages Port	
Name	mWater
OU Contains AAgent	Organizational Unit BuyersCommunity, Organizational L
OU Contains Norm	
OU Contains Product	
OU Offers Product	
Structure	Plain

Figure 3: mWater Organization External diagram

Figure 3 shows a snapshot of the EMFGormas tool in which the organization external view of a case study is developed. This figures shows that the EMFGormas tool offers a traditional interface of CASE tools. This fact reduces the learning time and makes its usage more intuitive.

4. CONCLUSIONS

We have proposed a unified meta-model for engineering large-scale open systems in which the constituent entities interact among them by means of services. It supports dynamic features for creation, modification and deletion of organizational structures, norms, services, etc. A CASE tool based on the proposed meta-model has been developed. It simplifies the modeling task and check the interdependencies between the models and their coherence with the meta-model. The use of the ecore standard improves the interoperability with other tools and other Eclipse plugins. In the near future, we plan to improve the verification module of the CASE tool and add a new module to automatically generate code from the ecore model to a service-oriented multiagent platform called Thomas [4].

5. REFERENCES

[1] E. Argente, V. Botti, and V. Julian. Gormas: An organizational-oriented methodological guideline for open mas. In *AOSE*, pages 85–96, 2009.

[2] N. Criado, E. Argente, V. Julian, and V. Botti. Designing Virtual Organizations. In *PAAMS'09*, volume 55, pages 440–449, 2009.

[3] E. Garcia, E. Argente, and A. Giret. A modeling tool for service-oriented open multiagent systems. In *The 12th International Conference on Principles of Practice in Multi-Agent Systems. PRIMA 2009*, volume 5925 of *LNAI*, pages 345–360. Springer-Verlag, 2009.

[4] V. Julian, M. Rebollo, E. Argente, V. Botti, C. Carrascosa, and A. Giret. *Using THOMAS for Service Oriented Open MAS*, pages 56–70. Springer, 2009.