Designing a multi-agent environment by using network management concepts & standards

Babak Esfandiari (LIRMM)
Joël Quinqueton (INRIA-LIRMM)
Richard Terrat(LIRMM)
161 rue Ada F-34392 Montpellier Cedex 5
esfandia@lune.lirmm.fr, {jq,terrat}@lirmm.fr

1 Introduction

If the use of Distributed Artificial Intelligence in Network Management seems to be promising, the study of network management concepts and standards has also much to offer in the understanding of communication between autonomous entities :

1. how to choose a minimal language and an appropriate protocol to enable a good information exchange
2. how to represent information.

ISO and CCITT, the most important standardisation institutes, have already designed an object-oriented model in order to depict network entities (G.D.M.O.) and a query language (C.M.I.S.) used by management platforms to retrieve information from network nodes. Our simulation of a network management platform seems to lead to an actual autonomous entity capable of interaction with its environment and cooperation with other entities. This platform will be used to test several ways of introducing Artificial Intelligence in network management :

1. Intelligent assistance to a network supervision operator
2. Definition of interaction protocols between network management entities.

2 MAGENTA : a Simulation of a Network Management Platform

2.1 MAGENTA's Architecture and Implementation

MAGENTA's first goal was to be a simplified simulation of a Network Management Platform which respects network management standards such as GDMO and CMIS. But we found out that MAGENTA could also be used as a true multi-agent generic platform. MAGENTA implements the OSI Manager-Agent model : MAGENTA is a Smalltalk program which can play either the manager or the agent role, and that communicates with other Smalltalk programs (which can run on distant machines) via a TCP/IP protocol. Smalltalk has the advantage of being object-oriented, which makes GDMO very easy to apply. The genericity of MAGENTA is also mostly due to the choice of this language.

The platform can be divided into 4 distinct parts :

- The communication manager, which receives and sends CMIS-like messages to other agents through Unix sockets
- The object manager, which processes the CMIS-like queries (GET, SET, ACTION, CREATE, DELETE) and updates data (a sorted collection of managed objects)

- The event manager, which determines if the objects modifications worth sending event notifications (EVENT-REPORT) to other agents
- The management application, which decides, depending on received events, how to behave. It is able, for instance, to compose queries and send them to other agents via the communication manager. This part is the only one specific to each problem. It can simply consist in an interface with a human operator.

2.2 Programming a Multi-Agent Application with MAGENTA

MAGENTA facilitates the implementation of multi-agent applications by offering ready-to-use communication tools, a minimal set of speach acts represented by the CMIS primitives, and an adequate mechanism to generate and interpret these primitives. Therefore here are the steps to follow for each agent to implement an application :

1. First create the classes and their initial instances. The default "behaviour" of the classes will derivate from the "GDMO-Object" class.
2. Then determine the conditions of sending an event report for each class.
3. Implement the behaviour of the "Management Application" when receiving EVENT-REPORTs. This part should be augmented in the future to enclose cognitive notions such as a "Mental State" [7] including "commitment".

It would be highly useful to augment the platform with a presentation protocol. Here again, the Network Management standards will be very helpful.

3 Two possible uses for MAGENTA

3.1 Intelligent assistance to a network supervision operator

This is mainly inspired by [6]. The goal is to implement an intelligent assistant who acquires its knowledge by "looking over the shoulder" of the operator and uses this knowledge in order to replace him in case of repetitive tasks such as filtering useless alarms, and thus allows him to better concentrate on important events. An event has to be related to its context, and time is a very important factor. Therefore the patterns to be recognized and learnt are *chronicles* [5], which are sequences of events terminated by an action, with time constraints between the events. A particular unit will afterwards use the "chronicle base" in order to recognize in realtime the sequence of events and choose the proper action. Christophe Dousson [2] has already implemented a Chronicle Recognition System. We have so far imagined incremental algorithms to learn the chronicles, but it would be off-topic to describe them in this paper.

MAGENTA can be used here as a simulation of a network management platform. Events will be EVENT-REPORT messages received by the manager, and actions will be queries (GET, SET, CREATE, ACTION, DELETE).

3.2 Definition of Interaction Protocols between network management entities

Another idea is to introduce more cooperation between managers and agents in order to decrease the sum of exchanged messages. This can be done by using specific interaction protocols [1] [3]. These protocols may demand the extension of CMIS primitives. KQML *performatives* can be a good source of inspiration [4].

MAGENTA can be used here as an actual multi-agent system, where interaction protocols would represent the upper layer.

4 Acknowledgements

This work is part of the "Réseau Futé" project, which is supported by CNET under contract no 93 1B 141/142/143 5115. by

References

[1] Yves Demazeau, Olivier Boissier, and Jean Luc Koning. Using interaction protocols to control vision systems. In *International Conference on Systems, Man and Cybernetics*, San Antonio, Octobre 1994. IEEE.

[2] Christophe Dousson. *Suivi d'évolutions et reconnaissance de chroniques.* Thèse d'université, Université Paul Sabatier, Toulouse, LAAS, Septembre 1994.

[3] Babak Esfandiari, Joël Quinqueton, Jean-Luc Koning, and Yves Demazeau. Systèmes multi-agents et gestion de réseaux. In *Actes des JN du PRC-GDR Intelligence Artificielle*, pages 317–330, Nancy, 1995. Teknea. rr LIRMM 94077.

[4] M. R. Genesereth and S. P. Ketchpel. Software agents. *Communications of the ACM*, 37(7):100–105, Juillet 1994.

[5] Malik Ghallab. Past and future chronicles for supervision and planning. In Jean Paul Haton, editor, *Proceedings of the 14th Int. Avignon Conference*, pages 23–34. EC2 and AFIA, EC2, Paris, Juin 1994.

[6] Pattie Maes and Robyn Kozierok. Learning interface agents. In *Proceedings of the 11th Nat Conf on Artificial Intelligence.* AAAI, MIT-Press/AAAI-Press, 1993.

[7] Yoav Shoham. Agent oriented programming. Research Report STAN-CS-90-1335, Stanford University, Octobre 1990.