

A Jabber-based Multi-Agent System Platform *

Miguel Escrivá Gregori
Dept. de Sistemas
Informáticos y Computación
Universidad Politécnica de
Valencia
Camino de Vera s/n
Valencia, Spain
mescriva@dsic.upv.es

Javier Palanca Cámara
Dept. de Sistemas
Informáticos y Computación
Universidad Politécnica de
Valencia
Camino de Vera s/n
Valencia, Spain
jpalanca@dsic.upv.es

Gustavo Aranda Bada
Dept. de Sistemas
Informáticos y Computación
Universidad Politécnica de
Valencia
Camino de Vera s/n
Valencia, Spain
garanda@dsic.upv.es

ABSTRACT

This paper presents a new multi-agent system platform that takes advantage of XMPP. Among the new message transport protocols that have been recently developed, Jabber (XMPP) is a protocol that is based on Instant Messaging networks that use real-time messaging between applications. This platform is fully FIPA compliant and provides a simple interface to create agents using this new communication concept.

1. INTRODUCTION

In this paper we introduce a new multiagent-system platform (called SPADE) that takes advantages of a new communication model and that provides a simple interface to create agents using this new communication concept. This communication model is based on Instant Messaging protocols that use real-time messaging between applications. Agent communication is improved by using this protocol due to its new features, such as the Contact List management for agents, the ability to receive a notification whenever any other known agent (included in its Contact List) comes online or the possibility of engaging a simultaneous multi-conference with many agents.

This new platform is fully FIPA compliant to allow interoperability with other platforms. Its main objective is to provide good performance while retaining strong scalability options and to provide an open design that allows the implementation of platform components in several high-level programming languages.

Our proposal provides a human-like framework to communicate agents. It allows them to have a conversation (with all the meta-information required in a real conversation included) with any other 'entity'. This platform is described

in depth in the following section.

2. SPADE

SPADE (a Smart Python multi-Agent Development Environment) is a new platform aimed at solving the drawbacks of the communication models from other platforms. The multi-agent system platform SPADE is built around a new communication framework (Jabber[1]) which provides new and powerful capabilities to the communication layer. The platform core is designed over this layer, which communicates the platform elements with one another.

2.1 The Communication Model

Communications in SPADE are handled by means of the Jabber protocol. Jabber is the Internet standard protocol for instant messaging and presence communication[1, 2]. Its base technology is the Extensible Messaging and Presence Protocol (XMPP) standard, a set of streaming XML protocols and technologies that enable any two entities on the Internet to exchange messages, presence, and other structured information in close to real-time.

Presence notification is one of the most useful Jabber features that SPADE provides. It is a system by which an entity involved in a Jabber communication can know at any time what the current status of any of the other entities involved is. Jabber entities can declare what their current status is and all other entities who share a bond with them will automatically be notified. An entity status usually indicates its availability and readiness to engage in a communication (e.g. 'Available', 'Busy', 'Don't disturb', ...), but it is not closed to a short predefined list. Entities can define their own status and their meanings. Two or more Jabber entities can be bonded (or *subscribed*) to each other. Entities that share a bond may make use of presence notification as described above.

Another interesting SPADE feature is **Multi-user Conference** (MUC). MUC features can be used to create forums for agents to connect to when performing a multi-agent activity. For instance, in a virtual auction scenario performed by agents, an agent playing the role of the auctioneer can create a password-protected channel and give the password only to previously certified agents. Then, in the channel, the auctioneer can give the right to speak only to agents who are going to play the bidder role, leaving the rest of the agents in spectator mode. Once the auction starts, every time a bidder wants to place a bid, the agent only needs to 'speak' in the channel, and all the other bidders and the auctioneer

*This work is partially supported by the TIC2003-07369-C02-01 and TIN2005-03395 projects of the Spanish government.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.
Copyright 2006 ACM 1-59593-303-4/06/0005 ...\$5.00.

will know the bid at the same time. MUC channels can also be used for broadcasting announcements, public platform information, selected distribution lists, etc.

