An Ontology-based Methodology for Communicating Negotiation Agents over Internet

Azucena Montes[1], Maricela Bravo[1] and José C. Velázquez[2]

[1]Centro Nacional de Investigación y Desarrollo Tecnológico
[2]Instituto de Investigaciones Eléctricas

Abstract

In this article we describe a new methodology to facilitate communications between heterogeneous negotiation agents; in particular we present an ontology solution which allows the publication and sharing of communication primitives over Internet. Traditional negotiation systems exchange messages based on particular language definitions implicitly encoded, giving different syntax and semantics to their messages. We implemented a Web service-oriented architecture for executing experiments with different negotiation agents. The results of experiments show that the proposed methodology improves the communication between heterogeneous negotiation agents.

Keywords: Ontology, Negotiation, Agents.

1. Introduction

Negotiation plays a fundamental role in electronic commerce activities, allowing participants to interact and take decisions for mutual benefit. Traditional negotiation systems have been implemented in controlled and homogeneous agent-based environments, where the number of agents is limited, and the agent communication language is fixed previously, reducing the complexity. Recently there has been a growing interest in conducting negotiations over Internet, and constructing large-scale agent communities based on emergent Web service architectures. The challenge of integrating and deploying multiple negotiation agents in open and dynamic environments is to achieve effective communications.

Agent communication language (ACL) allows an agent to share information and knowledge with other agents, or request the execution of a task. KQML [1] was the first standardized ACL from the ARPA knowledge project. KQML consists of a set of communication primitives aiming to support interaction between agents. KQML includes many performatives of speech acts. Another ACL [2] standard comes from the Foundation for Intelligent Physical Agents (FIPA) initiative. FIPA ACL is also based on speech act theory, and the messages generated are considered as communicative acts.

The objective of using a standard ACL is to achieve effective communication without misunderstandings, but this is not always true. Because, standards specify the semantics of communicative acts, but the software implementation is not explicitly defined, leaving developers to follow their own criteria. Furthermore, standard ACL specifications consider the incorporation of privately developed communicative acts. In this paper we describe a new methodology to solve the language heterogeneity problem. This methodology is based in the incorporation of an ontology which has been previously designed and described in [3].

2. Methodology

In this section we describe the methodology for facilitating communications between heterogeneous agents.

- Describe negotiation agents. The first step of the methodology is to describe clearly the agent data, for instance: company name, identification, agent electronic address (URL), negotiation parameters (desired price, quantity, quality…), and the set of negotiation primitives with their textual description.
- The second step of our methodology is an iterative process: for each negotiation primitive identify to which class it belongs to. The general classification of negotiation primitives is based on the work presented by Jürgen Müller in [4], he classifies negotiation primitives into three groups: *initiators*, if they initiate a negotiation, *reactors*, if they react on a given statement and *completers*, whether they complete a negotiation. The process of

classifying negotiation primitives depends on the description provided by the developer.

- The third step of the methodology consist of aligning negotiation primitives using finite state machines, in order to compare the primitive's intended usage in the negotiation protocol. We believe that a separated analysis between primitive description and protocol allows a clearer understanding of the intended usage.
- Based on the previous classification and analysis the fourth step consists of establishing new relations of equality and similarity between negotiation primitives.
- The next step is the codification and publication of negotiation primitives in the ontology.
- The final step is the execution of negotiations between agents.

3. Validation of the Methodology

To validate the methodology we have executed a set of experiments. In this section we describe the application of the methodology in one of these experiments to show the contribution of our approach.

3.1. Describe Negotiation Agents

Let A and B be the names of the negotiation agents that will be described next.

Table 1. Description of agent A

Agent A	Values
Identification	ID: 00015 Company: "Firefox" Acting as "Buyer"
Negotiation values	Quantity: 1500 Max. Payment: $1,000
Language	{(CFP, "Initiate a negotiation process by calling for proposals"), (Propose, "Issue a proposal or a counterproposal"), (Accept, "Accept the terms specified in a proposal without further modifications"), (Terminate, "Unilaterally terminate the current negotiation process"), (Reject, "Reject the current proposal with or without an attached explanation"), (Acknowledge, "Acknowledge the receipt of a message"), (Modify, "Modify the proposal that was sent last"), (Withdraw, "Withdraw the last proposal")}

Table 2. Description of agent B

Agent B	Values
Identification	ID: 00012 Company: "Tlahuica" Acting as "Seller"
Negotiation values	Initial selling price: $2,500 Last selling price: $1,750
Language	{(Initial_offer, "Send initial offer"), (RFQ, "Send request for quote"), (Accept, "Accept offer"), (Reject, "Reject offer"), (Offer, "Send offer"), (Counter-offer, "Send counter offer")}

3.2. Classification of Primitives

Analyzing the description of each negotiation primitive we can identify to which classification it belongs. Table 3 presents the classification result of negotiation primitives of agents A and B.

