# Loubeth: An interactive Intelligent Agent for Electronic Negotiation*

Osvaldo Cairó and Juan Gabriel Olarte

*Department of Computer Science*
*Instituto Tecnológico Autónomo de México*
*Río Hondo 1, Tizapán San Angel, 01000 México D.F., México*
*Tel: +52-5-628-4000, E-mail:{cairo, olarte}@itam.mx*

**Abstract.** Information systems with an *intelligent* or *knowledge* component are now prevalent and include knowledge-based systems, intelligent agents, and knowledge management systems. These systems are in principle capable of explaining their reasoning or justifying their behavior. This paper attempts to rectify this situation by drawing together the considerable body of work on the nature and use of explanations. Empirical studies, mainly with knowledge-based systems, are reviewed and linked to a sound theoretical base. In this paper we analyze: a) the process of negotiation between a user and an intelligent agent in a web store. The main purpose is to determine how the agent can utilize the user's information to select the best price for him, b) The formulation of the negotiation allows us to determine how the agent works using a protocol for the user's satisfaction, and c) The use of KQML language for an application to interact with an intelligent agent named Loubeth, who will communicate with the user on his behalf for the acquisition of CD music according to his preferences. The agent presented can find an optimal path to achieve its goal using its mental states and libraries designed for the business roles.

* This project has been funded by CONACYT as project number 33038-A, and Asociación Mexicana de Cultura, A. C.

## 1 Introduction

Automated negotiation is a key form of interaction in systems composed of autonomous agents [1]. Given its ubiquity, such negotiations exist in many different shapes and forms. However, we consider a particular class of automated negotiation: competitive negotiation. This negotiation appears over a single price between an agent and the user that both have firm deadlines. This is exemplified by the e-commerce scenario in which both a buyer and a seller agent negotiate over the price of a CD or service associated to the product. The buyer clearly prefers a low price, while the seller prefers a high one. In order to attempting to obtain the best price, agents also usually need to ensure that negotiation ends before a certain deadline. However, the end point may not be the only way in which time influences negotiation behavior. This may be the case, for instance, when one of the participants, say the buyer, is losing utility with time as a result of not getting the service. On the other hand, the seller may perhaps gain more utility by providing the service as late as possible. In short, it is clear that agent can have different attitudes toward time[2].

This paper presents the description of an agent that is capable of negotiating (proposing and asking to the offers) in B2B and B2C models. The project addresses the problem of making interactive systems in a context-web fashion. We first define the system we wish to create. This leads us to focus on a negotiation protocol and its operation. To support the web design, we define both an intelligent reactive agent and the architecture associated to its environment.

It is important to remark that we include animations in order to show an agent that represents an actual person. We prove that the idea of providing virtual person for showing to the consumers a person with the interaction is possible. We believe it should reduce the impact and gap created by the new technology. The final system improves the critical business processes with intelligent agents to make it robust.

The remainder of the paper is structured in the following manner. Section 2 include an introduction to the system in a formal description. Section 3 describes the basics of our negotiation model. Specification about agent and its behavior is shown in section 4. Section 5 presents the KQML language and the formulation of knowledge bases used by the agent within information obtained from the user. In section 6 we analyze the application architecture. Section 7 discusses related work. Finally, section 8 presents conclusions.

## 2 Definition of the system

In this section, we present an abstract formal model of Loubeth and the environment it occupy. Then, we use this model to describe the decision of design. The system is constructed in a web environment. This allows us to define the interaction between agent and the user who uses the system. We are taking terminology of sets to include all the elements involved and the logic based on standard notation[4].

Loubeth is perceived to be as a reactive agent, because it interacts with the environment by performing actions in response to some stimulus created by a user. We are assuming that the environment can establish a set of states where the agent can react. In this sense, we incorporate a set of states as $E = \{e_1, e_2, ...\}$ of possible states. In addition, Loubeth has a set of possible actions for the stimulus defined as $Ac = \{\alpha_1, \alpha_2, ...\}$.

