A generic conceptual framework for self-managed environments

E. Lavinal, T. Desprats, and Y. Raynaud

IRIT, UMR 5505 - Paul Sabatier University
118 route de Narbonne, F-31062 Toulouse cedex 9
{lavinal, desprats, raynaud}@irit.fr

Abstract. The high complexity of existing managed environments has led to the need to increase the autonomy of network and service management solutions. This short article presents a generic multi-agent organizational framework dedicated to the management domain. This conceptual framework constitutes a first step towards the design of self-managed environments.

1 Introduction

The self-management concept is a vision of network and service management (NSM) which results from the evolution of management solutions facing the increasing complexity of the managed environment [1]. This vision of management consists in providing managed resources with a high degree of autonomy, allowing them to manage themselves according to high level objectives and to situations they observe in their environment. Due to its inherent autonomy property, the multi-agent system (MAS) paradigm seems a propitious candidate to support this self-management concept. Generally speaking, a MAS can be described as a large number of autonomous entities, called agents, in interaction with each other within a common environment.

In this short article, we briefly present an organizational multi-agent framework dedicated to NSM as a first step towards the conception of self-managed services and network resources. This conceptual framework consists of groups of management agents coupled with managed resources and endowed with specific management skills. Our objective is to keep this framework as generic as possible in order to be independent of the management functional domain (configuration, protection, etc.) as well as the target managed environment (physical or logical resources, atomic or compound, etc.). Moreover, we define a dedicated model of the managed environment which captures different types of relationships existing between managed elements. Within the multi-agent framework, this model is shared by groups of management agents that can thus take management decisions according to their own managed resource as well as the rest of the managed environment.

2 A generic multi-agent management framework

Based on existing works emanating from the domain of multi-agent organizations [2], this section successively presents the different concepts of a multi-agent framework dedicated to self-management [3]. This specialization process of the MAS paradigm for the NSM domain has been achieved by adopting an organizational centered multi-agent design methodology (meaning that the stress is put on the relationships the agents have one with the other in a specific organization).

- Managed Element (ME). A ME corresponds to any real world resource that has to be managed. It can consist of either a hardware resource (e.g. modem), or software resource (e.g. service), or even an abstract resource (e.g. an administrative domain). Note that a ME may be either an atomic or compound resource, and that it may eventually depend on one or more other ME(s).
- Managed environment (MEnv). A set of one or more MEs – eventually interrelated – which constitutes an operational system that has to be self-managed.
- Manager Agent (MA). It is a specialization of the agent concept for the management domain. A certain number of management skills which act on one or many ME(s) are embedded into the MA. These management skills constitute the management functionalities an agent is able of accomplishing. Diagnosing faults or configuration actions are examples of management skills. Naturally, a MA also incorporates aptitudes inherent to the agent concept such as viability preservation, reactivity to environmental changes, or inter-agent interaction. We thus give the following definition: *a MA is an autonomous agent, part of a multi-agent system dedicated to self-management, endowed with specific management skills and coupled with one or many MEs.*
- MA-ME coupling. The coupling relationship allows the overall MAS to integrate itself to the existing managed environment. Individually, a MA-ME coupling relationship means that the MA can at least perceive the ME (i.e. it has read access to the resource) and eventually act on it (i.e. it has write access to the resource). In concrete terms, examples of this coupling relationship can consist of local system calls, web-based or SNMP messages. This relationship also implies that the MA should have in its knowledge base a logical representation of its coupled ME(s) (as a classical manager would have of the resource it is responsible for using a management information model).
- Generic Management Functionality (GMF). It can be one of the following: monitoring, configuration, healing, optimization and protection. These functionalities are generic enough to include most of the existing management functionalities. Other management functionalities, more explicit, can inherit from these GMF. For example, an access control functionality can be considered as an extension of the protection functionality. Notice also that these management

functionalities can be based on legacy code (e.g. wrapping an existing intrusion detection engine as a management skill).

