

A Shopping Model in Agent-mediated Electronic Commerce

Ziming Zeng & Liyi Zhang

School of Information Management, Wuhan University, Wuhan 430072, China

E-mail: zmzeng1977@yahoo.com.cn

Abstract

As the development of Internet, electronic commerce technology in which agent takes the role of medium is becoming more and more popular recently. It accentuates how to make use of agent technology to enhance the automation and efficiency of shopping in the Internet and businessman's selling process. However, today's first generation shopping agent systems are limited to comparing product usually on price instead of their full range of attribute and don't reflect the customer's actual purchase preference. For these limitations, we propose a new shopping model for electronic commerce based on the agent technology and elaborate how to analyze and design the agent system. Based on the agent technology, the combination of a multi-attribute evaluation model and preference evaluation model is presented. Finally, an experimental prototype system for car shopping on IBM's Aglets is developed for demonstrating the proposed shopping model above.

Keywords: Agent, Electronic commerce, Multi-attribute evaluation, Preference evaluation

1. Introduction

There are more and more electronic commerce websites with the development of the Internet and it is not easy for the customers to select the most desirable products. Therefore, it is inevitable for us to compare the shopping pattern and develop the shopping websites. Through a shopping website, a customer can search for his or her interesting products and select the suitable one in the Internet.

Agent-based systems technology have provided lots of exciting applications for electronic commerce in recent years (Guttman R.H., 1998, pp. 147-159). because of its promise as a new paradigm for creating software that operates in environments that are distributed and open, such as the Internet Currently. So the researches of electronic commerce based on agent are expanding for the intelligence and automation of shopping. The purpose of agent technology in the shopping websites is to stand for buyers to search for suitable products and bargain with the suppliers.

The main purpose of shopping agent systems is to help customers to have information retrieval and information comparison in the massive information environment of the Internet. It is responsible to search for massive information on behalf of customers and select the preference information that is fit for the demands of the customers.

The paper is organized as follows. In section 2, we first give an overview of related work. Then in section 3, we analyze the intelligent shopping system and propose a solution about it. Following that, in section 4 we implement the shopping model based on multi-agent and in section 5, we develop an experiment prototype System on IBM's Aglets. Finally, we present the conclusions in section 6.

2. Related Work

Agents are intelligent software entities that possess the properties of autonomy, social ability, reactivity, pro-activeness and even mobility that mean they are able to migrate across different executions in the Internet (Wooldridge and Jennings, 1994). Recently, agent technology has been applied to various applications of electronic commerce (EC), such as information searching data filtering, electronic marketplace, bargaining and etc., to reduce the cost of transaction and promote enormous economic benefits.

The SICS MarketSpace (Eriksson, 1998, pp. 41-53) prototype was developed in Intelligent Systems Lab, Swedish Institute of Computer Science using Java. And it consists of a personal assistant allowing the user to describe his or her interests. Two agents augmented web shops. It provides a directory service that allows user and service agents to register interests and find agents with matching interests.

AuctionBot (Wurman, 1998, pp. 301-308) is general purpose internet auction server at University of Michigan where users create new auctions to buy products by choosing from a selection of auction types and specifying its parameters. And a buyer can bid with a seller according to the bilateral distributive negotiation protocols of the created auction. In a typical scenario, a seller would bid a reservation price after creating an auction and let AuctionBot manage and enforce a buyer bidding according to the auction protocols and parameters.

Magnet (Steinmetz, 1998, pp. 105-125) was developed by University of Minnesota. The goal of the Magnet project was to develop a semantic model for the integration of planning, contracting, scheduling, and execution in an agent-mediated market domain.

However, these shopping agent systems are limited to comparing products usually on price instead of their full range of attributes. Since the criteria for evaluating products has several attributes, Multi-attribute decision-making is required for the intelligent shopping systems.

3. The Analysis of Intelligent Shopping System

3.1 Relevant Problem

The main purpose in the paper is to design an intelligent shopping system and help a customer to search for his or her interesting products in the Internet. In the system, agents will search for the massive shopping information in real time and select the suitable product automatically based on the customer's preference on behalf of the customer.

The problems that intelligent shopping system will solve include:

- (1) How to search for the suitable products automatically in the Internet; and build a decision-making model to calculate the evaluation result of each product.
- (2) How to improve the candidate product and negotiate with the suppliers until the customer obtains the desirable product.
- (3) How to study and imitate the customer's individual purchase preference automatically from the previous shopping records of the customer.