The elements presented previously can be used to model the agent and the environment for the system. The model describes the states and the corresponding actions associated to each stimulus. Depending of interaction with the user, the agent choose an action that includes the collect of information that informs which option can result most favorable for the user in order to buy a product. The visit to the web store (r) can be resumed as a sequence of states in the following manner, as defined in [3]:

$$r : e_0 \xrightarrow{\alpha_0} e_1 \xrightarrow{\alpha_1} e_2 \xrightarrow{\alpha_2} e_3 \xrightarrow{\alpha_3} \cdots \xrightarrow{\alpha_{u-1}} e_u \cdots = z \,. \quad (1)$$

Formally, we say an environment *Env* is a triple $Env = \langle E, \tau, e_0 \rangle$ where *E* is a set of environment states, τ is a state transformer function, represented concisely, and $e_0 \in E$ is the initial state of the visit to the web store. Then, the agent is modeled as follows[6]:

$$Ag \;:\; R \;\rightarrow\; Ac \quad (2)$$

As a result of the previous definitions, the system is finally a set that evolves both the agent and the environment:

$$Sys = \langle Ag, Env \rangle \quad (3)$$

With the definitions presented in (1), (2), and (3), we can design a system of scenes for interaction between the agent and the user, determine the protocol of negotiation, and the roles for this negotiation. Thus, the agent will carry out tasks in behalf of us.

## 3 Negotiation Model

The bilateral negotiation model presented in this prototype works as many multilateral negotiations that can influence each other. A bilateral negotiation starts after the two parties - the buyer and the seller - match their objectives (i.e., they agree that what the buyer wants to buy is what the seller intends to sell), and consists on a sequence of proposals and counter-proposals. It ends when either one party accepts the other one's proposal or when it withdraws from the negotiation.

The negotiation can be observed like a process in which a joint decision is done by two organizations that have contradictory demands. The participants move towards an agreement by means of a process of concessions in search of new alternatives [7].

Currently, the proposed negotiation model is being used by two important firms in Mexico and now incorporate efficiency in electronic trading using intelligent systems. The protocol has been tested with trading scenarios and is available on the World Wide Web [18,19].

### 3.1 Definitions of negotiation's terms

The negotiation mechanism is based on a protocol and a strategy of negotiation. More formally, negotiation can be represented in the following terms:

- $P_{min}$ = Minimum price. It is the lowest price that a store is willing to sell a product and is obtained from the data base of the system.
- RP =Regular price. This is the price that is currently obtained in the market.
- Price of reserve ($P_{res}$). The price of reserve is defined as:

$$P_{res} = f(P_{\min}, FSD, LB, FS, COMM) \quad \textbf{(4)}$$

where

- FSD = Means the factor supply-demand of the product.
- LB = Expresses loyalty of the buyer. Qualification that is granted to the buyer according to his/her consumption in the site.
- FS = Shows the factor of season time of the year.
- COMM = Commissions decided with the store.

The price reserve (4) is the threshold of the agent to accept a supply. It is the lowest price that a product can be bought. It is calculated dynamically for each client and is influenced by factors of supply and demand, loyalty of the client, file of the article, etc.

- The initial price ( $P_{ini}$ ) is defined as:

$$P_{ini} = f(P_{\min}, RP, COMM, LB) \quad \textbf{(5)}$$

The price function in (5) can or it cannot be equal to the regular price of the product, depending on the loyalty of the consumer in the website.

- Round of negotiation. It corresponds to each of the cycles of negotiation, where the agent sends a proposal and the stores respond. It is defined as.

$$Round_i = (ask_i, bid_i) \quad \text{for} \quad i = 0, \ldots, i_{\max} \quad \textbf{(6)}$$

where
- $ask_i$ = Proposal of the agent in the round i.
- $bid_i$ = Proposal of the store in the round i.
- $i_{max}$ = Maximum number of stores.

- Supply or ask. It is the proposal that the agent of purchases makes to the user or consumer. It is defined as follows:

$$ask_i = \begin{cases} P_{ini} & i = 0 \\ f(bid_{i-1}, ask_{i-1}, HN, P_{res}) & i = 1, \ldots, i_{\max} \end{cases} \quad \textbf{(7)}$$

where
- $bid_{i-1}$ = Proposal of the consumer in the previous round.

- $ask_{i-1}$ = Proposal of the agent in the previous round.
- HN = Hardness of the negotiation.
- $i_{max}$ = Maximum number of rounds.

