A Hybrid Approach for Directory Facilitators in a FIPA Multi-agent Platform

Fernando A. Martin
fam@cs.uns.edu.ar

Alejandro J. García
ajg@cs.uns.edu.ar

Grupo de Robótica Cognitiva
Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA)
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Avda. Alem 1253 - (B8000CPB) Bahía Blanca
Tel: ++54 291 4595135 - Fax: ++54 291 4595136

In this work we present a research line that focus on the design and implementation of a Directory Facilitator for a FIPA multi-agent platform. We will introduce three different approaches: a centralized version, a distributed one, and a hybrid solution. As we will explain below, the hybrid approach will have several advantages and it will be our choice for future implementations.

1 The FIPA Agent Platform

This investigation is based on the standards provided by the Foundation for Intelligent Physical Agents (FIPA) [4]. According to FIPA, an Agent Platform (AP) should provide the physical infrastructure in which agents can be deployed. This infrastructure should contain, besides the hardware, three main logical component: A Message Transport System, an Agent Management System, and the Directory Facilitator (see Figure 1). We will briefly describe them next. More details can be found in the FIPA Agent Management Specification [3].

The Message Transport System (MTS) is the default low-level communication between agents, not only in the same AP, but between agents in different AP's. The Agent Management System (AMS) is a mandatory component of the AP. Each AP will have only one AMS and its primary objective is to offer white pages services to the agents in the AP. The AMS has the location of every agent in the system, and his function is to provide to authorized agents the location of anyone in the platform. The Directory Facilitator (DF) provides yellow pages to other agents. Agents may register their services with the DF or query the DF to find out what services are offered by other agents.

Partially supported by SGCyT Universidad Nacional del Sur, and Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 Nro 13096)

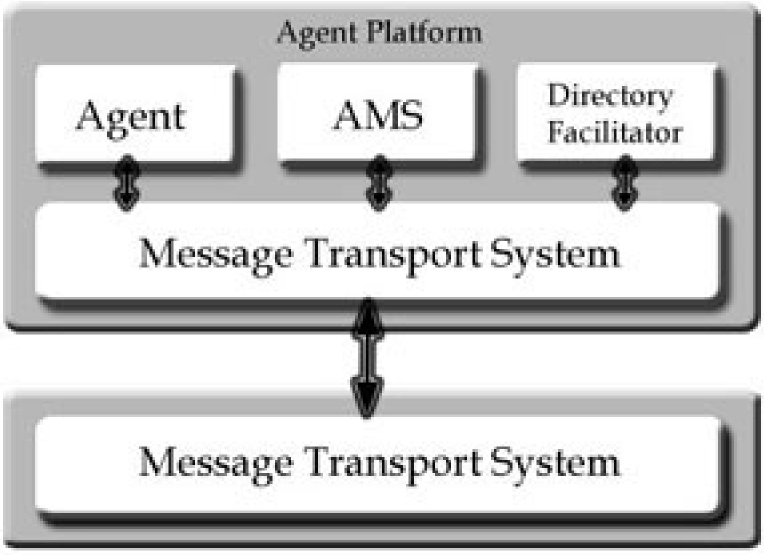

Figure 1: FIPA Agent Management System

In this research line we will consider the design and implementation of DF. Next we will introduce the main features that FIPA propose, and in the following sections three different approaches for its implementation will be described.

The primary objective of a DF is to provide an environment where agents may register their services or query for the services that are offered by other agents in the AP. The DF, like any other agent in a FIPA AP, uses the FIPA ACL Message Structure [2] as standard way of communication. Agents may interact with a DF with four different communicative acts:

Register: The execution of this function has the effect of registering a new object into the knowledge base of the DF. The DF cannot ensure the accuracy of the information that the agent is registering.

Deregister: An agent may request to remove an item of the knowledge base of the DF. This act liberates the obligation of the DF to broker the information related to the agent.

Modify: In this case, the agent may make a modification of an object that in the past has been inserted in the knowledge base.

Search: An agent may ask to the DF to find one or more agents that can satisfy a particular service. Again, the DF cannot guarantee the accuracy of the information provided in response of the search act.

Many DFs can exist in one AP, and DFs may register with each other to maintain a global consistency of the knowledge base. Though the a facilitator is responsible for granting (or denying) access to the information, it has no responsibility whether the content is either accurately or valid. The decision of the distribution of de DFs is mandatory, and has a direct influence on the overall performance of the AP. Our research will focus on different possible implementations for a Directory Facilitator.

In the following sections two possible approaches for DFs are described and analyzed. A brief explanation of how this two possibilities can be merged to fit the requirements of an Agent Platform will be also introduced.

