Service Oriented Multi-agent System

T. Thilagam¹, S. Rekha², Manoj Kumar. D. S¹
¹,²,³Asst. Professor, CSE & Gojan School of Business & Technology

Abstract – Agent communication is used for solving standard multi-agent problems, like coordination or negotiation. Coordination and communication among the agents have a critical role for the success of the dynamic composition of new services. Integrating Web services and software agents brings about an obvious benefit: connecting application domains by enabling a Web service to invoke an agent service and vice versa. However, this interconnection is more than simply cross-domain discovery and invocation; it will also allow complex compositions of agent services and Web services to be created. JADE, an agent development framework, is used for creating agent services. Social Model is more promising in agent communication. So Commitment or institution based agent communication on the social model of agents is considered for agent communication. A social model for communication is built as an extension of JADE.

Keywords - Multi-agent System, Social Model, JADE, Web service, Agent services, Commitments, Agent Communication Language.

I. INTRODUCTION

Agents are entities that perceives from the environment using sensors and act on that environment using actuators. Those agents are autonomous, heterogeneous, proactive etc. Multi-agent system is an area where agents are present and communicate or interact with each other to reach a consensus. So communication plays a vital role in a system to function appropriately. The Agent Communication Language (ACL) used must be simple, understandable, extensible and reliable.

Due to the complexity associated with the development of multi-agent systems, which typically involves thread control, message exchange across the network, cognitive ability, and discovery of agents and their services, several architectures and platforms have been proposed. Many agent platforms namely Jason, JACK, Jadex, and the 3APL Platform are available. These four platforms are based on the Java language.

However, even though the underlying language is a general purpose programming language, agents are implemented in these platforms in a new programming language – Agent Speak (L), JACK Agent Language, a Domain specific Language (DSL) written in XML, and 3APL, respectively. Source code written in these languages is either precompiled or processed at runtime by the agent platform.

The adoption of this approach prevents developers from using advanced features of the Java language and it makes it complicated to integrate the implementation of a multi-agent system with existing technologies.

JADE is an agent framework which facilitates the development of MAS. JADE is largely an implementation of the FIPA (Foundation for Intelligent Physical Agent) specifications. It provides a runtime environment where JADE agents can "live", a library of classes that programmers can use to develop their agents and graphical tools that allows administrating and monitoring the activity of running agents. We built a social model for agent communication as an extension of existing JADE platform.

This paper is organized as follows. Section II is about the Agent Communication Language. Section III is about the brief description of the work done in this paper, Section IV is about the integration of agents’ services and web services. The conclusions and remarks are presented in Section V.

II. AGENT COMMUNICATION LANGUAGE

There are number of Agent Communication Languages (ACLs) namely KQML and FIPA-ACL. The message format in JADE uses FIPA-ACL and it has the parameters like sender, receiver(s), performatives (communicative acts), protocol, ontology, content, language, conversation-id, etc. Of all these, Communicative Acts (CAs) are mandatory fields. They indicate the action to be taken. Examples of performatives are REQUEST, INFORM, QUERY, AGREE, PROPOSE, CFP, ACCEPT_PROPOSAL, etc.

The FIPA supports 22 CAs. The parameters of a FIPA-ACL message is shown in Table 1

III. RELATED WORK

A Social model for agent communication is based on exchanging messages. The meaning of the messages passed is based on social concepts like commitments and conventions. A social semantics naturally lends itself to observation and verification whereas in mentalist approach, the internal states of the agents are not observable and hence not verifiable.

A. Social Commitment

A (social) commitment is an elementary social relation between two agents. A debtor commits to a creditor to bring about a specified consequent if a specified antecedent obtains.
For example, in the common purchase setting, one can specify the meaning of the offer message as creating a commitment from the merchant to the customer for the delivery of goods in return for payment. Commitments are distinct from arbitrary obligations: commitments may be created, discharged, delegated, or otherwise manipulated only by explicit communication among agents. Commitments are public and it can be used as the basis for compliance. Commitments support the following key properties that make them a useful computational abstraction for service-oriented architectures.

Commitments can be written using a predicate C. A commitment has the form \( C(x, y, p, G) \), where \( x \) is its debtor, \( y \) its creditor, \( p \) the condition the debtor will bring about, and \( G \) a multi-agent system which serves as the organizational context for the given commitment.

Table 1: Parameters of ACL message.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performatives</td>
<td>Type of the communicative act of the message</td>
</tr>
<tr>
<td>Sender</td>
<td>Identity of the sender of the message</td>
</tr>
<tr>
<td>Receiver</td>
<td>Identity of the intended recipients of the message</td>
</tr>
<tr>
<td>Content</td>
<td>Content of the message</td>
</tr>
<tr>
<td>Language</td>
<td>Language in which the content parameter is expressed</td>
</tr>
<tr>
<td>Protocol</td>
<td>Interaction protocol used to structure a conversation</td>
</tr>
<tr>
<td>Conversation-id</td>
<td>Unique identity of a conversation thread</td>
</tr>
</tbody>
</table>

B. Policy

The performatives in agent communication acts (messages), are translated (by a set of polices) to a set of social commitment operators, which either add or delete a specific class of social commitments. We model a social commitment as the promise by a debtor agent to a creditor agent(s) to do some action:

\((\text{Debtor; creditor; action})\)

And we model a social commitment operator as either an add or delete of a social commitment:

\((\text{add/delet}; \text{social Commitment})\)

We have defined several polices (e.g. propose, accept, reject, counter, and inform) which can be applied to an agent’s outgoing and incoming messages and set of social commitment operators. The description of various policies is shown in Table 2.