- Bid. It is the proposal that the user or consumer makes the agent of purchases.

The limits of this variable are the following ones:

$$0 < bid_i \leq ask_i \quad \text{if} \quad i = 0$$
$$bid_{i-1} < bid_i \leq ask_i \quad \text{if} \quad i = 1,\ldots,i_{\max} \qquad \textbf{(8)}$$

### 3.2 The protocol's operation

The entities that participate in the negotiation's model have limited knowledge of the preferences and restrictions of the opponent. That is why the rank in possible agreements is represented by the intersection of the individual areas with particular interests that are typically not known a priori. For this reason, it is necessary to find an agreement that benefits to both parts. The entities must move and explore possible agreements by means of the exchange of information (in form of supplies).

The negotiation's model works as follows: each one of the entities begins the negotiation offering its preferred solution from its area of interest. If the supply is not acceptable for the other entity, this one makes bids so that are more near reaching an agreement. During the process the ranks of possible options for each entity are reduced until it is possible to be reached an agreement or some of the entity retires of the negotiation. In other words, if the area of common interest is reduced to a solution, then one reaches an agreement and the negotiation is successful. Otherwise it is not possible to reach an agreement.

The process of satisfaction of restrictions will be used by the agent of negotiation, being the primary target that the agent sells a product, instead of maximizing the gains.

### 3.3 Possible scenarios of the negotiation

The actions taken by the agent at a certain moment are influenced by the prices that each store offers and by the preferences of users. The main movements that the agent is able to make are:

1. To finish the negotiation - accepting or finishing the purchase -,
2. To generate a supply to the user (ask),
3. To show new supplies that is attractive to the buyer or any of the following messages:

- Take it or leave it. This message indicates the user that it is the only available thing and suggests the purchase.

- This it is the lowest price. With this option the agent indicates the user that the comparison of the product was the optimal one.

## 4 Agent's specification

The agent proposed in this prototype is reactive, since its actions depend on the interaction with the buyer. The figure 1 shows the architecture that describes how Loubeth communicate with the environment by the stimulus received from it and the corresponding actions that relates to that stimulus.

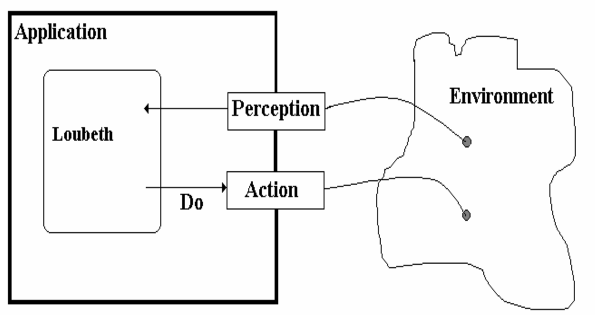


**Fig. 1.** Agent's architecture

The information that the agent contains must relate to the profile of the buyer and its necessities. Loubeth is responsible of informing about the products for sale and the terms of negotiation in a commercial transaction. One of the principal objectives of the agent is to coordinate the information related to the electronic store. The performance of this task includes the information about the buyers and the discs that are demanded and offered.

### 4.1 Loubeth's mental models

The importance of the agent-mediated systems is the believability of the agent involved with the user's goals because it should behave as a real human. This behavior should include capabilities such as: emotion, personality, perception, language understanding and many others. In order to perform these activities, the agent has beliefs, capabilities, and commitments.

- **Beliefs.** Represent the current state of the agent's internal and external world and are updated as new information about the world is received

- **Capabilities.** Is a construct used by the agent to associate an action with that action's necessary preconditions.
- **Commitments.** A commitment is an agreement, usually communicated to another agent, to perform a particular action at a particular time.

### 4.2 Agent's behavioral rules

In Shoham's model[8], all actions were performed only as the result of commitments. Reticular has extended the idea of a commitment rule to a include a general behavioral rule[9]. Behavioral rules determine the course of action an agent takes at every point throughout the agent's execution.

