# Open MAS Architecture. Providing Real Time Solutions

Martí Navarro, Sara Rodríguez, Vicente Julián, and Vivian F. López

**Abstract.** This presents a study in which a high level abstract architecture was used to design open multi-agent systems and virtual organizations that offer services with temporal constraints implemented by Real-Time Agents. The results will demonstrate how the proposed architecture, with features that make it suitable for development of open MAS (Multi-Agent Systems), allows us to add specific functionality and real time services.

**Keywords:** Open MAS , Real-Time Agents, Virtual Organizations.

## 1 Introduction

One of the goals of multi-agent systems is to construct systems capable of autonomous and flexible decision-making, and of cooperating with other systems within a "society". This "society" should take certain characteristics into account, such as distribution, constant evolution, or a flexibility that allows its members (agents) to enter and exit at will, a correct organizational structure, and the ability for the system to be deployed on different types of devices. Each of these characteristics can be achieved through the open multi-agent system and virtual organization paradigm. This paradigm was conceived as a solution to the management, coordination and control of the agents' behavior. Organizations should not only be able to describe the structural composition (i.e. functions, agent groups, interactive and relationship patterns between roles) and the functional behavior (i.e. agent tasks, plans or services), but should also be able to describe the behavioral norms

Sara Rodríguez · Vivian F. López
Departamento Informática y Automática Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, Spain
e-mail: vivian@usal.es, srg@usal.es

Martí Navarro · Vicente Julián
Departamento Sistemas Informáticos y Computación. Universidad Politécnica de Valencia
46022, Valencia, Spain
e-mail: mnavarro@dsic.upv.es, vinglada@dsic.upv.es

Y. Demazeau et al. (Eds.): Trends in PAAMS, AISC 71, pp. 69–76.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

for the agents, the entry and exit of dynamic components, and the formation, which is also dynamic, for the groups of agents. In general, it is necessary to define the standards and platforms required for the interoperability of the agents that meet these requirements. This article attempts to present a study in which a high level abstract architecture was applied with the specific intent of addressing the design of open multi-agent systems and virtual organizations using services with temporal constraints offer by Real-Time Agent. It will be shown how the proposed architecture, with its features that make it suitable for development of open MAS (Multi-Agent Systems), make it possible to add specific functionality and services for real time.

It has been necessary on many occasions to include specially designed agents within a society so that they can carry out particular services involving certain temporal restrictions when they are executed. These special agents, referred to as real time agents [5], are specifically designed to work in environments whose systems are characterized by strong temporal restrictions and require a specific infrastructure to function correctly and ensure that the restrictions are satisfied. It is further necessary to consider that in this type of environment, correcting the system depends not only on the logical computed result, but also on the moment at which the result is produced [10].

The article is structured as follows: section 2 describes an abstract architecture specifically adapted to work with open multi-agent systems and virtual organizations. Section 3 explains the principal characteristics of a real time system. Section 4 shows how the open architecture has been extended to support agents in real time; and finally some conclusions are given in section 5.

## 2 Ovamah

The primary concepts used in this study include the OVAMAH platform than can use agent technology in the development process and apply decomposition, abstraction and organization techniques; and the main features of temporal bounded services that are provided by Real-Time agents with the capability of guaranteeing the service execution time.

The proposed methodology used in this study uses the OVAMAH platform. OVAMAH is based on THOMAS (MeTHods, Techniques and Tools for Open Multi-Agent Systems) architecture [1] [3]. THOMAS is essentially formed by a set of services that are modularly structured. It uses the FIPA architecture, expanding its capabilities with respect to the design of the organization, while also expanding the services capacity. The information and mechanisms used by the traditional FIPA Directory Facilitator (DF) are insufficient to deal with open systems and system dynamics, having several limitations such as their very basic service descriptions (name, type, protocol, ontology, language, ownership and properties) and permissible functionalities (register, deregister, modify and search), which do not at any point take into consideration the virtual organizations. Another DF limitation concerns the service discovery, whose search algorithm does not use any semantic information and it does not consider service compositions. Given that Service Oriented Architectures have dealt with these