3.2 Solution

In order to solve the problem above, the combination of a multi-attribute evaluation model and preference evaluation model is presented in this paper (Wang, 1986). The shopping process is listed as follows and its workflow is shown in figure 1.

- (1) First, intelligent agent system performs searching task, and it can be implemented in detail by a group of Purchase Agent that search for the quotations of products from the suppliers distributed in the Internet.
- (2) After receiving the quotations of products from the suppliers, intelligent agent system computes the utility value of each product based on the multi-attribute decision-making model and selects the product that has max utility value as the candidate product.
- (3) Then intelligent agent system submits the candidate product to the customer. If the customer is satisfactory to the candidate product, go to step (6).
- (4) If the customer is still not satisfactory to the candidate product, the intelligent agent system will acquire the buyer's individual preference storing in the preference knowledge database at once and perform the preference analyzing.
- (5) The intelligent agent system asks for a further quotation of the candidate product from its supplier and improves the quotation, then goes to step (3).
- (6) The intelligent agent system stands for the customer to carry out the electronic transaction, and the process of shopping is completed.

4. Implementation Model of the Shopping Based on Multi-agent

In the intelligent agent system, the concept of horizontal hierarchy is introduced in the paper. It means that the whole system consists of the following agents interacting with each other. They are Interface Agent, Search Agent, Purchase Agent, Evaluation Agent and Preference Agent. These agents cooperate with each other and make the whole system works together. The structure of system is shown in Figure 2.

This partition method makes each agent module relatively independent, promotes the reuse of each agent component, and each agent can execute simultaneously, so the whole performance of the system can be increased. For example, when Search Agent receives the quotation of a new product sent from Purchase Agent, Evaluation Agent can evaluate the former sent products at the same time, while Interface Agent is responding the request of a customer. Therefore, the system can perform efficiently.

In fact, multi-agent system originates from the development of the distributed artificial intelligence. It is the great advantage of multi-agent systems that cooperate with each other and solve the problems in the parallel computing pattern. The detailed functions of each agent in the system are described as follows.

4.1 Interface Agent

Interface Agent provides a friendly operating interface to the customers. It is responsible to exchange the information with the customers: on the one hand, it receives the purchase requirements of the customers and sends the requirements to the other agents in the system to proceed, on the other hand, the purchase result that other agents submit is displayed to the customers through the Interface Agent. So Interface Agent is the interface between the customers and the intelligent shopping Agent system.

4.2 Search Agent

Search Agent responds the purchase requirements of the customers and it is responsible for the whole searching task of the products. In order to promote the efficiency of searching, Search Agent creates a group of Purchase Agent and dispatches each Purchase Agent to search for a supplier side in the Internet. These Purchase Agents perform parallel searching, so Search Agent must manage the running state of all Purchase Agents and coordinate task distribution among them. The process of Search Agent is described in Figure 3.

4.3 Purchase Agent

When Purchase Agents is created, they will be dispatched to search for the suppliers distributed in the Internet. After they find respective supplier, each Purchase Agent negotiates with the supplier and asks for a quotation about the product from supplier. After Purchase Agents get the quotations, they return the searching results to Evaluation Agent. The process of Purchase Agent is described in the Figure 4.

4.4 Evaluation Agent

Evaluation Agent is the indispensable part of the whole intelligent system. After receiving the quotations of all products from the suppliers, Evaluation Agent will decide which product is the best among all the submitted products. Since the criteria for comparing quotation of products has several attributes such as price, quality, reliability, appearance and supplier credit etc, multi-attribute decision-making method is required for computing the utility value of each product. Its mathematical model can be described:

Considering $X = \{x_1, x_2, \dots, x_m\}$ as the product set searched in the Internet, $Y = \{y_1, y_2, \dots, y_n\}$ as the attribute set, the utility value of product $X_i (1 \leq i \leq m)$ about the attribute $Y_j (1 \leq j \leq n)$ can be expressed as $f_{ij} = f_j(x_{ij})$. Therefore, the evaluation matrix which consists of $m \times n$ f_{ij} can be expressed as:

$$F = \begin{bmatrix} f_{11} & f_{12} & \dots & f_{1n} \\ f_{21} & f_{22} & \dots & f_{2n} \\ \dots & \dots & \dots & \dots \\ f_{m1} & f_{m2} & \dots & f_{mn} \end{bmatrix} = (f_{ij})_{m \times n} \quad (1)$$

For easy mutual reference to the attributes that have different dimensions, the evaluation matrix should be normalized and can be followed by:

$$f'_{ij} = \frac{f_{ij}}{\sqrt{\sum_{i=1}^m (f_{ij})^2}} \quad (2)$$

After normalizing the evaluation matrix, $F' = (f'_{ij})_{m \times n}$ is figured out and f'_{ij} is limited in $[0,1]$. Evaluation Agent calculates the utility value of each product $x_i \in X = (x_1, x_2, \dots, x_m)$ based on the equation (3):

$$U(x_i) = \sum_{j=1}^n \omega_j f'_{ij} \quad (3)$$

In the equation (3), $U(x_i)$ is the utility value of the product $x_i (1 \leq i \leq m)$; $\omega_j \in [0,1]$ is the weight of the attribute $j (1 \leq j \leq n)$, and $\sum_{j=1}^n \omega_j = 1$.