2.2 The Platform Model

The SPADE platform is modeled according to the FIPA standard proposal for a multi-agent platform [3]. It features standard basic services (such as an Agent Management System and a Directory Facilitator) that are designed as Jabber server components (or add-ins). Although the core communication system relies on the Jabber technology, other message transport protocols, such as the *HTTP* protocol[4], are also supported.

A brief schema of the SPADE Platform can be viewed in Figure 1.

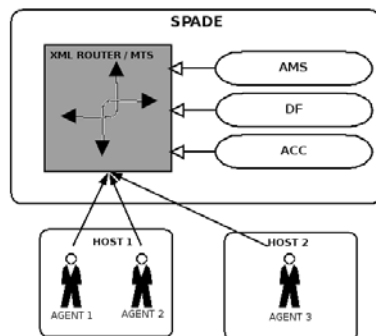


Figure 1: Platform Model

The main platform element is the **XML Router**, which the rest of the platform components and agents are connected to. It is a standard XMPP server[1] that routes all the messages from its sender to the specified receiver with no user intervention. This XML Router acts as the Message Transport System (MTS) and is the only 'third-party' component of the platform. One of this components connected to the router is the **SPADE Agent Communication Channel (ACC)**, which manages all the communication within the platform. It receives the FIPA-ACL¹[5] messages that arrive to the platform and it redirects them to the correct destination element. This destination can be either an agent or another component.

2.2.1 Security Concerns

The SPADE framework provides some built-in security mechanisms that help to maintain the system's integrity. These mechanisms are developed in different security levels:

- ♣ Logging to the XML Router requires a **username and a password**. This mechanism prevents an unauthorized connection to the platform if it is not approved by the platform administrator.
- ♣ A connection to the XML Router can be encoded with a symmetric cryptographic algorithm using SSL (Secure Socket Layer). SSL provides data **cyphering**, server **authentication**, message **integrity** and, optionally, client authentication for TCP-IP connections.

¹FIPA Agent Communication Language

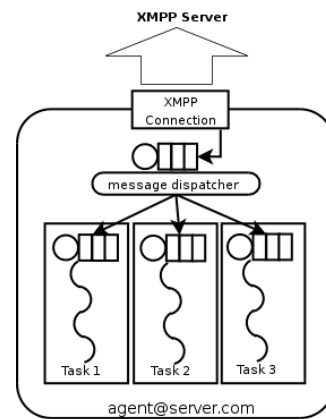


Figure 2: SPADE Agent Model

- ♣ To ensure the authentication of a message, the XML Router provides another mechanism to avoid identity theft. The platform prevents identity spoofing by overriding a 'from' field that does not match the sender's.

2.2.2 Platform Inter-operability

The Message Transport System can manage as many MTPs as there are plugged to the platform. If the protocol is XMPP (Jabber protocol), the MTS delivers the message directly to the receiver agent. This feature allows SPADE to communicate with any kind of FIPA compliant platform that supports the XMPP Message Transport Protocol.

We have also developed a new JADE plug-in that provides a MTP implementing the XMPP protocol. This plug-in has been accepted by the JADE Team and is published in [6]. Thanks to this new MTP, a SPADE platform and a JADE platform can communicate with each other using the Jabber protocol. Actually, two JADE platforms can use this new XMPP plug-in to communicate their agents and take advantage of this new proposed Transport Protocol.

2.3 The Agent Model

Every component inside the platform is built as an agent. SPADE agents are elements that are connected to the MTS that can send messages to each other and to other platforms.

The Agent Model is basically composed of a connection mechanism to the platform, a message dispatcher, and a set of different tasks that the dispatcher gives the messages to (–see Figure 2). Every agent needs an identifier called Jabber ID (JID) and a valid password to establish a connection with the platform. If the platform registration process is disabled, then the platform administrator should define the registration policies. An agent can run several tasks simultaneously. A **task** is a process that an agent can execute using repeating patterns. SPADE provides some predefined task types: Cyclic, One-Shot, Periodic, Time-Out, and Finite State Machine Task. Those task types help to implement different the tasks that an agent can perform. Every agent can have as many tasks as desired. When a message arrives to the agent, the message dispatcher redirects it to the correct task queue (–see Figure 2).