aspects issues in depth, THOMAS proposes to incorporate this SOA vision into the way services are han-dled in an agent platform. Essentially, THOMAS updates the FIPA approach by presenting all the architecture functionalities specified as SOA services and updat-ing the Directory Facilitator (DF), that is, the FIPA module in charge of managing services, in order to use services specified according to SOA standards. This mod-ule, now called the Service Facilitator (SF), can not only register and offer affor-dability and discovery of services, but can also manage the service requests. Moreover, these services can be related not only to agents, but also to organiza-tional units. THOMAS also includes a new module called Organization Manage-ment Service (OMS) which is responsible for managing the lifecycle of virtual or-ganizations (in the same way that AMS deals with the lifecycle of agents) and for managing norms. More specifically, it controls how Virtual Organizations (or Or-ganizational Units) are created and the way entities participate inside such organi-zations (which entities participate, how they are related to each other and which roles they play). Another important module in THOMAS is the PK (Platform Ker-nel), which controls not only the communication but also the life cycle of internal agents. It is a classical standard FIPA abstract architecture PK module. All these modules can be seen, imbedded into the case study, in Figure 1.

## 3 Real-Time Agents and Temporal-Bounded Services

Sometimes, the concept of real-time generates confusion. In some situations, real-time applications are interpreted as on-line applications, but in the context of the present study, real-time applications refers to an action that answers to an ex-ternal event in a timely and predictable manner.

There can exist various tasks within an organization that require some sort of temporal control in their execution. Because of their particular characteristics, these tasks cannot be performed by agents that do not possess the appropriate in-frastructure. In order to ensure the correct execution, as per the temporal restric-tions associated with these tasks, the tasks must be deployed on a real time operating system (RTOS).

The Real-Time agent model incorporated in the OVAMAH architecture is an agent that can offer Temporal Bounded Services and execute the different tasks associated with these services. The tasks are executed according to the temporal restrictions that define the maximum allowable time for the service to provide a result, and the temporal restrictions associated with each of the tasks that comprise the service. A Temporal Bounded Service (TBS) is considered a service that imposes a limit on the allowable response time. Therefore, apart from the usual information related to Inputs, Outputs, Preconditions and Effects, the service description should contain an explicit representation of time as non-functional parameters. With the use of a time parameter, it is possible to indicate how much time the service will need to complete its execution in the worst case scenario. In order to offer this information to the clients, an OWL-S service description, extended with non-functional parameter, has been used. The value of the service execution time is represented by an average execution time, which is obtained by

measuring the cost of the service off-line. Additionally, preconditions and effects could contain modal expressions that make it possible to express relationships between different services and states. These relationships are expressed using an extension of the RuleML for modal logics [2].

In order for a real time agent to offer TBS, it must possess certain qualities such as:

(i) The ability to determine whether a TBS can be executed before a maximum time. The client will establish the start time for executing the TBS, the maxi-mum time to complete the TBS and, if the execution of the TBS must be peri-odic, the frequency of execution. These parameters will determine the temporal constraints of the TBS. The real time agent should include a planning compo-nent that can perform a viable analysis of the request. There are TBSs that, be-cause of their characteristics cannot be analyzed by the planning agent. These must then be implemented by the real time agent, which would otherwise not offer that particular TBS. There are certain schedules, for example, that can only manage periodical tasks. Thus a TBS that must be executed aperiodically cannot be analyzed by the real time agent.

(ii) The ability to execute TBSs and check that they are provided on time in order to fulfil the client's request.

As an additional option, the RTA can possess the ability to establish a relationship with clients that request a TBS through an agreement protocol, and negotiate the maximum allowable time to execute the TBS in the event that the RTA cannot execute the TBS within the time indicated by the client.