Table 3. Classification of primitives

Agent	Starter	Reactor	Completer
A (Buyer)	CFP	Propose Modify Withdraw Acknowledge	Accept Reject Terminate NotUnderstood
B (Seller)	RFQ	Initial_Offer Offer Counter_Offer	Accept Reject NotUnderstood

3.3. Align Primitives

To compare the intended usage of negotiation primitives we selected a finite state machine diagram, because is an easy formalism to describe negotiation protocols.

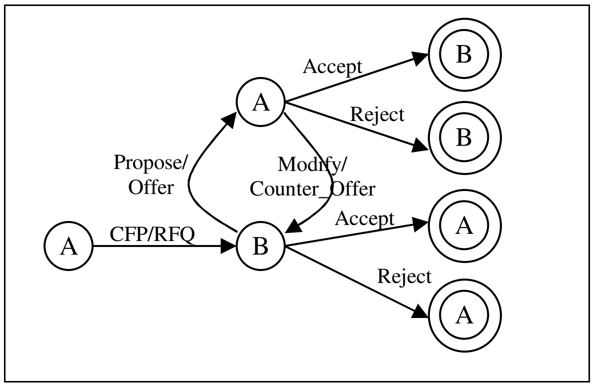

Fig. 1: Alignment of primitives in a finite state machine.

3.4. Establish Relationships

Based on the finite state machine diagram, we can identify relationships between primitives. The explicit relations between primitives that need to be coded in the ontology are those that have syntactic differences and will not allow communication. These relations are shown in table 4.

Table 4. New relations between primitives

Agent A	Relation	Agent B
CFP	Is synonym of	RFQ
Propose	Is synonym of	Offer
Propose	Is synonym of	Initial_Offer
Modify	Is synonym of	Counter_offer
Withdraw	Is similar to	Counter:Offer
Terminate	Is similar to	Reject

In this case the *Acknowledge* primitive is left with no relation; although there is the possibility that agent B can incorporate it.

3.5. Publication of Primitives

For the publication of primitives and relations we are using the ontology language OWL, because it is the most recent development in standard ontology languages from the World Wide Web Consortium (W3C)[1]. We use Protégé [5, 6], an open platform for ontology modeling and knowledge acquisition. Protégé has an OWL Plugin, which can be used to edit OWL ontologies, to access description logic reasoners, and to acquire instances of semantic markup.

3.6. Execution of Negotiations

For the execution of experiments we implemented the system architecture illustrated in figure 2. In this section we briefly describe the functionality and implementation techniques for each component.

(1). *Matchmaker* is a Java module which is continuously browsing buyer registries and seller descriptions, searching for coincidences.

(2). *Negotiation process* is a BPEL4WS-based engine that controls the execution of negotiation processes between multiple agents according to the predefined protocols. BPEL4WS provides a language for the formal specification of business processes and business interaction protocols. The interaction with each partner occurs through Web service interfaces, and the structure of the relationship at the interface level is encapsulated in what is called a partner link.

(3). *Seller and buyer agents* are software entities used by their respective owners to establish their preferences and negotiation strategies. For example, a seller agent will be programmed to maximize his profit, establishing the lowest acceptable price and the desired price for selling. In contrast, a buyer agent is seeking to minimize his payment. On designing the negotiation agents, we identified three core elements, strategies, the set of messages and the protocol for executing the negotiation process. The requirements for these elements were specified as follows:

a. Strategies should be private to each agent, because they are competing and they should not show their intentions.
b. Messages should be generated privately.
c. The negotiation protocol should be public or shared by all agents participating, in order to have the same set of rules for interaction.

(4). *Translator* is invoked whenever the agent misunderstands a negotiation message from another agent. The translator module was implemented using Jena[2], a framework for building Semantic Web applications. It provides a programmatic environment for OWL, including a rule-based inference engine.