After calculating the utility value of all the products, Evaluation Agent will select the product that has the max utility

value as the candidate product. The candidate product can be expressed as: $A^* = \{x_i \mid \max_{1 \leq i \leq m} [U(x_i)]\}$. Finally, Evaluation Agent submits the candidate product to the customer through Interface Agent. All the computing processes are carried out by Evaluation Agent automatically. The process of Evaluation Agent is described in Figure 5.

4.5 Preference Agent

If the customer is still not satisfactory to the candidate product, Preference Agent will acquire the customer's individual preference from the preference knowledge database at once and perform the preference analyzing. Because the inference and analyze of Preference Agent is based on the rules, relevant rules must be stored in the preference knowledge database. Here we illustrate some rules as examples:

R_1 : If (attribute y_i is more preferential than y_j) when a buyer asking for a new quotation

Then (y_i is improved first).

R_2 : If (reference quotation is X_1) and (the concession of supplier is x)

Then (concession of buyer is y)

R_3 : If (the preference is determined) AND (the quotation is modified)

Then If (the buyer and supplier are both satisfied with the quotation)

Then negotiation is terminated with success

Else asking for a further quotation.

The rule R_1 indicates the rule of attribute preference of the customer when he or she modifies the quotation of the candidate product; R_2 is the rule about the negotiation strategy: a new quotation must be superior to the reference quotation and when the concession of supplier is x , the concession of the buyer can be y accordingly; R_3 is the rule about negotiation control.

Preference Agent will modify the quotation of the candidate product according to the customer's individual preference and obtain a more satisfactory quotation for the customer. In the whole process of the shopping, Preference Agent will study and imitate the customer's individual preference automatically from the purchasing behavior of the customer and store it in the preference knowledge database after the shopping process is completed with success.

5. The Experiment about the System

We utilize IBM's Aglet to develop a website for car shopping. IBM's Aglet is a Java-based tool for developing mobile agents (Aglets, 2002). It includes the Aglet API, documentation, sample aglets, and Tahiti aglet server. The Aglet API, created and operated aglets, is a Java package consisting of classes and interfaces. The basic functionality and runtime properties of aglets are Aglet, AgletProxy and AgletContext classes. The Aglet class defines the fundamental methods to control the mobility and life cycle of an aglet. The AgletContext class provides the runtime environment for aglets with the Tahiti server. The AgletProxy class provides a common way of the aglet. It provides location transparency and acts as a shield object that protects an aglet from direct access by other malicious aglet. All communication with other aglets must occur through this interface.

In the experimental work, we create four Aglet servers. One stands for the agent server of the shopping website in order to simulate the processes of electronic shopping. The other three stands for the suppliers. Figure 6 illustrates the experimental architecture. In Figure 6, Server A is the web server of shopping website, the suppliers are located at server B, server C, and server D. Server A, server B and server C are located on the same local-area network, while Server D is located in the different remote network.

In the shopping Server A, a customer can set the demand of purchasing, including price, brand, quality, reliability and supplier credit etc. The agent system of the shopping website will search for the suitable suppliers according to the demands. After receiving the quotations of different cars from the suppliers, the agent system will utilize multi-attribute evaluation model to select the most desirable car for the customer. Then, the agent system is able to study and simulate the customer's preference automatically and modify the quotation of the candidate product until the quotation is satisfactory to the customer.