On occasion, the TBS service request cannot be met within the deadline. In this case, the service can offer the possibility of negotiating a new deadline with the client, based on the conditions analyzed by the RTA provider. To ensure that the negotiation is carried out, it is necessary for both the client and the provider to implement the agreement protocol shown in [8].

The following section explains how the OVAMAH architecture was expanded to include real time agents.

## 4 Extending the Architecture with Real-Time Agents

This section will show how the OMS and SF platform modules afford the possibility of managing open systems that permit the integration of real time agents and offer TBS. As a whole, these modules can be seen, as a platform-independent set of services in a framework that manages virtual organizations for open systems. This is known as the OVAMAH Framework. The OVAMAH framework provides a virtual organization in which any entity is automatically included, and a general role that allows the entity to ask for service descriptions so that it can fullfil its needs. Using a service description, the client can be informed about the roles required to fulfill the given request. The client can also provide a specific service within the organization. OVAMAH makes it is possible to incorporate agents that can openly provide the system with new services. This ability can be used to add agents to the platform, particularly those that can execute tasks and/or

services that, because of their characteristics, must be executed according to certain temporal restrictions. The use of these agents in a real time environment is thus quite beneficial. OVAMAH provides a valid logic for modeling a real time MAS specification.

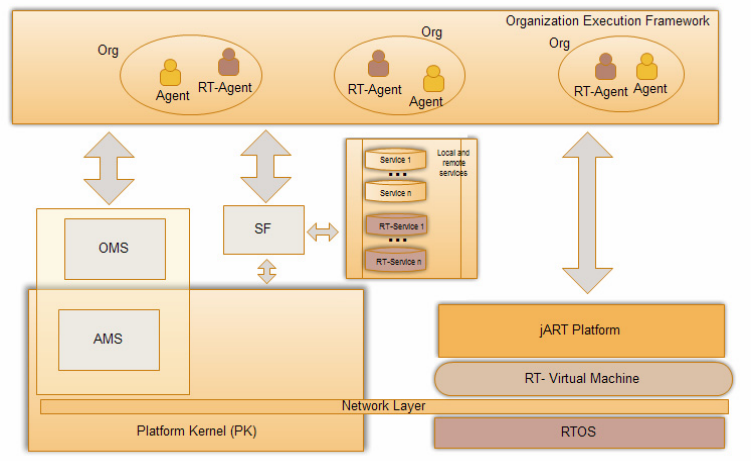


**Fig. 1** OVAMAH Architecture for Time-bounded Services

One of the first factors to bear in mind is the need for a specific software to execute agents in real time. As mentioned in the previous section, in order to ensure that temporarily restricted tasks are carried out in time, they must be executed on a RTOS. To properly manage and control real time agents and their corresponding tasks, they must be executed on a platform that offers real time planning, temporally bounded communication between agents, and complete control over the processes executed in the system. One example of a system that offers these capabilities is the jART [9] real-time MAS platform, which can be easily integrated within the OVAMAH architecture s it follows the FIPA standard that was used to design the components of the OVAMAH architecture. The OMS in OVAMAH offers the services that are needed to ensure the correct functioning of an organization. These services can be dynamic or structural. The dynamic services can manage the dynamic entrance and exit of the agents in the system. The structural services are what make it possible to register or unregister roles, norms and units within the organization. The roles will have associated rules of interaction and behavior that allow us to define which type of services can be offered and required by an agent that is performing a specific role. The rules allow us to define restrictions, obligations and permission between agents at the moment they execute, request or implement services. The type of services available is controlled by the system through these norms. In our case, the norms require the TBS providers to be RTA, that these agents have a planner capable of analyzing the TBS, and furthermore, that they be able to control the execution of tasks associated with the TBS.