– Role of a MA (MRole). Although many organization-centered multi-agent design methodologies have their own specificities on how to represent the role concept, they all agree on its general definition: a description of an abstract behavior of agents. In the management context, a role played by an agent is strongly related to a functional dimension of the system to design (what to do), but also to an environmental one (on whom). Both of these generic aspects should be included in the expression of a MA's role. A GMF is used to represent the functional part of the role while the coupling relationship MA-ME is used to represent the environmental component. A MA can thus have numerous roles for a same ME if it implements several management functionalities (e.g. a MA can simultaneously play a *configure firewall* role and an *optimize firewall* role).
– Group of MAs (GrpMA). Generally speaking, a group represents a set of agents that share some common characteristic. According to the approach we follow, MAs belonging to the same group should obey to two conditions: first, they should share a common representation of the managed environment (MEnv) and second, at least one of their coupled MEs should be included in this representation of the MEnv. Notice that although a MA has a representation of the overall MEnv of the group it belongs to, it can only perceive and act locally on the MEnv via its coupled ME(s).

  This group definition allows a MA to belong to several groups if it has a representation of the MEnv for each group it belongs to (with eventually different representation languages for each group) and if it is coupled with a ME included in each of these MEnv.

The concepts we have just presented are illustrated in figure 1 through a UML class diagram. They constitute the core of an organizational multi-agent architecture dedicated to autonomous NSM solutions.

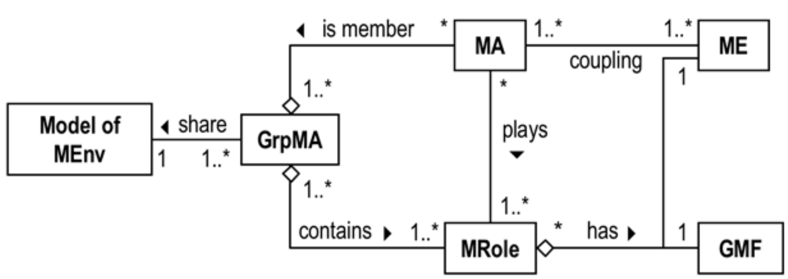

Fig. 1. Generic management multi-agent framework

At runtime, it should be essential for each MA to adapt its local and social behavior according both, to its coupled ME and to the relationships its ME has with the rest the MEnv. Within a group of MAs, to achieve this self-adaptation process, the existence of shared model representing the MEnv is primordial.

3 A dedicated model of the managed environment

One of the main factors of complexity of the managed environment resides in the interrelated nature of all the resources to be managed. Although a MA has a partial perception of the complete managed environment (i.e. its coupled MEs), a local management action on a ME can indirectly affect a large part of the rest of the managed environment. Therefore, in order for a MA to take locally the most appropriate management decision, the MA needs to be aware – as much as possible – of the nature of the existing relationships its coupled ME has with the rest of the MEnv. To capture these relationships, we have defined a dedicated model based on the definition of different types of MEs as well as different types of relationships between these MEs. The objective of this model is not to describe in details the characteristics of the MEs such as a classical management information model would do but rather to express the different types of relationships the resources have with each other within the managed environment. According to the nature of these relationships, the MAs will automatically adapt their local and social behavior, thus participating in the overall autonomy of the management multi-agent framework.

3.1 Expression of the managed elements

Currently we have restricted the expression of the MEs to three categories commonly found in the NSM domain. We will briefly describe each one of them.

- Service. This type of managed element is used to model any application service independently of its function (e.g. an email service or any business service).
- EndSystem. This type of managed element is used to model any kind of end system (e.g. a computer system) which hosts services.
- MidSystem. This type of managed element is used to represent any entity directly manipulating data flows such as in the Laborde's et al. specification language [4]. Three subtypes of this MidSystem entity can be defined depending on the management functionality it accomplishes: the channel functionality (Channel) which propagates data flows to all the entities connected to it (e.g. a LAN or network link), the filter functionality (Filter) which blocks or forwards data flows (e.g. a firewall), and the transform functionality (Transform) which modifies a data flow (e.g. a cryptosystem or a NAT proxy).

3.2 Relationships between managed elements

In order to connect the MEs among each other, we have defined three different types of relationships which constitute different operational dependencies. The identified relationships are the following:

- Service Dependency (SD). It is used to model a dependency between two application services, meaning that a service is dependent on another service to be fully operational (e.g. a web service using a directory service).

– Hosted Dependency (HD). It is applied to model the fact that a service is hosted on an end system (e.g. a directory service hosted on a server).
– Flow Dependency (FD). It represents a dependency at a network level between *EndSystems* and *MidSystems*. These MEs are connected to each other with this relationship in order to obtain a logical network topology.

