An Integrated Analysis Strategy and Mobile Agent Framework for Recommendation System in EC over Internet

Chuan-Feng Chiu[1], Timothy K. Shih[2] and Ying-Hong Wang

Department of Computer Science and Information Engineering
Tamkang University
Tamsui, Taiwan 251, R.O.C.
E-mail : [1]cfchiu@cs.tku.edu.tw
[2]tshih@cs.tku.edu.tw

Abstract

Internet has become a popular medium for information exchange and knowledge delivery. Many people get the useful information that they wanted from the Internet and network. Several traditional social activities have changed and evaluated to Internet, like distance learning and tele-medical system. Traditional buying and selling activities also follow the trend. Almost all things will be sold in the Internet and user will buy the product from the Internet too. However with the advent of the World Wide Web, online merchant must know what users wanted or user's interests and let users to buy something in their site. So recommendation process became an important strategy for the merchants. In this paper we analyze users' behavior and their interests, and then we recommend something to these users. The analysis mechanism is based on the correlations among customer, product items, and product features. In this paper we propose an algorithm to classify users into groups and recommend product items based on these classified groups. And the system will help merchants to make suitable business decision and push personal information to customers. In the other hand we also propose a generic mobile agent framework for electronic commerce applications and collaborative agent computing architecture for the recommendation system based on the framework.

Key Words: Electronic Commerce, Recommendation System, Internet, World Wide Web, Mobile Agent

1. Introduction

Because of the popularization of the Internet, many traditional activities have changed. In recent years electronic commerce has become a popular trend on the Internet. No spatial and temporal constraints are the advantages of the Internet. These benefits make the business process more flexible. The applications in electronic commerce environment can be divided into categories that include business-to-business and business-to-customer applications. Business-to-business applications focus on the enterprise activities. That includes enterprise information management, information exchange, and supply chain management, etc. On the other hand, business-to-customer applications focus on the relation of merchants and consumers. In [1] they proposed the consumer buying behavior model, which describes the fundamental stages of buying behavior. The stages include need identification, product brokering, merchant brokering, negotiation, purchase and delivery, and product services and evaluation. These functions are so complex and huge that we cannot conclude all processes. In this paper we focus on the business-to-customer application and the product and merchant brokering stage in the consumer buying behavior model.

In traditional business behavior we get recommendation or suggestions from other people. But in the Internet environment customers buy things without assistance or suggestions. So the recommendation system becomes an important application area and an academic research topic. The recommendation process can be divided into on-line processes and off-line processes. At the on-line stage we discover the customer's behavior and transform the information to useful data for analysis. At the off-line stage we analyze the collected data and classify customer's behavior. Merchants can use the application to recommend items to customers based on their behavior and to increase the sales. It is not possible for the users to process huge amount of information. Thus recommendation systems are useful applications in the area, especially for the recommending product items in the electronic commerce environment. Furthermore, the community is a special situation on the Internet. Not only can we recommend to the user on a certain site but also to the user on other sites among the site communities. On the basis of the recommendation system and site communities, we propose a recommendation architecture based on the mobile agent technology.

We have surveyed several related research areas and explain briefly in Section 2. We describe the system software components in Section 3. And in Section 4 we propose our research approach and discuss the recommendation engine in detail. We then delineate our implementation method and mobile agent framework in Section 5. Finally, we draw our conclusion and discuss the future work in Section 6.

2. Related Work

In this section we discuss what researchers have done in the recommendation research and agent technology areas. Paul Resnick and Hal R. Varian propose definitions and directions fro recommendation systems in 1997 [2]. The idea is that users cannot make a choice without sufficient personal experience or other people's suggestion. Thus, automated recommendation by the computer system would be very helpful in filtering product information. Marko Balabanovic and Yoav Shoham proposed a recommendation system --- Fab in 1997 [3]. They combine content-based filtering and collaborative filtering approaches to design their system. Content-based recommendation recommends product items to users who have purchase records or showed interest from browsing on the web. On the other hand, collaborative recommendation classifies users into several groups. Users in the same group are of similar behavior or interest on product items.

Collaborative filtering [3-9] is a popular approach in the recommendation research field. Badrul Sarwar et al. discuss several techniques of recommendation system in 8]. They have concluded three basic steps in recommendation systems, including representation of input data, neighborhood formation and recommendation generation. In [10] they propose a collaborative filtering system that combined personal information filtering and agent techniques. On the other hand, traditional knowledge discovery is also popular in analyzing on-line behavior. In [11,12] they use the data mining technology to find association rules to generate meaningful patterns.

In [13] the authors presented the concept of the agent-based software architecture. The agent communication language and the agent architecture are two important issues. In order to achieve information and message sharing, the agent communication language is the basic mechanism. In order to build the agent computing environment, the agent architecture is another critical problem.

