Roles and hierarchy in multi-agent organizations

Emmanuel ADAM and René MANDIAU

LAMIH UMR CNRS 8530, University of Valenciennes
Le Mont Houy - FRANCE – 59313 Valenciennes Cedex 9
{emmanuel.adam, rene.mandiau}@univ-valenciennes.fr

Abstract. Holonic Multi-Agent organisations are particular pyramidal organisations where agents of a layer (having the same coordinator) are able to communicate and to negotiate directly between them. Holonic agents are generally structured by services (for examples: to search information, to interact with the user). Some holonic agents, distributed in the system, can have same roles without being in a same layer. However, it is essential to interact with agents according to theirs roles: most of multi-agent methods and platforms are based on the roles management. In order to beneficiate of the two advantages which are the control allowed by the holonic architecture and the roles management, we propose a particular holonic architecture based on the workings mechanisms of human organisations. We have implemented this architecture by offering a role management capacity to a hierarchical multi-agent platform.

1 Introduction

Our researches aim at integrating multiagent organization into human organization, which are more and more based on data exchanges, in order to help actor these system to manage data, to communicate and cooperate between them. Admittedly, tools for aid in co-operative work have already been suggested, some with success, but they do not tackle the overall organization.

So, in order to take into account the human organization and the human factors, such as the human cooperation or even the human-machine co-operation, we have previously developed a method (AMOMCASYS, meaning the Adaptable Modelling Method for Complex Administrative Systems) that has helped us to design and to set-up some information multi-agent systems (IMAS). We have designed the different IMAS following a holonic architecture, and the resulting architecture is quite similar to architectures generally used to build the IMAS. Of course our architecture is not limited to IMAS design and is able to be used in other context (we have started some application in the area of house automation, intelligent vehicles).

In order to design an IMAS, we use both notion of role and hierarchy, and we propose the use of role managers, acting like Human Resources Agent in the human organisation (we call them HRA for Holonic Resource Agents). We have developed some functionalities on a platform dedicated to hierarchical multi-agent systems, that allows us to design and deploy our MAS by implementing the roles only and by describing the positions of the agents in a XML file.

This paper describes firstly the underlying concept of our works, which is the holonic concept; and then some details about the build of our platform.

2 Holonic MultiAgent Platform

We use holonic principles: in order to understand the workings mechanisms of the human organizations, in which we plan to set up information multi-agent systems; and to design the multi-agents organizations.

Agents of holonic system are organized following levels of responsibility. So, a holonic multi-agent system has a hierarchical structure where each agent is responsible of a holonic multi-agent sub-system.

We have used the social rules defined in the holonic concept in order to simplify and to accelerate the design of a multi-agent society (in the [1] sense), indeed, they provide a framework to build a fixed multi-agent society (which does not imply rigidity).

Applications of holonic MAS lead us to take into account the roles more explicitly. So our approach uses both the notion of role and notion of hierarchy in a same multi-agent system. Indeed, generally, like in the propositions of [2][3], the roles define links between agents; the interactions between agents depend of the roles. In our proposition, the interactions are defined by our MAS architecture. A holonic agent communicates with its coordinator, its neighbours (having the same coordinator) and its assistants, according to theirs roles or not.

We defined each agent by its name, its location, its acquaintances, its knowledge/or beliefs (sets of data relative to the environment, the acquaintances and the agent itself) and its role/s. Each role is defined by some knowledge (sets of resources that are necessary to play the role), and some behaviours (or skills). Finally each behaviour is composed of a set of methods that describe the interactions with the knowledge and with the other agents.

For example, we can define a seller role composed of an inventory management behaviour and a financial behaviour. The inventory management behaviour contains, for instance, the procedures 'inventory control', 'products order'. The financial behaviour contains procedures as the 'account management', 'price management'.

The buyer role could be composed of a call for proposals behaviour, which contains the procedures 'send proposals', 'wait responses', 'choose best proposal', ...

Each agent of the MAS is not limited to one role. So, we can find in the MAS, several seller agents, buyer agents and seller-buyer agents. Likewise, the behaviours are not used only in one role but can be used in other roles.

We have defined notion of essential behaviour and secondary behaviour for a role. Indeed, agents have to be able to modify the behaviours associated to their roles (by receiving a new version of these ones from the user or others agents for example, or by deleting one to free resources). However, an agent that deletes an essential behaviour of a role must be removed from the list of agents that play this role in the

MAS. In our example, a secondary behaviour could be a complex display of the stock for the inventory agent, behaviour that could be updated or remove in run time.

Our architecture can be summarized by the following notation:

```

MAS = {role*, agent*}
agent = {name, location, coordinator:agent, assistant:agent*, knowledge*, role*}
role = {name, knowledge, essentialBehaviour:behaviour*, secondaryBehaviour:behaviour*}
behaviour = {name, attribute*, procedure*}
knowledge = {agentKnowledge|environmentalKnowledge|socialKnowledge}

```

To manage agents and roles of the system, we propose the use of a roles manager: the agent, which plays this role (called HRA for Holonic Resources Agent), owns the roles list, the tasks list linked to the roles, and the names of the agents (and their roles) that compose the multi-agent organization. In our architecture, we give this role to an agent located at the top of the system. Relatively to the FIPA multiagent platform architecture, this agent can be considered as an extended Directory Facilitator agent.