The system will specify the profile of the service providers by defining roles as well as the norms for processing service requests or the available results. This way, in the event of any illicit or improper behavior on behalf of the provider, the system will respond with sanctions. In this case study, the system will have a type of super-role RTA, which will be an "external" or accessible role that specializes in different types of sub-roles according to the type of planner used by the RTA. The following sub-roles will be available:

-RTA_Pd: The RTA agent role that can perform only periodical tasks, as it imple-ments a viability analyzer that can only analyze this kind of task. For example: monotonic rate algorithm [7] or earliest deadline first [7].

-RTA_Apd: The RTA agent role that can perform aperiodical roles using planners that can analyze them. For example: the slack stealing algorithm [6]. The planners capable of analyzing aperiodical tasks can also be used to analyze periodical tasks.

At the software level, OMS establishes a hierarchy of roles so that an agent with a role can offer services associated with roles hierarchically superior to its own, so long as this action is not contrary to the established norms. In this case, for example, an agent with the role of RTA_Pd can directly request services as-signed to the RTA role, but inversely the RTA agent must first request permission from the OMS to acquire the RTA_Pd role to use the services assigned to that role.

If we move to a hardware level, a unit that offers or includes TBS services, it should contain a real time agent that includes the following intrinsic characteristics: (i) It should be executed on a RTOS. (ii) It should have an appropriate planner available to carry out a viability analysis and plan tasks associated with the TBS:

The SF announces services that are required for the system to function properly. These services only have a Profile assigned to them, a structure for entrances/exits, preconditions/postconditions that it must comply with. The external agents can request the existing list of services and determine whether entering or not entering forms part of the organization, and under which roles. In our case, a RTA can request carrying out a TBS and it will be the platform responsible for confirming if the agent is capable of carrying out the specified type of service. If an that agent lacks the appropriate RTA characteristics attempts to provide a similar type of service, the platform will let it know the request has been rejected.

The type of services that can be offered is controlled by the system norms. All of the posible behaviors within the system will be ccontrolled by the norms defined with the following syntax:

<norm>::=<deontic_concept>(<action>[<temporal_situation>][**IF** if_condition>])
[**SANCTION**(<state>)] [**REWARD**(<state>)] donde <deontic_concept>::= **OBLIGED** | **FORBIDDEN** | **PERMITTED**

and the action will be dialogical (send a message) or involving a request, provision or service registration (REQUEST | SERVE | REGISTER <service>). In <if_condition> it will be posible to indicate the results of the functions or services. The norms will reflect the necessary restrictions for real time tasks, including:

-The role that a RTA can perform is associated with the type of planning that it has implemented. An agent with a planning capacity limited to analyzing periodical tasks cannot perform the RTA_Apd role. However, an agent with a planning capacity that can analyze aperiodical tasks can, in fact, perform the RTA-Pd role.

```
FORBIDDEN RTA_Pd REQUEST AdquireRole(RTA_APd,GlobalUnit)

PERMITTED RTA_APd REQUEST AdquireRole(RTA_Pd,GlobalUnit)
```

-The RTA_Pd can only execute periodical services.

```
OBLIGED PK SERVE Suspend(RTA_Pd) IF RTA_pd REQUEST AdquireRole(RTA_APd,GlobalUnit)
```

-An agent that commits to carrying out a service is required to do so within a determined period of time, otherwise it will be sancioned within the organization.

```
OBLIGED PK SERVE Suspend(RTA) IF RTA_Serv_Profile (ServiceResult(requestTime ?time)) > "spentTime"
```

-An agent can delegate a TBS so long as the agent to which it assigns the service is an RTA that meets the necessary specifications.

```
PERMITTED RTA REQUEST TBS(TBSProfile) AFTER RTA REQUEST TBS1(TBS1Profile) AND Quantity(ServiceResult(TBS1,RTA,?Client, "completed")) > "1"
```

-If an agent requests a service from a RTA, the service cannot be carried out if either the allotted time has expired, and/or there are no available resources.

```
OBLIGED PK SERVE Suspend(RTA) IF RTA_Serv_Profile (input(requestTime ?time))>"currentTime"
```

All of the information related to existing units, their roles, service profiles and norms will be defined in the OWL-S file using an ontology and service organization. Upon initializing the system, the OMS will read the structural information from the file and create and initiate the organization. The OMS will internally save the list of norms that define the affected role, the norm´s content and the roles responsible for controlling if the norm is satisfied.