BASAR is a personalized agent system that keeps the web links based on the user bookmarks [14]. The system is able to support information updating and reduce the number of links by deleting less used items. In [15], a web-based information browse agent is proposed. The system uses the KQML as the agent communication language and reduces networking load. And in order to reduce the complexity of browsing, they use the structure meta-information mechanism.

The mobile agent technology overview can be found in [16-18]. Aglets [19] is a mobile agent platform based on the Java technology. The system uses Agent Transfer Protocol as the agent communication infrastructure and the architecture is able to support persistence, security, and agent collaboration. Voyager [20] and Concordia [21] are another two mobile agent systems that also support the agent communication and agent computing environment. The mobile agent platform, MAGNA, and its architecture are proposed in [22].

3. Recommendation System Overview

In this section, we show the overview of the recommendation system for making suggestions to the customers. Our proposed system includes two basic components. The first one is the mobile recommendation agent in the client machine. And the second one is the mobile recommendation and the agent platform on the server side.

3.1 Mobile Recommendation Agent

We define a *CARAgent* as the mobile recommendation agent in the client machine to capture the customer behavior and *CARAgent* is a mobile agent that will help customers to find their favorite products and service. And *CARAgent* will capture user navigation and buying behavior for customer behavior analysis. On the other hand, if customers cannot make decisions or they do not have any idea what to buy, they can ask the *CARAgent* and give what they want to the agent. Then the agent will recommend something to them. Besides this, the agent can also be a broker between the merchant and the customer. In the context of the mobile recommendation agent, the basic information and agent tasks are as following :

- **User real-time action:** The action information includes hyperlinks that the user clicked, the types of the hyperlinks, and products the user orders.
- **User profile:** The agent holds the user preference when the user logins to the system.
- **Gateway:** This component is the bridge between the system platform and the client machine.
- **Monitor:** This component is the administrator that captures the user action.

The recommendation system platform includes several components and the following are the overview of these components.

3.2 Interface Agent

The agent is the communication channel between the platform and the mobile recommendation agent in the client machine. The agent will verify the customer's privilege and identification. Correct profiling a specific customer must be insured.

3.3 Extraction Agent

When customers are on the merchant site, customers would navigate the products or services and buy something on the site. The information would be useful data to find their favorites and interests. The task of this agent is to transform the user real-time action and navigation paths to suitable meta-data from the mobile recommendation agent.

3.4 Predict and Analysis Agent

The agent analyzes the user behavior and predicts the best recommendation products for the user. The basic function of the agent is the user classification method and the automatic inference machine that generats recommen-dation targets.

3.5 MA

It is the merchant agent that holds the merchant strategy to assist generating the recommendation. The strategy ontology includes discount for the users or value-added service bound with the product or service.

3.6 Recommendation Generation Agent

After processing all the information, the result is sent to the agent to generate the recommendation. The recommendation agent also recommend something to the merchant for making better strategies.

3.7 Database Management System

The system will get and store several kind of information in the database. So a good database management mechanism is an important function. The database will stored the customer behavior information, personal profile, the site topology and product information. The customer and product relation would be store in the database also. In the following we define our database structure that plays as a major repository center.

Member Table

- member ID: the identification of the member
- member name: member's name
- age: actual member's age
- education: member's education level
- career: the category of member's profession
- country: the name of the country where the member lives
- address: the address where the member lives or members contact location
- email address: the electronic mail address of the member
- login Id: the identification of the user to login in the system

- password: The login passport

Merchant Table

- merchant ID: the identification of the merchant
- merchant name: the merchant's name
- login ID: the identification for the merchant login
- password: the passport of the merchant
- address: the merchant contact address
- email address: the merchant electronic mail address

Product Table

- product ID: the identification of the product
- merchant ID: the identification of the merchant who sells the product
- price: the product selling price

Site Topology Table

- product ID: the identification of the product
- page ID: the identification of the site presentation page

Linked Table

- page ID: the identification of the site page
- link ID: the link page identification

Feature Table

- feature ID: the feature identification
- characteristic name: the feature name

SF Relation Table

- page ID: the identification of the site presentation page
- feature ID: the feature identification

CB Table

- member ID: the identification of the member
- transaction ID: the transaction identification
- product ID: the product identification
- volume: the amount of the product the user bought

Transaction Table

- transaction ID: the transaction identification
- start time: the start time of the transaction
- end time: the end time of the transaction

CN Table

- member ID: the identification of the member
- page ID: the identification of the presentation page
- transaction ID: the identification of the transaction
- duration start time: the navigation start time that the user brows the page
- duration end time: the navigation end time that the user brows the page

4. The User Behavior Model

In the electronic commerce environment the customer navigates what they wanted on World Wide Web. Customer's behavior is showed according to the navigation processes. The customer's navigation sequence is useful for us to find why they buy or why they are interested. This information also hides some meaningful information for the merchant. First we define a customer behavior record to describe the user's preferences and on-line history. And then we propose a graphic model to analyze for recommendation. The model is used for discover correlation between customers, product items and product features.