Behavioral rules match the set of possible responses against the current environment as described by the agent's current beliefs. If a rule's conditions are satisfied by the environment, then the rule is applicable and the actions it specifies are performed. Behavioral rules can be viewed as WHEN-IF-THEN statements. The WHEN portion of the rule addresses new events occurring in the agent's environment and includes new messages received from other agents. The IF portion compares the current mental model with the conditions that are required for the rule to be applicable. Patterns in the IF portion match against beliefs, commitments, capabilities, and intentions. The THEN portion defines the agent's actions and mental changes performed in response to the current event, mental model, and external environment. These may include:

a) Mental model update
b) Communicative actions
c) Private actions

### 4.3 Loubeth's gestures and interaction

It is important to note that we incorporate animations in order to represent an agent as a virtual person. Providing a virtual human with humanlike reactions and decision capabilities is complicated. Simulated actions and decisions are used to convince the viewer of Loubeth's skills and intelligence in negotiation. This level of performance entails the possibility of proving the idea of incorporating virtual person for showing to the consumers a person with the interaction. We believe this should reduce the impact and gap created by the new technology. In our work we create such representations by incorporating natural behaviors with the purpose of obtaining a robust prototype that improves intelligent systems for electronic commerce.

The prototype presented illustrates the importance of social reactivity by considering some responses from the agent to users who want to purchase CD music from an electronic store. In this case each reaction from the agent corresponds to a emotion suggesting signals that must accompany animated gifs and recorded audio, providing a more natural feel and a greater communication [10].

When Loubeth is reacting to obtain results or reporting difficult, the system provide some of the following items:

1. I was unable to find your Compact Disc with that description (sad).

2. Hey, I think this is a CD for you! (proud).
3. Well, I'm not sure, but I think this Compact Disk is closest to your requirement (hesitant).
4. Wait a moment, it only took me 10 seconds to find it! (embarrassed).
5. I'm afraid I don't understand what you mean (confused).
6. I though I just told you that I don't know anything about this topic (irritated).

The figure 2 shows the different ways the agent can interact with the user in a graphical manner depending on the user's actions.

The interaction between Loubeth and the user is explained by means of a visual interface. The use of a knowledge base allows to know the different answers of the agent by means of messages from audio. The figure 3 shows how we can relate a gesture of the agent with several options of messages that are stored. The combinations depend on the number of images and the messages that the knowledge base contains. Although the project is made using animated gifs and voice recorded, it can be easily replaced by a set of videos presenting the same expressions.

