

Using the Active Object Model to Implement Multi-Agent Systems

E. Cardozo

J.S. Sichman *

Y. Demazeau †

DCA/FEE

LIFIA/IMAG

LIFIA/IMAG

UNICAMP

46, avenue Félix Viallet

46, avenue Félix Viallet

13081, Campinas, Brazil

38031, Grenoble, France

38031, Grenoble, France

Abstract

This paper discusses the implementation and runtime support for Multi-Agent Systems (MAS). We start presenting MAS in the context of Open Distributed Processing (ODP). Next, a model of a cognitive agent being currently developed at LIFIA is detailed. Taking this model as reference, we examine alternatives for supporting cognitive agents in distributed and heterogeneous environments. Finally, a distributed processing tool developed by the authors is presented. This tool follows the active object model and we show that active object and agent are strongly related concepts.

1 Introduction

Since the last decade, computer users and manufacturers were concerned about connectivity. Connectivity has enabled the replacement of large mainframes by a set of small and low priced personal computers and workstations. In order to integrate different computer systems, a large standardization effort is still being conducted. ISO's Open System Interconnection (OSI) model, ANSI's standard UNIX interface (POSIX), Darpa's internetworking transport protocol (TCP/IP) and SUN's network file system (NFS) are good examples of standards favoring connectivity.

Academic research in Distributed Artificial Intelligence (DAI) has started in the late seventies [3], particularly in Distributed Problem Solving (DPS). DPS is a field of DAI that takes advantage of connectivity. Instead of focusing on problem solving *techniques*, DPS relies on *environments* where:

- The problem is solved by a set of autonomous modules, dispersed among several computers connected via network;

- The modules are organized according to an architecture. Architectures define the flow of control and communication among the modules;
- Modules run in parallel in order to speed up the problem solving activity;
- Modules cooperate by solving different pieces of the problem (tasks), or compete by applying different problem solving techniques for the same task. It is the control's responsibility to promote cooperation and to manage competition.

DPS is therefore centered on the design and implementation of architectures. The reuse of a module when the problem changes is not a major concern in DPS.

By the mid-eighties a new standardization effort has begun. This new trend focuses on highly complex network services with the following properties:

- *Distribution*: service providers and customers may be geographically dispersed;
- *Heterogeneity*: different computational systems will provide the same services independent of vendor, technology, configuration, etc.;
- *Openness*: service providers and customers are free to join or leave the environment without external control.

Two major efforts in this line are the ISO's Open Distributed Processing (ODP) standards [8] and the Open Software Foundation's Distributed Computing Environment (DCC) standards.

A new trend in DAI, Multi-Agent Systems (MAS) [6], follows the ODP philosophy. Like ODP, the idea behind MAS is to define *agents* able to provide complex services. Agents have the following properties in common:

- Agents are autonomous in the sense they can define their own internal goals and plans;

*On leave from PCS-EPUSP, Brazil, and supported by FAPESP, grant number 91/1943-5

†Research Fellow at CNRS