# An Architecture for Modeling Internet-based Collaborative Agent Systems

Roberto A. Flores[1], Rob C. Kremer[1], Douglas H. Norrie[2]

[1] Department of Computer Science, University of Calgary, 2500 University Dr. NW, Calgary, Canada, T2N 1N4
{robertof, kremer}@cpsc.ucalgary.ca
[2] Department of Mechanical & Manufacturing Engineering, University of Calgary, 2500 University Dr. NW, Calgary, Canada, T2N 1N4
norrie@enme.ucalgary.ca

**Abstract.** This paper describes an architecture for modeling cooperating systems of communicating agents. The authors' goal is not that of providing a framework to implement multi-agents systems (there are tools—such as CORBA, Java and DCOM—that do an excellent job on that), but rather to provide an architectural metaphor upon which collaborative multi-agent systems could be modeled. The approach is based on requirements defined with a practical view of the communicational and resource-oriented nature of distributed collaborative multi-agent systems.

## 1 Introduction

Increasingly, the Internet will be used for commerce, industry, and educational interactions between multiple parties. These interactions are intermittent but sustained over days, months, and years. They will involve multiple sources of information and often record, transform, and store considerable quantities of information for subsequent access and re-use. The Collaborative Agent System Architecture (CASA) presented in this paper provides a structural framework to support the modeling of such distributed, collaborative multi-agent systems.

In this architecture, agents are seen as software entities that pursue their objectives while taking into account the resources and skills available to them, and based on their representations of their environment and on the communications they receive [1]. In the case of agents in collaborative systems, agents are also capable of delegating task realization to agents capable of performing the required task. Internet-based systems pose an additional challenge to this scenario in that agents collaborate in a dynamic, distributed environment where agents from heterogeneous sources could interact.

It is common to find collaborative mechanisms in currently available implementation frameworks; however, modeling systems based on these facilities result in systems with less flexibility and adaptation to changes given previous design commitments to the underlying framework. The collaborative architecture presented here aims to separate the modeling of multi-agent systems from the specifications that

T. Wagner and O.F. Rana (Eds.): Infrastructure for Agents, LNAI 1887, pp. 56–63, 2001.

designers need to commit given the low-level mechanisms of proprietary frameworks used in the implementation of multi-agent systems.

There are three elementary components that we have identified as fundamental in the design of collaborative multi-agent systems: computer resources, agents, and owners. These are interrelated, since both agents and computer resources are bound by the regulations set by the human institutions owning and controlling those agents and resources. The latter are further described below.

- Computer resources: Computer resources are hardware and software resources that are available to agents for the execution of their tasks
- Agents: Agents are communicational and collaborative entities that perform their duties using the computational resources available to them. Agents can take the role of requesters or suppliers without these being one of their intrinsic characteristics. Agent requesters and providers rely on the definition of roles and on mechanisms of advertisement to locate other agents that can perform tasks for them.

## 2 Architecture Requirements

In this architecture, we model communities of agents in two ways: based on the computer resources agents use, and on the communicational contexts upon which they interact.

On the one hand, agents are seen as communicational entities that interact with other agents to achieve their goals. In this view, agent communities are formed as a result of interactions and the preferred locations in which interactions take place.

On the other hand, agents are seen as entities with affiliation, performing their tasks on behalf of human institutions that endorse and exert authority over them. In this view, affiliation enables resource-oriented communities by binding agents and resources. It is common that affiliation is addressed in terms of the low-level mechanisms implemented by different frameworks. We believe that these decisions should be reflected at the modeling stage, independently from the proprietary mechanisms offered by implementation frameworks.

Based on the concepts introduced above, we present several minimal requirements upon which we base our architecture.

1. The architecture should provide means (for humans) to organize computer resources in identifiable clusters.
2. The architecture should provide means to control agents' usage of computer resources according to human policies.
3. The architecture should provide means for agents to locate other agents for the purpose of collaboration and delegation of tasks.
4. The architecture should provide means for agents to create and maintain settings where agents interact (settings may include the state and history of agent interactions, and any artifacts pertinent to the context of interaction).