The policy selects a performative to be sent back by the receiver. For ex., if a sender sends an INFORM message, the receiver(s) is/are committed to send back the ACK message to the sender according to the policy, \( P_{\text{Ack}} \). But according to policy \( P_{\text{Agree}} \), if the receiver sends an AGREE message, then the commitment to send ACK message does not exist.

Table 2: Informal Description of Policies.

<table>
<thead>
<tr>
<th>Policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-Inform</td>
<td>commits the addressee to acknowledge</td>
</tr>
<tr>
<td>P-Ack</td>
<td>releases informed agents of the commitment to acknowledge</td>
</tr>
<tr>
<td>P-Request</td>
<td>commits the proposed agents to reply</td>
</tr>
<tr>
<td>P-Counteroffer</td>
<td>commits addressees to reply</td>
</tr>
<tr>
<td>P-Reply</td>
<td>releases proposed agents of the commitment to reply and releases counter offered agents of the commitment to reply</td>
</tr>
<tr>
<td>P-Agree</td>
<td>an acceptance realizes the shared uptake of proposed/counteroffered commitments</td>
</tr>
<tr>
<td>P-Done</td>
<td>releases accepted agents of the commitment earlier agree</td>
</tr>
</tbody>
</table>

C. An Example

Let us consider a simple example where two agents A and B communicate and the resulting commitments between them are noticed.

Message 1: A sends a message with a REQUEST performative to B. The content of the message describes A’s call for meeting.
So the existing Commitment between A and B are:

1. According to policy P_Request, the commitment (B, A, reply) exists.
2. According to policy P_Inform, the commitment (B, A, ack) exists.

i.e., B (debtor) is committed to send REPLY and ACK performative to A (creditor). The commitments are added.

1. (add, (B, A, reply))
2. (add, (B, A, ack)).

Message 2: Now B sends back a message with an AGREE performative. Since it has replied using AGREE performative, it need not acknowledge again. So few commitments are added and existing commitments are deleted. The commitments that already exist are deleted and a new commitment is added.

1. (delete, (B, A, reply)).
2. (delete, (B, A, ack)).
3. (add, (A, B, ack)).

Message 3: Again A sends message with an ACK performative to B to acknowledge the reply sent by B. Now all the commitments have been deleted.

(delete, (A, B, ack)).

Figure 1: A simple interaction among agents in JADE

After the communication has taken place, if there are any pending commitments, we conclude that the commitments are violated. Hence this social commitment model serves as a basis for compliance checking.

D. Operations

The commitments can be created, cancelled, discharged, and delegated. For ex., a buyer is committed to pay money after the arrival of goods. But if the arrived goods are damaged then the commitments can be cancelled.

The sender can delegate its commitment to some other agent also. i.e., the receiver is not committed to pay now; it can delegate its commitment to some other agent.

IV. INTEGRATION OF AGENTS’ SERVICES AND WEB SERVICES

A. JADE

The JADE internal Architecture is shown in Figure 2. The WSIG is provided as an add-on by the JADE, through which the web service can call the agent’s service and vice versa.

The JADE architecture is shown in Figure 2. The JADE default agents are AMS and DF.

The Directory facilitator takes care of yellow page services, which is similar to UDDI registry of web services.

Figure 2: JADE architecture.

B. WSIG

The Web Service Integration Gateway is shown in Figure 3. The JADE agent gateway is a specialized JADE agent, that manages the entire WSIG system.

Its operations are:

1. Receive and translate agent service registrations from the JADE DF into corresponding WSDL descriptions and register these with the UDDI repository as tModels. This also applies to deregistration and modifications.
2. Receive and translate Web service operation registrations from the UDDI repository into corresponding ACL descriptions and register these with the JADE DF. This also applies to deregistration and modifications.
3. Receive and process Web service invocation requests received from JADE agents. Processing includes retrieving the appropriate tModel from the UDDI repository, translating the invocation message into SOAP and sending it to the specified Web service. Any response from the Web service will be translated back into ACL and sent to the originating JADE agent.
4. Receive and process agent service invocation requests received from Web service clients. Processing includes retrieving the appropriate tModel from the UDDI repository, translating the invocation message into ACL and sending it to the specified agent. Any response from the agent will be translated back into SOAP and sent to the originating Web service.
C. Compliance Checking

Using the social commitment model, we can check whether the communication is taking place in the right manner for dynamic composition of services. The social model helps in compliance checking whereas other agent communication techniques will not support in compliance checking.

V. CONCLUSION

FIPA performatives are not enough to demonstrate the social commitment. Additional performatives are added to JADE by extending existing classes and Social Model for agent communication is to be built on the top of JADE. Services are composed and compliance checking is done.