4.1 Custom Behavior Record

In order to express the customer behavior in the electronic commerce environment, we build a customer buying and navigation record (CBNR) according to the user behavior. First we define a customer buying history record, *CB* set.

$$CB_{i,\,t} = \{\ p_1, p_2, \ldots\ldots, p_n\ \} \qquad (1)$$

$CB_{i,\,t}$ defines what products user i buys in set P in transaction T. Product set P is a set includes product items customer i bought. T is a transaction customer bought something. Secondly, we define a customer navigation history set $CN_{i,t}$.

$$CN_{i,t} = \{\ N_1, N_2, \ldots\ldots, N_n\ \} \qquad (2)$$

$CN_{i,t}$ defines what item page user i navigates in transaction T. And N_j is defined as a record $N_j = \{\ P_{id}, stime, etime\ \}$, where P_{id} is the identification of the page, *stime* and *etime* are start time and exit time that customer i enter the page and exit the page.

According to the CBNR record we can build a graphic model – the CPF model. The model has three views including the customer-product view, the customer-feature view and the product-feature view.

4.2 Customer-Product-Feature Model

Based on users' browsing or navigation information, we proposed a graphic model to analyze the user behavior. The model is referred to as the CPF model. The model has three dimensions including the product dimension, the customer dimension and the feature dimension. Figure 1 shows the CPF model with respective information. The model has three views that are explained in the following.

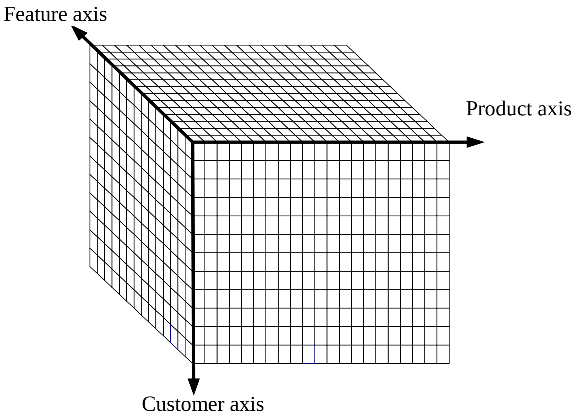

Figure 1. The Customer-Product-Feature model

4.2.1 Customer-Product Matrix

This view stores the relation between the product and the customer. The value in the view shows whether customer *i* buy product *j* or not. The value is of the set {0, 1}. If customer *i* buys product *j* on the E-commerce site, the corresponding value is "1". Otherwise, the value is "0". For example, if customer i bought product 2, product 4, and product 7, the content will be filled "1". Otherwise, the content will be filled "0". And the metric shows the buying relation arrangement between customers and products. In this view we can discover potential customers, who bought most products in the past. And we can identify the potential products, which are bought by most users.

4.2.2 Product-Feature Matrix

This view shows the relation between the product and the feature. The value in the view represents whether product *j* has feature *k* or not. The value is defined in the set {Y, N}. The symbol "Y" represents a "have" relation. On the contrary, the symbol "N" denotes a "not-have" relation. The statistic data record the navigation time of the product by the customer will be stored in the metric. And the information can be used to compute the user similarity and classification. P_jF_k in the metric is a two element content $P_jF_k = \{ V_{jk}, S_{jk} \}$. V_{jk} is a Boolean value which is defined as the above and S_{jk} is the statistical data computed with Equation (3).

$$S_{jk} = \textit{total navigation time of feature } k \textit{ /total navigation time of product } j \qquad (3)$$

4.2.3 Customer-Feature Matrix

The final matrix is the Customer-Feature metric which shows the relation between the customer and the feature with respect to the product. When the customer browses the site, his/her interests on certain features of the product would reveal. The metric content is in the real number domain. And in this view each row is a navigation vector corresponding to a customer. With the navigation vectors, we can compare the customers with each other to find their relationship and classified them into heterogeneous groups.

When a customer navigates from a page to another linked page, we can compute a navigation vector according to the site topology and product features. If a product has five features, the navigation vector would have five elements. The navigation vector format of customer *i* would be the form $NV_i = (f_1, f_2, f_3, f_4, f_5)$. Each element in a navigation vector shows the interest of a customer to a particular feature of the product. Before calculating the navigation vector, we would calculate the navigation factor Nf_k, which value is in the domain {0,1}, for the navigation vector formula. If the page shows feature *k* the value of Nf_k is "1". Otherwise, the value is "0". Finally element *k* in navigation vector *i* is calculated by Equation (4).

$$\text{Navigation element}_k = \frac{\sum Nf_k}{K} \times Avg(Nf_k) \qquad (4)$$

$$Avg(Nf_k) = \frac{time\_of(Nf_k)}{total\_time(P_j)}$$

where K is the length of the navigation sequence of product j on which customer i navigates, $time\_of\ (Nf_k)$ is the total time that the customer navigates feature k of the product, and $total\_time(P_j)$ is the total time that customer navigates on all the feature of product j.