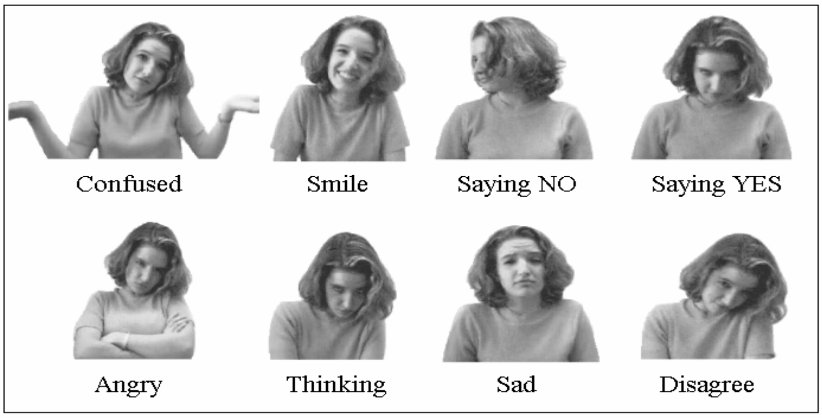


**Fig. 2.** Different reactions of the agent

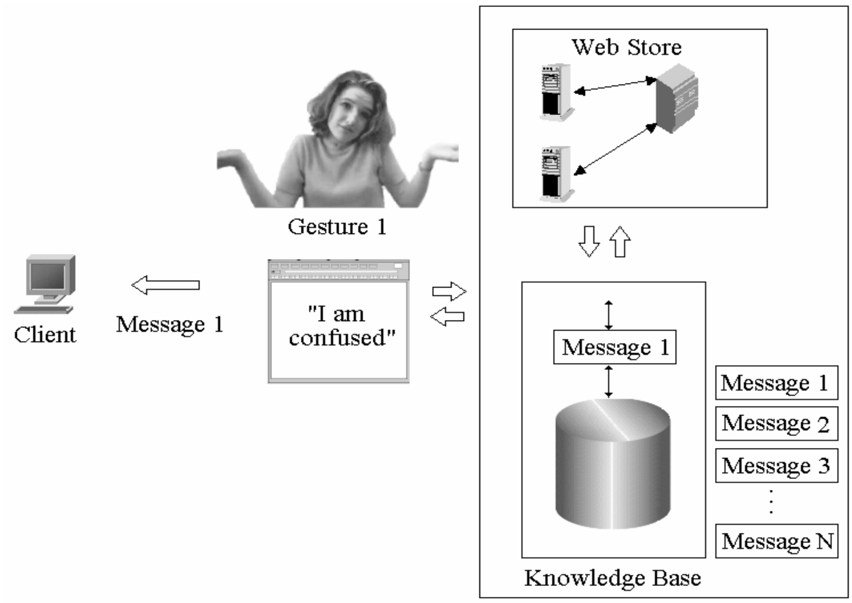


**Fig. 3.** Knowledge base representing several forms to show messages for users

In order to exhibit the different gestures from the agent, we have a series of answers associated a group of gestures. Later, each gesture has a small list that contains variations of the same gesture. Figure 4 exhibits Loubeth's behaviors for different gestures.

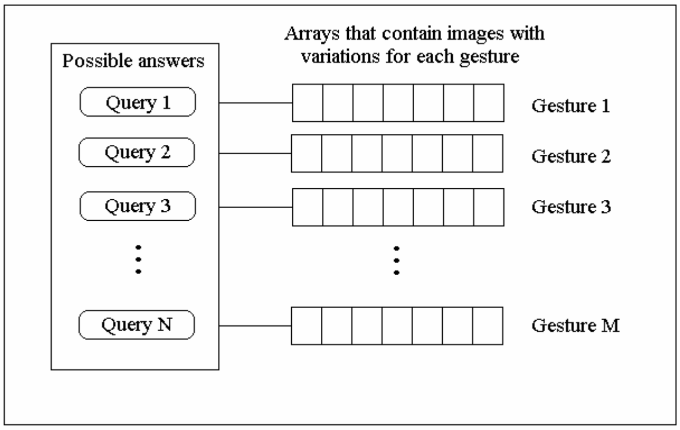


**Fig. 4.** Loubeth's gestures planning design

## 5 KQML Language

The Knowledge Query and Manipulation Language (KQML) is a high-level language intended to support interoperability among intelligent agents in distributed applications. It is composed by  a message format and a message-handling protocol to support run-time knowledge-sharing among agents. It is an interlingua, a language that allows an application program to interact with an intelligent system. It can also be used for sharing knowledge among multiple intelligent systems engaged in cooperative problem solving. This language, originally developed as part of a DARPA Knowledge Sharing initiative, is becoming a de facto standard for interagent communications languages [11,12,13,14,15].

### 5.1 KQML Semantics

The semantic model underlying KQML is a simple and uniform context for agents to view each others' capabilities. Each agent appears as if manages a knowledge base. That is, communication with the agent is with regard to this knowledge base. For example, there are questions about what a KB contains, statements about its content, requests to add or delete statements from it, or requests to use knowledge in order to route messages to appropriate other agents.

The implementation of an agent is not necessarily structured as a knowledge base. The implementation may use a simpler database system or a program using a special data structure as long as wrapper code translates that representation into a knowledge-based abstraction for the benefit of other agents. Thus, we say that each agent manages a virtual knowledge base (VKB).

The English-prose performatives make reference to these terms, but this view of the VKB is especially important in the formal semantics of KQML [13].

### 5.2 Interaction with knowledge bases

The figure 5 is an example of the interaction among Loubeth, three stocks, Data Base, and the Knowledge Base created for the negotiations.

We can observe that three stocks participate in the selection of the lowest price that Loubeth obtains with the help of KB. During the negotiation, the KB is evaluating the properties associated to each product. The main characteristics are used to discriminate certain properties, which may cause a non-desirable results for the client. The negotiaion follows rule-based strategies according to the business goals.