Each agent that appears in the multi-agent organization informs the RMA on its name, its location and on the roles that it wishes to play. If the RMA knows the roles, it sends it the behaviours linked to the requested roles and record the agent and its characteristics in its knowledge.

Each agent can modify its own behaviours (if they are not essential behaviours), and thus can have personal behaviours although it has same roles than other agents of the MAS. However, when a role is modified in a same way in a sufficient number of agents (number defined by a threshold relative to the number of agents playing the role in the whole organization), the role description is updated in the knowledge of the roles manager. Each agent, which asks for a role, receives the last version of this one.

We defined also the “*holonic role*” that describes the tasks relative to the management of the holonic acquaintances and the communication with them. Each agent of our organization owns this holonic role. It allows them to send messages to their acquaintances (the coordinator, the neighbours, the assistants) and the set of holonic agents whose they are the direct or indirect coordinators. The messages can be addressed to acquaintances having particular roles.

As the HRA is essential in our system, we have developed replication mechanisms for it, and we use in fact a hierarchy of sub-HRA, that are each dedicated to the roles management of an agents layer. These sub-HRA are not replicated but recreated by the main HRA if needed. These developments are currently under tests.

We do not have yet proposed particular communication protocols, all the “classical” protocols can be used, but we plan to propose adaptations of existing protocols to respect the holonic concepts.

The first prototype of this platform has been developed using the MAGIQUE [4] multi-agent platform and we have built an application that allows us to deploy into a network a holonic multi-agent system from its description (by a XML file).

3 Holonic multi-agent platform implementation

We generally use the MAGIQUE [4] platform to build our prototypes. This platform provides libraries dedicated to hierarchical multi-agent systems development: a MAGIQUE agent is an empty shell having only communication capacities (with its supervisor and its team (agents under it)); an agent skill is a Java class, composed of functions or sub-processes; messages exchanged between agents consist in calls to functions or to sub-processes that are located in the skills¹.

In order to generate holonic multi-agent systems, we have defined two skills:

- HolonicOrganizationSkill, associated to the highest multi-agent organization supervisor. It owns the roles list, the skills list linked to the roles, and the names and the roles of the agents that compose the multi-agent organization.
- HolonSkill, linked to each agent. It defines the attribution of the neighbours and assistants, and the role oriented communication processes. A holonic agent is able to send a message to its neighbours or assistants having a particular role.

We have built an application that allows us to deploy into a network a holonic multi-agent system from an XML file that describes it. This XML file describes the roles, the agents, their hierarchic relations and the methods of the skills, associated to the roles linked to the agents, which are to be launched at the start of the multi-agent organization.

MAGIQUE platform allows us to easily create hierarchical multi-agent systems, but it is not yet FIPA compliant. We are currently developing our holonic platform as an extension to JADE [5], in order to have a FIPA compliant multi-agent platform. We hope that this will facilitate exchanges with MAS built on others FIPA compliant platform.

4 Conclusion

Our works use the holonic concept to understand and analyse human organizations and design multi-agent system particularly adapted to the studied human organizations. From this work, we propose a multi-agent organization using both notion of role and notion of hierarchy. The tools that we use to automatically deploy such an organization allow us to focus and to develop only on the role played by the agents and avoid us to have to develop the global organization. This platform allowed us to develop several IMAS, but need to be improved. So, currently we “consolidate” our platform by replication mechanisms and regeneration of essential agents.

¹ If a request cannot be satisfied, because the agent does not know how to answer (the function asked is not present in its skills), it is stored by the platform until the agent learns to answer to it.

References

[1]

Bibliography

Mandiau, R. ; Le Strugeo, E. ; Agimont, G.. Study of the influence of organizational structure on the efficiency of a multi-agent system. *Networking and Information Systems Journal*, 2(2), (1999), 153-179

1. Mandiau, R., Le Strugeon E. & Agimont G. Study of the influence of organizational structure on the efficiency of a multi-agent system. *Networking and Information Systems Journal*, 2(2) (1999) 153-179.
2. Mathieu, P., Routier, J-C., Secq, Y. RIO: Roles, Interactions and Organizations. *CEEMAS 2003: Multi-Agent Systems and Applications III, Lecture Notes in Artificial Intelligence* 2691. (2003)
3. Kendall, E. A. Role modeling for agent system analysis, design, and implementation. In *First International Symposium on Agent Systems and Applications (ASA'99)*, Third International Symposium on Mobile Agents (MA'99), (1999).
4. Mathieu, P., Routier, J-C., Secq, Y. Dynamic skills learning: a support to agent evolution. *Proceedings of AISB'01*, York. ISBN 1 902956 17 0. (2001) 25-32.
5. Bellifemine F., Caire G., Poggi A., Rimassa G., "JADE - A White Paper", Sept. 2003, <http://jade.tilab.com/papers/WhitePaperJADEEXP.pdf>