For example customer A navigate the product P and he/she walks through the page sequence in the Figure 2. The navigation graph is an extended sub-graph of the site topology. In Figure 2 we can know customer A has a navigation sequence { a, b, c, d, b, c, e, d, f } and we call the sequence a transaction. Table 1 shows the description of the product presentation page, features and navigation time of each feature. We can build a navigation vector according to the navigation flow. In the transaction we established nine navigation factors (1,1,0,1,0,0,0), (1,0,0,1,1,0,1), (0,1,1,0,0,0,1), (1,0,0,1,0,1,1), (1,0,0,1,1,0,1), (0,1,1,0,0,0,1), (0,0,0,1,0,0,0), (1,0,0,1,0,1,1), (0,0,1,0,1,0,0) and a navigation vector, which is calculated by the Equation (4) (0.323, 0.127, 0.112, 0.442, 0.091, 0.059, 0.461) for the customer.

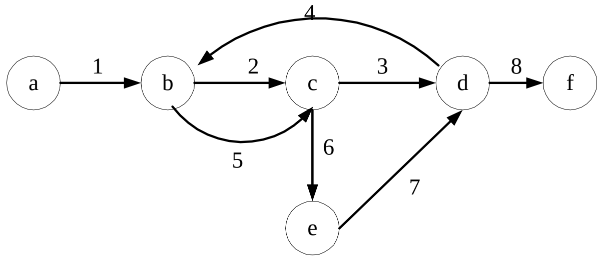

Figure 2. An example of customer navigation sequence

Table 1. A navigation sequence of transaction table

Sequence ID	Features ID	Navigation time
(0, a)	1,2,4	15
(1, b)	1,4,5,7	13
(2, c)	2,3,7	17
(3, d)	1,4,6,7	20
(4, b)	1,4,5,7	7
(5, c)	2,3,7	10
(6, e)	4	9
(7, d)	1,4,6,7	9
(8, f)	3,5	10

4.3 Recommendation Algorithm and Process

In the above we build a CPF model for representing the user behavior. In this section we propose two algorithms to classify customers into groups based on user behavior and to find frequent buying patterns, respectively.

Algorithm 1: Generate User Groups for Products
Input: *CF matrix, k as the cluster number, n as the customer number*
Output: *User groups*

```
{
    for i = 1 to k
        Gi <- i
    do
    {
        for i = 1 to n
        {
            most <- ∞
            for j =1 to k
            {
```

$$temp = \sqrt{\sum_{z=1}^{m} \left| C_i F_z - C_j F_z \right|^2}$$

```
                if temp < most
                    index = j
                    most <- temp
            }
            Gindex <- Gindex ∪ i
        }
        for i = 1 to k
        {
            temp <- 0
            for j = 1 to Gi.length
                temp <- temp + Gi, j
            Gi <- temp / Gi.length
        }
    } while set G changes
    output G
}
```

Algorithm 2: Discover Buying Pattern
Input: *CP Matrix, minimum support count*
Output: *Buying patterns*

```
{
    for each product j
    {
        count <- 0
        index <- j
        for each Ci
            if CiPj = 1
                count <- count + 1
        S_index <- j
        S_index.count <- count
    }
    max <- index
    do
    {
        for i =1 to max
        {
            max_temp <- max
            for j = i to max
            {
                count <- 0
                for each Ck
                {
                    if (CkPi = 1) & (CkPj = 1)
                        count = count + 1
                }
                if count >= MSC
                    S_i.count <- count
                else
                {
                    for m = i to max-1
                        S_m <- S_m+1
                    max_temp <- max_temp – 1
                }
            }
        }
```

```
            BP <- S
            max <- max_temp
        } while max > 0
        output BP
}
```

With Algorithm 1, which is based on the k-means classification method, customers can be classified into different classification groups with respect to a certain product. Combining with merchant strategies, the resulting information then can be used to make personal recommendation. *A priori* algorithm is a method for finding frequent itemsets [11,12]. We enhance the algorithm for the discovery of buying patterns. The general sequences of buying behavior can be found in Algorithm 2. With the information about buying patterns, we can decide what to recommend to the users who have not bought.

Based on the above algorithm, we can find the basic recommendation targets according to the user's behavior. Besides the real-time user behavior, we can also define a rule pool based on the user's background information, market trends and merchant's strategies etc.. Rules can be added or removed from the rule pool according to the actual market trends and information.

In the following, we describe four basic recommendation events in our system. The events include existing users, new users, new products and special events.