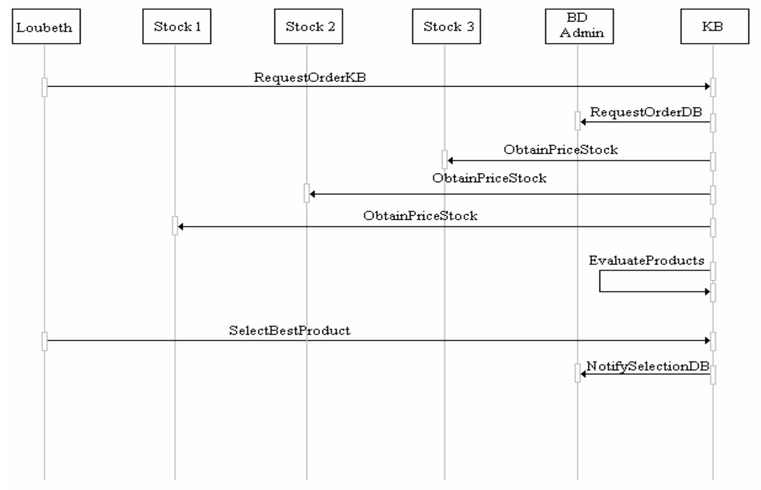


**Fig. 5.** Negotiation Interaction Diagram

### 5.3 Knowledge declarations

To determine proposals related to a certain product, all the information about products must be compared according to its properties. This can be considered by using one Knowledge Base, defined for attributes and the product abstraction. The specification for matching and selection are presented.

**WHEN** KB find match
  **IF** product_of_interest (Loubeth:Agent, P:properties, S: Selection ) and
    in_selection(P:properties, ST: Stock)
  **THEN**
    possible_purchase(S:Selection)

The knowledge base specifies the fact that a product can be a good candidate for purchasing, and is attracted by the product selected. Although this KB can determine possible proposal, it is important to assume other knowledge base to complete decisions based on the user's perspectives and product management using sharing knowledge[16].

In the selection of products, the knowledge base evaluate the products obtained from the stocks that conform the agent's source information.

**WHEN** KB selection evaluate
  **IF** possible_purchase(S:Selection) and
    product_acceptable(P:properties, Loubeth:Agent)
  **THEN**
    Good_choice_purchase(Loubeth:Agent, P:properties, S: Selection)

## 6 Application architecture

The prototype presented is a web-centric application that uses a web client to send requests and receive results from a web application. A web application is a bundle of web components and their supporting classes, beans, and files. Web components are server-side J2EE components, such as Servlets and JSP pages[17]. The application consists of a single web module. A web module is the smallest deployable and usable unit of web resources in a J2EE application. A feature introduced in this application is the web module construct, which automatically creates the required directory structures, default versions of required data objects, and other special services required by the web module.

The negotiation model and the architecture in which the agent is embedded is shown in figure 6.

For purposes of efficiency and modularity, the architecture consists of separate componets running in parallel as separate processes. The components communicate by exchanging messages and include the following features:

-Business component.
-Agent component.
-Commerce component.

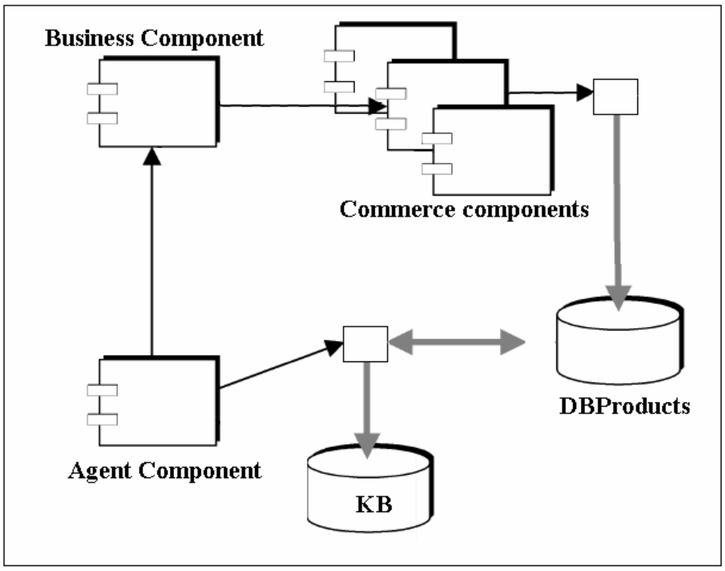


**Fig. 6.** Application's architecture and the relationship among components

## 7 Conclusions

This paper determined what the optimal negotiation strategies are for agents that find themselves in web environments with different information states about users who buy products. Specifically, we considered situations where agents act as assistants help the user to make good decisions in the acquisition of goods in a web store.