The interest of these relationships is to allow MAs to deduce local and social behaviors according to their underlying semantic. Therefore, MAs will be able to achieve management actions according to the state of MEs with which they are not necessarily coupled with, but with which they may be related to.

4 Instance example of the conceptual framework

We illustrate here an instance example of the multi-agent conceptual framework we have presented previously. We consider a target managed environment composed of two networks separated by a firewall. Two services (one depending on the other) are hosted on different end-systems connected to each of these networks. This environment is specified by the model *M1* in figure 2.

Based on this managed environment, a group of MAs sharing this model can be created. This group is composed of different MAs which are coupled with one or more ME(s) included in the model M1. Each MA implements a specific GMF which, when associated with the coupled ME, assigns a specific role to the MA. For example, in figure 2, MA1 has a “protect end-system es1” role and MA4 has a “configure firewall fw1” role.

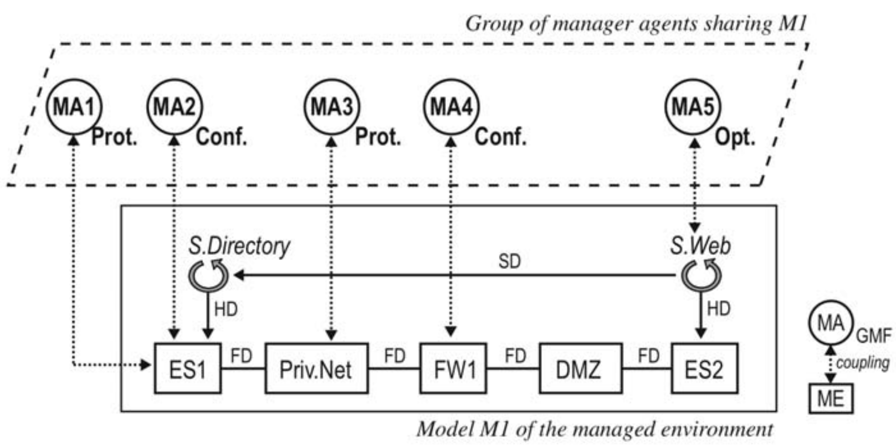

Fig. 2. Instance example of the multi-agent management framework

Currently, we are specifying several rules to analyze the managed environment model based on the different types of relationships the MEs have one with the other. The objective of these rules is to allow the MAs included in the same group to first, automatically adapt their local management behavior on their coupled ME while considering the rest of the managed environment and second, to deduce interactions with other MAs in order to accomplish management actions they could not achieve on their own. For example, in figure 2, by analyzing the managed environment M1, MA1 could deduce that protecting *es1* could result in adding an access control rule within *fw1* (i.e. MA1 interacting with MA3 in order to dynamically add a filtering rule on *fw1* which would allow MA1 to meet its management goal).

5 Conclusion

The approach we presented in this paper promotes the usage of the MAS paradigm to support autonomous management solutions. This approach results in the design of a generic conceptual framework which specializes existing multi-agent concepts for the management domain. Instances of this framework consist of multi-agent groups composed of manager agents playing management roles and interacting with each other to participate in the accomplishment of the global management objective on the underlying resources.

Current work address the automatic deduction of local and social behaviors between MAs based on the analysis of the shared managed environment. This dynamic adaptation will contribute to the self-organization of the MAs and therefore to the self-management of the overall environment.

References

1. J.O. Kephart and D.M. Chess. The Vision of Autonomic Computing. *IEEE Computer*, 36(1):41–50, January 2003.
2. J. Ferber, O. Gutknecht, and F. Michel. From agents to organizations: an organizational view of multi-agent systems. In *International Workshop on Agent-Oriented Software Engineering (AOSE'03)*, Melbourne, Australia, July 2003.
3. E. Lavinal, T. Desprats, and Y. Raynaud. An Organizational-driven Specialization of the Multi-Agent System Paradigm for Self-Management. In *IFIP/IEEE International Symposium on Integrated Network Management (IM'05)*, poster proceedings, Nice, France, May 2005.
4. R. Laborde, B. Nasser, F. Grasset, F. Barr`ere, and A. Benzekri. A Formal Approach for the Evaluation of Network Security Mechanisms Based on RBAC Policies. *Electronic Notes in Theoretical Computer Science*, 121:117–142, February 2005.