Existing Users: In Figure 3 we divided our recommendation process for existing users into four stages.

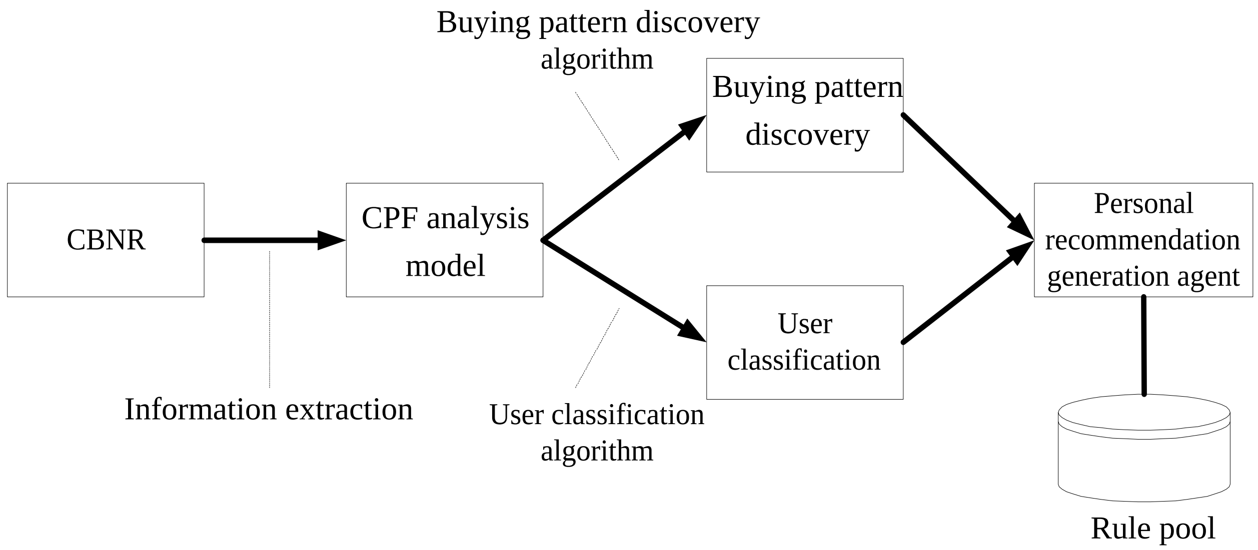

Figure 3. Recommendation process of existing users

The first stage is the CBNR user profile, which ecords user navigation behavior and buying information be generated from the agent in the client's machine. The result is combined with the CPF analysis model through information extraction procedure. And then the user classification algorithm (Algorithm 1) and the buying pattern discovery algorithm (Algorithm 2) are used to find products to be recommended. At the last stage, the agent generates personalized recommendation that is suitable for each customer. Each customer can get product information of his/her interest. In this stage we generate recommendation based on user groups and buying patterns. First, we generate the recommendation target according to each user group. If a user did not buy an item that the others bought in the same group and the user's navigation vector is larger than the average navigation vector, we recommend the item to the user. Secondly, if a user has bought an item, we search for other items in each buying pattern, and choose those items with the user's navigations vector larger than the average navigation vector according these items. Then we recommend these selected items to the user. At last we merge the two results from the above and then filter the results with the rules within the rule pool. The algorithms 3 is the detail of the recommendation process.

Algorithm 3: Main-Recommendation
Input: *CBNR, product catalog*
Output: *recommendation_list*

```
{
    for each group
        for each user
            for each transaction t
                for each CB_user,t
                    If !(product in CB_user,t ) then
                        recommendation_list[user] <- product
                else for each pattern
                    if user in the pattern then
                        for each product
                            if !(product in CB_user,t) then
                                recommendation_list[user] <- product
}
```

New User: Figure 4 shows the recommendation flow that for new users.

When a new user browses the site, he/she also leaves their navigation information that represents his/her behavior. So we use this information to build the CBNR user profile and analyze user preferences. We compute the similarity with existing users to find what can be recommend. As the same as the first recommendation event we also using personal recommendation generation agent to produce recommendations. And we use Euclidean distance as the similarity function to calculate the similarity of existing users and the new user. Equation (5) is the formula of the similarity measurement.

$$Sim(a,g) = \frac{\sum_{j=1}^{n} C_a F_j \times C_g F_j}{\sqrt{\sum_{j=1}^{n} C_a F_j^{\,2}} \times \sqrt{\sum_{j=1}^{n} C_g F_j^{\,2}}} \qquad (5)$$

where k is the new user and i is the existing users. And $C_i F_j$ express the value of the navigation element in the corresponding navigation vector.

If the new user did not leave any real-time behavior, we also provide a user interface for the new user to answer questions related to the product features and user interests. We can then calculate the similarity with all existing users and the new user and find the closest users who are inside a distance. According to the closest set, we make the personal recommendation based on the previous rating.