In the intelligent agent system, Purchase Agent is a mobile aglet, while other agents are stationary aglets including Interface Agent, Search Agent, Evaluation Agent and Preference Agent. In the experiment, the round-trip time associated with Purchase Agents traversing the Internet between Aglet servers is measured. When a Purchase Agent is created and dispatched on a round-trip between Shopping Server A and some Supplier Server (either server B,

server C or Server D), it registers timestamps corresponding to its departure from Server A and to its subsequent return. The average round trip time t_{avg} can be computed as

$$t_{avg} = \left(\sum_{i=1}^n t(a)_i - t(d)_i \right) / n \quad (4)$$

In the equation (4), i is for the i th trial run, $t(a)_i$ is the timestamp registered by Purchase Agent before departing from Shopping Server A; $t(d)_i$ is the timestamp computed by Purchase Agent immediately upon returning to Shopping Server A; n is the number of trial runs made by Purchase Agent between the same pair of servers. In the experiment, the number of trial runs is $n = 100$. Search Agent creates three Purchase Agents and dispatches each Purchase Agent to search for its respective supplier in the Internet simultaneously.

As expected, t_{avg} for Supplier Server B is least (123 ms) on the same local-area network and t_{avg} for Supplier Server D is most (864 ms) because D is located in the different remote network from Shopping Server A.

6. Conclusion

To enhance the automation of shopping in the Internet and help the customers to purchase the desirable products from the websites quickly, a shopping model for electronic commerce based on agent technology is designed and implemented in this paper. In the intelligent agent system, we have utilized the multi-attribute evaluation model and preference evaluation model to select the most suitable product for the customers. And we also develop a car shopping website on Aglet platform as an application example for the proposed intelligent agent system. The experimental results show that the system performed efficiently and effectively. This paper has done some available attempts in the intelligent electronic commerce fields. Future research of this work will include how to study and imitate the customer's individual preference more actively and introducing the negotiation mechanism in Purchase Agent.

References

- (Steinmetz, 1998, pp. 105-125) Steinmetz E. (1998). Bid evaluation and Selection in the MAGNET Automated Contracting System. *Lecture Note in Artificial Intelligence 1571, Agent Mediated Electronic Commerce*. 105-125.
- Aglets. (2002). [Online] Available: http://www.trl.ibm.com/aglets/index_e.htm, IBM, (June 2002)
- Eriksson J. (1998). SICS MarketSpace – An Agent-Based Market Infrastructure. *Lecture Note in Artificial Intelligence 1571, Agent Mediated Electronic Commerce*. 41-53.
- Guttman R.H. (1998). Agent-mediated electronic commerce: a survey. *Knowledge Engineering Review*. 13 (2): 147-159.
- Wang, ShouBong. (1986). *Social Choice and Multicriterion Decision Making*. Boston: MIT Press.
- Wooldridge M, and Jennings N.R. (1994). Agent Theories, Architectures, and Languages: A Survey. *Proceedings of ECAI-94 Workshop on Agent Theories, Architectures, and Languages*. Amsterdam, The Netherlands.
- Wurman P.R. (1998). The Michigan AuctionBot: A Configurable Auction SERVER for Human and Software Agents. *Proceedings of the Second International Conference on Autonomous Agents (Agents98)*, Minneapolis, MN. 301-308.

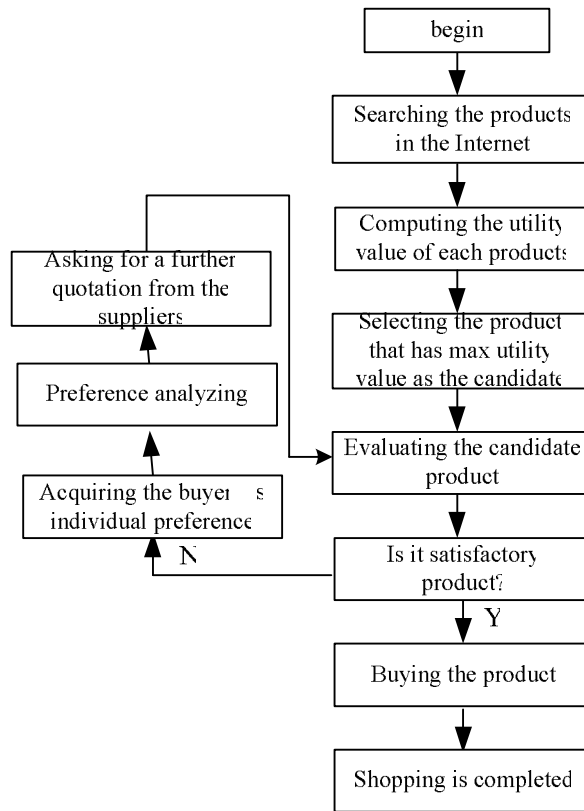


Figure 1. the combination model of a multi-attribute evaluation and preference evaluation