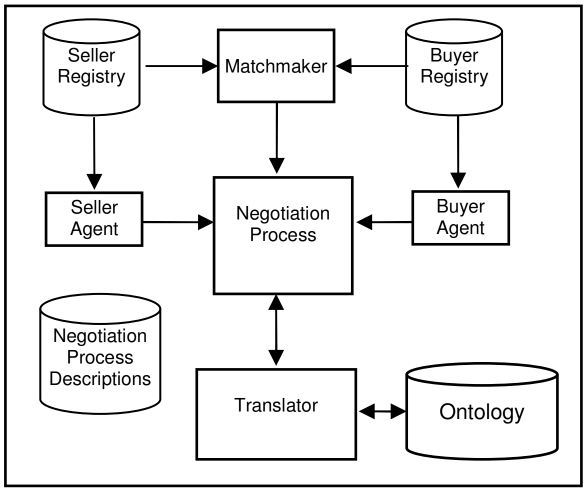

Fig. 2: Negotiation system architecture.

4. Results

We executed 15 test with agents A and B, generating random values for the negotiation parameters of seller and buyer. Table 5 shows the results of these tests.

[1] http://www.w3.org

[2] http://jena.sourceforge.net

Table 5. Experimental results

Test	Iterations	Quantity	Final Price	Result
1	12	1500	$ -	Reject
2	3	887	$ 1,674.00	Accept
3	12	1660	$ -	Reject
4	12	1270	$ -	Reject
5	2	1475	$ -	notUnderstood
6	12	56	$ -	Reject
7	7	8	$ 614.00	Accept
8	9	53	$ 2,137.00	Accept
9	9	56	$ 1,965.00	Accept
10	12	81	$ -	Reject
11	2	41	$ 1,172.00	Accept
12	2	67	$ -	notUnderstood
13	2	43	$ -	Reject
14	2	110	$ -	notUnderstood
15	3	4	$ 1,452.00	Accept

The results show that there were some negotiations that ended because of misunderstanding the message. This result is due to the *Acknowledge* primitive that was left without relationship. We suggest that agent B acquires the usage of this primitive to avoid completely the communication problem.

We can conclude that experimentation has good results. Agents A and B are not really too different, their primitives are very similar, but if they were to negotiate without using our proposed solution, they would not be capable of understanding since the beginning of the negotiation process, and the results of all negotiations would be the message *notUnderstood*.

5. Conclusions

In this article we have presented a new methodology to integrate multiple negotiation agents in a Web based platform. In particular we have described an ontology-based solution to align and publish descriptions of negotiation primitives.

Our approach is based in the analysis of descriptions and pragmatics of negotiation primitives. The result of this analysis helps the developer to identify equality and similarity relationships between primitives. Once these relations have been established, the codification of primitives in the ontology is straightforward.

We presented the system architecture for executing negotiation processes using service-oriented technologies, improving interoperability between agents at run time, in contrast to most of the existing work on negotiation, which is based on distributed agent technology.

We believe that language interoperability between negotiation agents is an important issue that can be solved by incorporating a shared ontology. The experimental tests showed that the proposed methodology improves the communication between heterogeneous agents.

6. References

[1] T. Finning, R. Fritzon, and R. McEntire, "KQML as an agent communication language", *Proc. Of the 3rd International Conference on Information and Knowledge Management*, November 1994.

[2] FIPA – Foundation for Intelligent Physycal Agents. FIPA Specifications, 2003, available at http://www.fipa.org/specifications/index.html.

[3] J. Pérez, Maricela Bravo, Rodolfo Pazos, Gerardo Reyes, Juan Fraustro, "Design of a Shared Ontology Used for Translating Negotiation Primitives", *Proc. Of the International Conference on Computational Science and its Applications*, to appear, 2006.

[4] Müller, H. J., Negotiation Principles, *Foundations of Distributed Artificial Intelligence*, in G.M.P. O´Hare, and N.R. Jennings, New York: John Wiley & Sons.

[5] J. Gennari, M. Musen, R. Fergerson, W. Grosso, M. Crubézy, H. Eriksson, N. Noy, and S. Tu, "The evolution of Protégé-2000: An environment for knowledge-based systems development", *Proc. Of the International Journal of Human-Computer Studies*, 58(1): 89-123, 2003.

[6] H. Knublauch: An AI tool for the real world: Knowledge modeling with Protégé, *JavaWorld*, June 20, 2003.