New Products: Figure 5 shows the process of recommend new products to the user.

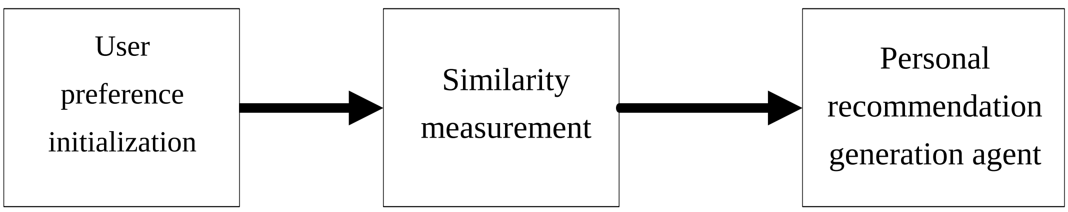

Figure 4. Recommendation process of new users

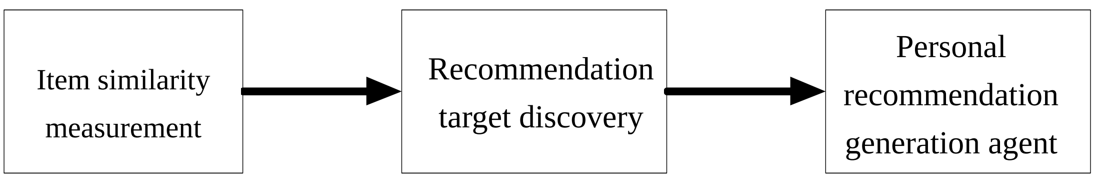

Figure 5. Recommend to users with new products

In this recommendation event, we must find existing products, which are to the new product. We use item similarity algorithm to find the related items. The similarity measurement can be obtained with the dissimilarity matrix show in the Table 2.

Table 2. Dissimilarity matrix

	Item i			
Item j		1	0	
	1	a	b	a+b
	0	c	d	c+d
		a +c	b+d	a+b+c+d

According the PF matrix, “a” is the count that both item *i* and item *j* have the some feature. “b” is the count that item *i* has the feature but item *j* does not. “c” is the count that item *j* has the feature but item *i* does not. “d” is the count that both item *i* and item *j* do not have the feature. So the dissimilarity function is showed in Equation (6).

$$dissimilarity(i, j) = \frac{b + c}{a + b + c + d} \tag{6}$$

We use the equation as the similarity measurement function. According to the similarity measurement results, the recommendation target discovery agent find the closest similar product and recommend to the user by the personal recommendation generation agent.

Special Events: The recommendation event is based on the marketing strategies proposed by the merchants. The recommendation events are dynamic and changing with time. The user’s buying history and background information are the basic knowledge for recommending with special events.

4.4 Recommendation Measurement

The recommendation quality is an important issue in the research. The recommendation quality means that the recommendation events are close to the user’s interesting and the user would be attracted by the recommendation events. Hence in the section we provide the measurement formula to define the better recommendation quality. And we also provide an enhance approach to adjust the recommendation events according to the recommendation quality.

We defined some variables to model the recommendation quality first. In the recommendation processes we would produce several recommendation materials to users, but some of these recommendation would be response by the users. So we define the response rate and active rate as the measurement of recommendation events.

4.4.1 Response Rate

The response rate means the number of response from users according to the number of all recommendation events. We can formula the definition as the Equation (7).

$$RR = \frac{number\_of\_response}{number\_of\_events} \tag{7}$$

4.4.2 Active Rate

The active rate means the number of products that user bought actually according to the number of all recommendation events. In the same way we can formula the definition as the Equation (8).

$$AR = \frac{number\_of\_bought}{number\_of\_events} \tag{8}$$

According to the measurement of the response rate and active rate. We can found the situations that we must adjust the recommendation generation processes.

Case 1: If the response rate is large enough but the active is low, we can know that the recommendation event attracted the users. But the user do not buy we can provide the special discount to users.

Case 2: If the response rate is low, we found the users would be not attracted by the recommendation. So, in order to produce better recommendation events we must adjust the recommendation generation processes. We used hierarchy approach to enhance the recommendation generation. The basic approach is that we would divide the existing groups into smaller groups. And each smaller group would be the better recommendation than previous groups. The Figure 6 showed the logic view of this hierarchy approach.