We listed conditions for convergence of these optimal strategies and studied the effect of giving best alternatives of buying on the negotiation outcome. In the future we intend to extend our analysis to determine if this strategic behavior leads to maximize gains and then analyze situations where agents have limited information about other negotiation parameters like the store's bargaining cost, its discounting factor or its strategy to compare their relative influences on the negotiation outcome.

The use of KQML assumes a model of an agent as a knowledge-based system (KBS). The KBS model easily a broad range of commonly used information agent models, including database management systems, hypertext systems, server-oriented software, simulations, and more. The contribution that KQML makes to this prototype is to offer a communication language as well as the information related to web applications

prototype presented is a web-centric application that uses a web client to send requests and receive results from a web application.

## References


1. P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *International Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.
2. S. S. Fatima, M. Wooldridge, and N. R. Jennings. Optimal Negotiation Strategies for Agents with Incomplete Information. In J.-J. Meyer and M. Tambe, editors, Intelligent Agents VIII Springer-Verlag Lecture Notes in AI Volume, March 2002.
3. M. Wooldridge and P. E. Dunne. Optimistic and Disjunctive Agent Design Problems. In Y. Lesperanceand C. Castelfranchi, editors, Intelligent Agents VII Springer-Verlag Lecture Notes in AI Volume, February 2001.
4. C. H. Papadimitriou. Computational Complexity. Addison-Wesley: Reading, MA, 1994.
5. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. Reasoning About Knowledge. The MIT Press: Cambridge, MA, 1995.
6. S. Russell and D. Subramanian. Probably bounded-optimal agents. Journal of AI Research, 2:575–609, 1995.
7. R. Kowalczyk and V. Bui (2001) On Constraint-Based Reasoning in e-Negotiation Agents. Agent- Mediated Electronic Commerce III, Lecture Notes in Artificial Intelligence. Vol. 2003: 1-14.
8. Y. Shoham. (1991). AGENT-0: A simple agent language and its interpreter. In proceedings of the Ninth National Conference on Artificial Intelligence, Vol. II. (pp. 704-709). Anaheim, CA:MIT Press.
9. Reticular Systems. An integrated Toolkit for Constructing Intelligent Software Agents. Version 1.3 Rev 0. April 30, 2000.

10. J. Casell, J. Sullivan, S. Prevost, and E. Churchill. Embodied Conversational Agents. MIT Press, 2000.
11. T. Finin, R. Fritzon, D. McKay, and R. McEntire. (1994/1994a). KQML as an agent communication language. In Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94), ACM Press.
12 T. Finin, R. Fritzon, D. McKay, and R. McEntire. (1994b). KQML - A Language and Protocol for Knowledge and Information Exchange (Technical Report No. CS-94-02). University of Maryland, Department of Computer Science.
13 T. Finin, J. Weber, G. Wiederhold, M. Genesereth, R. Fritzon, J. McGuire, S. Shapiro, D. McKay, R. Pelavin, and C. Beck. (1994c). Specification of the KQML Agent-Communication Language plus example agent policies and architectures (DRAFT Report), DARPA Knowledge Sharing Initiative External Interface Working Group.
14 Y. Labrou and T. Finin, (1994). "A semantics approach for KQML - a general purpose communication language for software agents," University of Maryland.
15 Y. Labrou. (1996). "Semantics for an agent communication language," in Computer Science and Electrical Engineering Department. Baltimore, MD: University of Maryland Graduate School, pp. 116.
16. M. Vetter and S. Pitsch. Towards a Flexible Trading Process over the Internet. Agent mediated Electronic Commerce. The European AgentLink Perspective. Springer. 2001.
17. http://java.sun.com.
18. http://www.compaq.com.mx.
19. http://www.elektra.com.mx.