2 A centralized approach

The simplest way to implement the Directory Facilitator is with only one agent in the entire Platform. Furthermore, possibly the Agent Management System (the one who is unique and has the information about the location of all agent in the AP) and the Directory Facilitator can be merged into one agent. This alternative has two main advantages: simplicity, and the problem of maintaining consistency is solved.

Like any other centralized approach, leaving the responsibility in only one agent has many drawbacks, including, amongst others, poor fault tolerance, scalability related issues, and, depending on the size of the AP, the DF may become a bottleneck in the system. Depending on the population of the platform, and where the agents are physically located, the performance of the centralized approach can drop to unacceptable levels.

This approach can be useful for a platform for a multi-agent system with a small number of agents. This kind of systems avoid the bottleneck issue, and have the advantage of the consistency of the knowledge base, making this implementation feasible. For a system which has no bounds on the number of agents, and furthermore, in an environment where the population can grow (or diminish) dynamically, this approach is not viable.

3 A distributed approach

The other possibility is having several DFs in the system. For every agent in the AP, there is a daemon running on the same machine where the agent is. In this approach, the primary goal is to provide a simple and fast communication between the agent and the facilitator (in this section we use daemon or DF indistinctly). The daemon has two main responsibilities:

- The first is maintaining his knowledge base as complete as possible. The agent depends entirely on the DF, and the only way for the agent to know the services of other agents in the AP is through it. In order to achieve simplicity and velocity (main goals of the this approach), the agent shouldn't wait while the daemon broadcasts the search to the other DFs in the AP, instead, the daemon has to respond the act immediately. To accomplish this, the DF periodically asks to other facilitators about the services of their agents.
- The DF has the responsibility of publishing the services of the agent to other facilitators. The success of the agent relies on that. Any kind of malfunctioning of a daemon (external, just as message problems, or internal, i.e. bad coding of the DF) could affect a group of agents, isolating them.

The benefit of this approach is the simplicity for the agent to gather information of the environment: the agent only needs to communicate with one daemon, and he gets te response immediately. Another advantage is that a failure on one daemon do not affect the entire platform, giving to the system a very high degree of fault tolerance.

Although this approach solves the bottleneck problem stated above, an even worse problem could arise: every daemon has to continuously communicate with every other daemon in the entire AP generating overload. Like the centralized approach, the performance of this system diminishes as the number of DF grows.

The number of DF in a distributed approach should be balanced: if there are only a few, each DF inherits the problems of the centralized approach; if there are too many, the overload can be unacceptable. The number of DF could be fixed, or decided dynamically by some agent.

4 A hybrid approach

The performance of both approaches described above relies on the number of agents in the AP. A fixed number of DF is not a feasible solution for an AP where the population of agents can grow dynamically.

Instead of having a fixed number of DF, our proposal is letting the AMS (Agent Management System) decide when the AP needs another Directory Facilitator. When an agent wants to join an AP, he must send a proper message to the AMS. At this point the AMS can decide whether to create or not a new facilitator in order to balance the number of DF. The AMS will respond to the agent with the address of a facilitator (either is new or not).

In this approach an agent will interact (register, deregister, modify or search for services) with only one DF. Thus, the AMS will know the number of agents associated with each DF. It is important to note, that the AMS only recommends one DF for the agent. The agent is free to ask for the location of any other DF in the AP to interact with. However, the AMS may refuse to associate an agent with an overloaded DF.

The decision process of the AMS can be implemented in many ways. One possibility is having one DF in each subnet. This grants fast communication with entities in the same physical area. When an agent request a search, the DF can try to satisfy the service required with an agent of the same subnet, if the service cannot be satisfied in the area, the facilitator may ask for the service to the other DFs in the AP. This approach solves, in part, the bottleneck issue, and makes the system easier to extend. In case that one subnet is highly overloaded, the AMS may decide to create another DF to diminish the overload.

We are planning to implement this hybrid approach using the framework reported in [5].

References

[1] Agustín Alejandro Denegri. *"Comunicación en Sistemas Multiagentes"*. Tésis de Magister en Ciencias de la Computación. Computer Science Department, Universidad Nacional del Sur, Bahía Blanca, Argentina, July 2002.

[2] FIPA. FIPA ACL Message Structure Specification. http://www.fipa.org/specs/fipa00061/.

[3] FIPA. FIPA Agent Management Specification. http://www.fipa.org/specs/fipa00023/.

[4] FIPA. Foundation for Intelligent Physical Agents. http://www.fipa.org.

[5] Mariano Tucat. *"Primitivas de Interacción para el Desarrollo de Sistemas Multi-agente"*. Proyecto Final de Ingeniería en Sistemas de Computación. Computer Science Department, Universidad Nacional del Sur, Bahía Blanca, Argentina, March 2005.