Finally, the SF will save the profiles of the services that are offered and/or needed by the system, as well as the relationship between the service provider entities and their corresponding *Process and Grounding* service During the grounding process it is possible to specify the specific parts of the real time systems that were not detailed by the norms, such as the type of planner that is used, the RTOS and/or multi-agent platform to use, as well as the entrance parameters that the service user must bear in mind when submitting a request (such as the moment to initiate the service, the frequency, and, when available, the corresponding priority and deadline.

## 5 Conclusions

The social factors that make it possible to arrive at a solution within multi-agent system organizations are also becoming increasingly important for structur-ing interactions within open and dynamic worlds. Any infrastructure that can sup-port the execution of multi-agent applications within these environments must be robust, efficient and adaptive over time. Given the characteristics of these open environments, particularly their dynamism, it is essential to find a new approach to

support the evolution of these systems and to facilitate their growth and update in execution time. OVAMAH is a platform for developing open MAS. It can integrate external systems that offer services that must be completed by a particular deadline, as specified by the requesting client. These services should be analyzed to ensure that they are carried out appropriately. The RTA integrated within OVAMAH can offer the necessary mechanisms for performing a viability analysis and ensuring a correct execution, complying with the temporal restriction. OVAMAH represents the first step in obtaining true deployed virtual organizations. This study presents the next step in the evolution of OVAMAH to support SMA in execution time.

**Acknowledgements.** This research has been partially supported by the project TIN 2009-13839-C03.

## References

1. Argente, E., Botti, V., Carrascosa, C., Giret, A., Julian, V., Rebollo, M.: An Abstract Architecture for Virtual Organizations: The THOMAS project DSIC-II/13/08 (2008)
2. Boley, H.: The rule-ml family of web rule languages. In: Alferes, J.J., Bailey, J., May, W., Schwertel, U. (eds.) PPSWR 2006. LNCS, vol. 4187, pp. 1–17. Springer, Heidelberg (2006)
3. Carrascosa, C., Giret, A., Julian, V., Rebollo, M., Argente, E., Botti, V.: Service Oriented MAS: An open architecture. In: Proc. of AAMAS 2009, pp. 1291–1292 (2009)
4. Dignum, V., Dignum, F.: A landscape of agent systems for the real world. Technical report 44-cs-2006- 061, Institute of Information and Computing Sciences, Utrecht University (2006)
5. Julián, V., Botti, V.: Developing real-time multi-agent systems. Integrated Computer-Aided Engineering 11, 135–149 (2004)
6. Lehoczky, J.P., Sacha, L., Ding, Y.: An optimal algorithm for scheduling soft-aperiodic tasks in fixed-priority preemptive systems. In: Proceedings of the IEEE Real-Time Systems Symposium, pp. 110–123 (1992)
7. Liu, C., Layland, J.W.: Scheduling algorithms for multiprogramming in a hard real-time environment. Journal of ACM 20(1), 46–61 (1973)
8. Navarro, M., Del Val, E., Rebollo, M., Julián, V.: Agent Negotiation Protocols in Time-Bounded Service Composition. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 527–534. Springer, Heidelberg (2009)
9. Navarro, M., Julián, V., Soler, J., Botti, V.: jART: A real-time multi-agent platform with rt-java. In: Proc. of IWPAAMS 2004, pp. 73–82 (2004)
10. Stankovic, J.A.: Misconceptions about real-time computing. IEEE Computer 12(10), 1–10 (1988)