2.4 The Main Features

The main features of SPADE are the following:

1. The SPADE platform has already covered the FIPA standard. AMS and DF ontology is supported.
2. SPADE supports two Message Transport Protocols: HTTP and XMPP. This feature allows different platforms to communicate using this new communication model or even older platforms that make use of the wide-spread HTTP protocol.
3. SPADE supports two different content languages: **FIPA-SL** and **RDF**, which are the two main options for content language used in today's agent communication.
4. Another feature defined by FIPA that SPADE supports is agent **mobility**. A SPADE agent can move from one platform to another authorized platform and continue its execution in the new target.
5. As we stated in Section 2.3, SPADE agents do reach their goals by running one or more tasks.
6. SPADE has a special agent to provide a user-friendly view of the platform. This GUI provides a mechanism to load and unload agents in the platform. It allows the user to search for agents and services and send messages to any agent.
7. SPADE was developed in Python. This allows the execution of the platform in several architectures and operating systems such as Windows, Linux, MacOS, Windows Mobile, PalmOS, SymbianOS for mobile phones, etc. Python also allows using some performance characteristics like *stackless* threads or the *psyco* module to massively speed up the execution.
8. One of the new and powerful feature of SPADE is the possibility to use **Presence Notification** between components. This allows the system to determine the current state of the components that are connected to the platform in real-time.
9. Managing a conversation with more than one agent is possible thanks to a Jabber mechanism. This mechanism (called **multi-user conference**) is based on chat rooms where an agent can join and share messages with a group of agents.
10. The Protocol used to send messages is a distributed client-server protocol. The servers that route the Jabber messages are designed to support a very large number of users and messages. This feature helps to improve the SPADE performance and scalability.
11. The platform provides a user and password authentication to improve security as well as a cyphered connection to the platform to perform confidentiality.

3. CONCLUSIONS AND FUTURE WORK

One of the main drawback on multi-agent platforms is the use of an inappropriate message transport protocol to communicate agents. To solve this, we have presented a new communication model using the Jabber Instant Messaging Protocol and we have developed a new agent platform using

this technology. The new platform, called SPADE, is a fully FIPA compliant platform that supports all the required features. It has been developed in a modern language that increases the productivity of the programmer. The main characteristic of SPADE is the support of the new communication protocol between agents. This protocol, which is based on Instant Messaging systems, uses a distributed network to route messages from one agent to another. It also provides new mechanisms that improve the communicative acts, such as Presence Notification or Multi-User Conference. These characteristics, which are not present in any other agent platform, provide new communication capabilities between agents, which make agents more versatile. SPADE also has some security properties which provide confidentiality and authentication systems to the platform.

SPADE is released under the General Public License (GPL) and can be downloaded and tested at [7]. The plug-in for the JADE multi-agent platform that enables communication through XMPP is also available for download at [7] and [6].

4. ADDITIONAL AUTHORS

Additional authors: Ana García-Fornes, Vicente Julian and Vicent Botti (Dept. de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia).
{agarcia, vinglada, vbotti}@dsic.upv.es

5. REFERENCES

- [1] Jabber Software Foundation. Extensible Messaging and Presence Protocol (XMPP): Core. Technical report, <http://www.ietf.org/rfc/rfc3920.txt>, October 2004.
- [2] Jabber Software Foundation. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. Technical report, <http://www.ietf.org/rfc/rfc3921.txt>, October 2004.
- [3] FIPA. Abstract architecture specification. Technical Report SC00001L, 2002.
- [4] FIPA. HTTP Message Transport Protocol. Technical Report SC00084, 2002.
- [5] FIPA. Agent ACL message structure specification. Technical Report XC00061E, 2001.
- [6] JADE XMPP-Plugin.
<http://jade.tilab.com/community-3rdpartysw.htm>.
- [7] SPADE: Smart Python multi-Agent Development Environment.
<http://gti-ia.dsic.upv.es/projects/magentix/>.