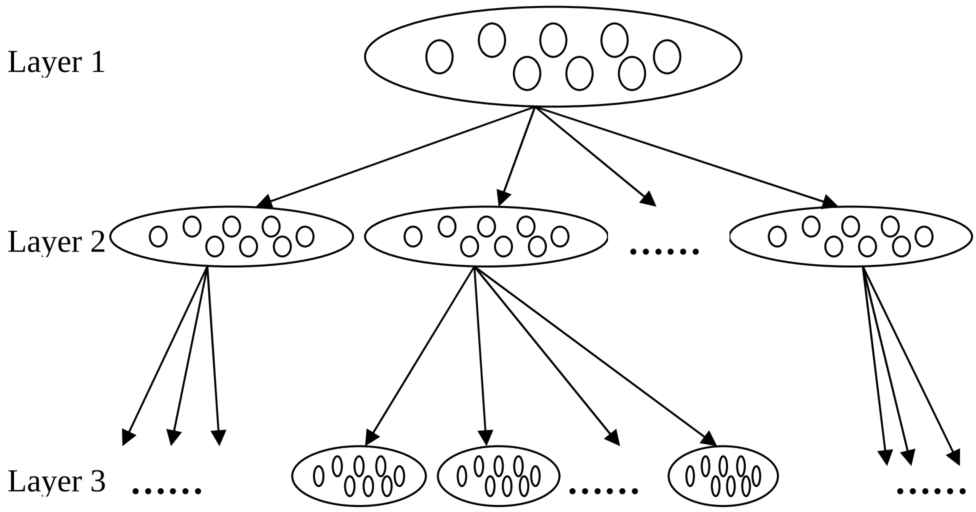

Figure 6. The hierarchy of sub groups

5. The Mobile Agent Framework

5.1 The Hierarchy of Generic Mobile Agent Platform

In the above sections we have presented our recommendation algorithm and analysis methods. In this section we propose a generic agent platform that supports the recommendation system. We use a hierarchical approach to design our mobile agent platform. Our goal in the designed architecture is not only to provide an environment for the recommendation, but also to provide an execution environment for electronic commerce applications. In this paper we focus only on the recommendation system part. The platform includes agent execution layer, agent control layer, agent computing layer, and agent common layer.

5.1.1 Agent Execution Layer

This layer is the place where the agent executes. Each agent in the layer is alive when the user navigates the electronic commerce site that hosts the mobile agent platform. When the user leaves, the agent is dead. When the user accepts the recommendation from another community site, the agent is suspended. And the suspended agent would turn dead if the user is not back within a time limit.

5.1.2 Agent Control Layer

The agents in the layer control the state of the agents in the above layer. The control agent would know how many agents in the electronic commerce site and all agents in the agent execution layer will communicate with each other through the control agents. The layer is also a mediator between the agent hierarchy and in the agent community. Furthermore, the agents' name and location would be stored in the layer's agent database. In order to communicate with other agents and agent platforms, we use MAF [23] interface to achieve the functionality. The MAF is a collection of destinations and interfaces that provide an interoperable interface for the mobile agent system. The interface includes MAFAgentSystem, which defines the agent operation and MAFFinder interface, which defines the agent registering, unregistering and locating services.

5.1.3 Agent Computing Layer

The agents in the layer would process the information from the users. The agent is a kind of the information process agent that processing information in background and would not communicate with the user agent directly.

5.1.4 Agent Common Layer

The layer includes the basic library that all agents would use. They included the agent security manager, the agent communication protocol, the agent transportation manager, the agent manager mechanism and the agent naming service. Figure 7 shows the generic mobile agent platform architecture.

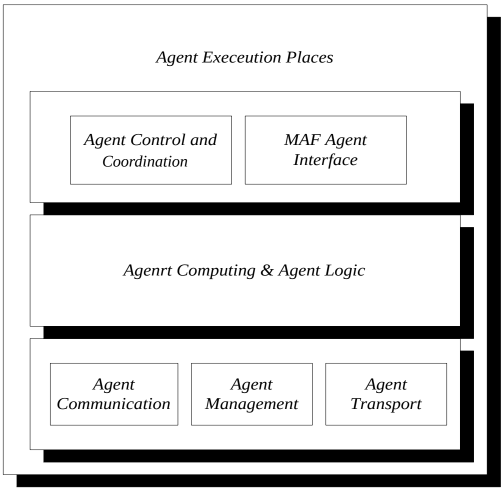

Figure 7. The generic mobile agent framework

5.2 Collaborative Agent Architecture for Recommendation Application

In this section we propose our agent computing architecture for the recommendation application. The architecture is a multi-agent architecture and the agents work collaboratively. In the architecture we conclude several types of agents including information processing agents, task agents, and information provider agents. Information process agents would process the complexity computing work in our system, e.g. customer classification. Task agents would handle the basic function on the electronic commerce site, e.g. user identification, mobile agent control, etc. And information provider agents would be the broker between other agents that need any information about the site. The three agent types form the fundamental of the distributed computing architecture. Figure 8 shows the collaborative agent computing architecture for recommendation application.