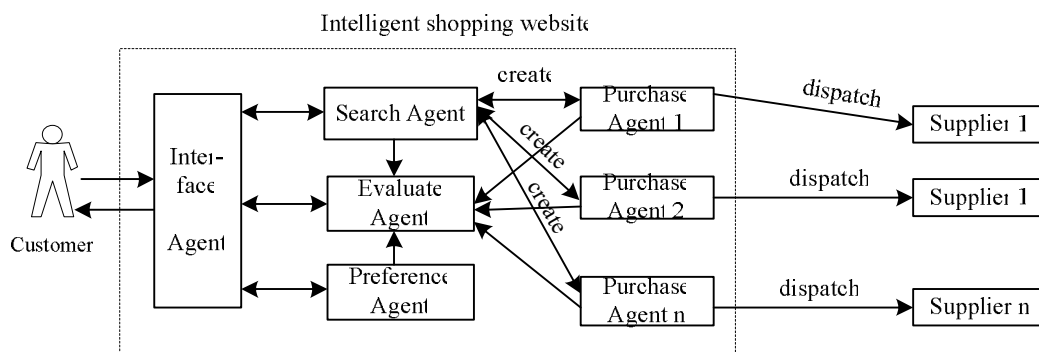


Figure 2. The structure of intelligent agent system

```

//class for Search Agent
import com.ibm.aglet.*;
public class Search Agent extends Agelet {
    public void onCreate(Object init){
        create the operative window;
    }
    public void createPurchaseAgent(){
        create a group of Purchase Agents;
    }
    public void dispatchPurchaseAgent(){
        dispatch Purchase Agents to find the suppliers;
        send the proxy of all Purchase Agents to each
        dispatched Purchase Agent;
    }
    public void sendCommand(String command){
        send the command message to each Purchase Agent;
    }
    public Boolean handleMessage(Message msg){
        if Purchase Agents send the current status
            record and show the status;
        if Purchase Agents send supplier's quotations of cars
            record and show the quotation;
    }
}

```

Figure 3. Implementation for Search Agent

```

//Class for Purchase Agent
import com.ibm.aglet.*;
public class PurchaseAgent extends Agle {
    public void onCreator(Object init){
        extract the information of supplier product
        from Search Agent
    }
    public boolean onArrivalC {
        obtain the proxy of supplier agent
        modify the current locator
    }
    public void runC {
        if current location is at the respective side {
            ask for contact
            if the supplier permit contact {
                ask for a quotation about the product
                get the quotation
                send the searching result to Evaluation Agent
            }
            send the current status to Search Agen
            dispose oneself
        }
    }
    public Boolean handleMessage(Message msg){
        if Search Agent send the Suspend command
            suspend oneself
        if Search Agent send the Resume command
            resume oneself
        if Search Agent send the Stop command
            dispose oneself
    }
}

```

Figure 4. Implementation for Purchase Agent

```

//Class for the EvaluationAgent
import com.ibm.aglet.*
public class EvaluationAgent extends Aglet {
    public void onCreate(object ini) {
        initialize oneself using multi-attribute decision model
        create the operative window
    }
    public Boolean handleMessage(Message msg) {
        if PurchaseAgents send supplier's quotations about products
            record and show the quotations
    }
    public void evaluation() {
        if all the PurchaseAgents send quotations {
            compute utility value of each product
            select the product that has max utility value as the
            candidate
        }
    }
    public void sendMessage(Message msg){
        send the quotation of the candidate car to InterfaceAgent
    }
}

```

Figure 5. Implementation for Evaluation Agent

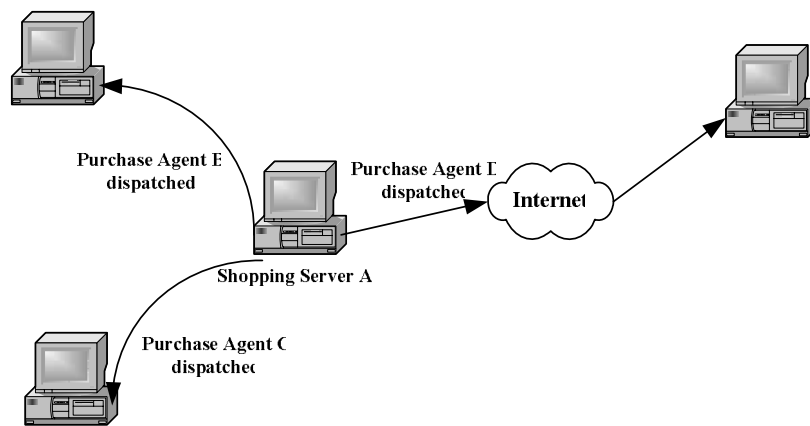


Figure 6. The experimental architecture of this system