In the figure we deploy a mobile agent, *CARAgent*, in the client machine for recording the user behavior. On the other hand, the client agent would be the mediator of the agents in the server. The client agent can accept recommendations from the recommendation agent. If users want to search the product, users can summit a query to the *CARAgent* and the agent will send the request to the processing agent in the server. And then the processing agent in the server will process the request and give recommendations based on the request and user preferences. So the *CARAagent* is a mobile agent that can assist the user and give personal recommendations.

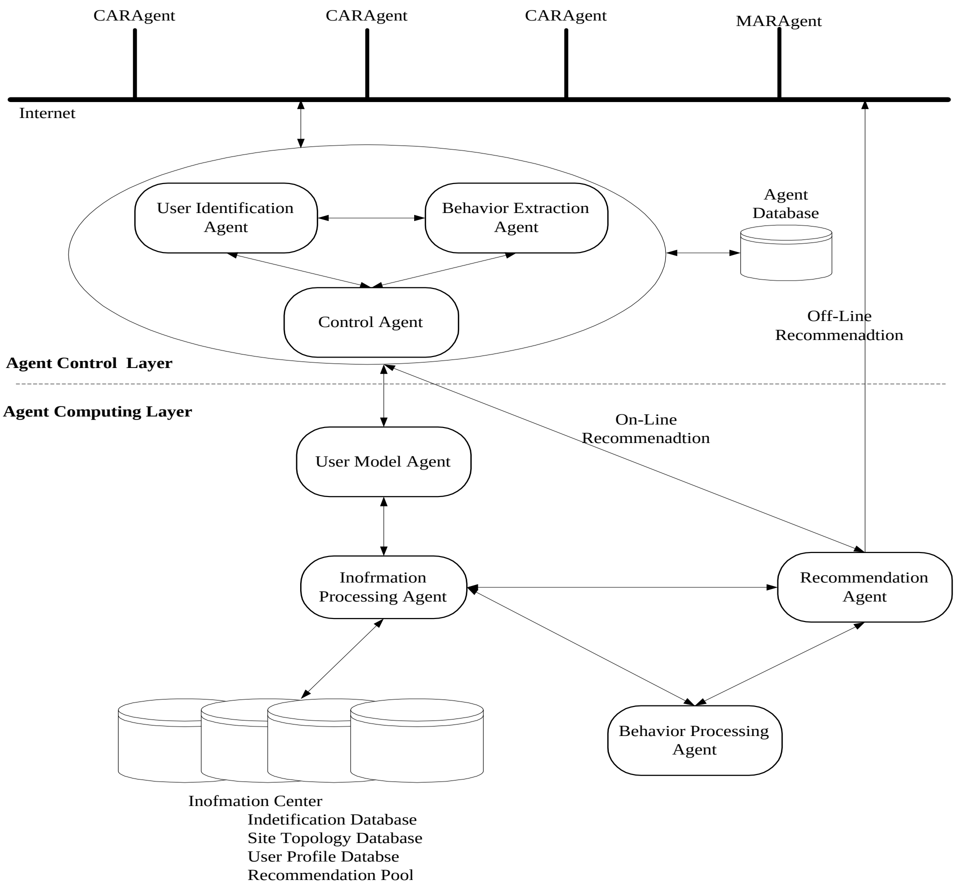

Figure 8. A collaborative agent architecture for recommendation application

We also deploy *MARAgent* in the merchant machine. The merchants can place their products on the electronic commerce site through the *MARAgent*. Moreover, the agent is the mediator between the server and merchants. Merchants can monitor the product state and then design a better strategy.

In addition to the user and merchant mobile agents, there are several agents in the server that assist the mobile agents. Upon entry to the server, an agent group is formed to be the manager of the mobile agents and in the group there exists three agents and a database. The user identification agent handles the authenticate task in the system and is the security manager in the system. When the user browse the site, the *CARAgent* records the navigation sequence that includes product navigation and buying behavior. The behavior extraction agent decodes the navigation information into suitable categories, and sends to the behavior process agent. The control agent controls the agent life cycle and monitors whether the agent is alive or not. The agent group holds an agent database that stores the agent information. The control agent is the coordinator between the mobile agents. If a *CARagent* or a *MARAgent* wants to communicate with the other one, the message is sent through the control agent.

When the agent group finishes the extraction task, the information is sent to the user model agent. The user model agent builds the user profile according to the information from the agent group and the individual user. The user profile is then given to the information process agent and the agent stores the profile into information center. The information agent is the coordinator of the database access. The agent processes all the actions in the information center.

The behavior processing agent analyze the user behavior using the CPF model proposed in section 4 and finds the recommendation targets sent to the recommendation agent. The behavior process agent is a deamon running in the server and find recommendation targets in real time.

The recommendation agent finally performs the recommendation. The recommendation agent accepts the recommendation targets and finds the suitable recommendation material. The agent then recommends to the user if he/she is still on the Internet. The process is an on-line real-time recommendation process. If the user is not on the Internet, the recommendation agent can recommend by email or other transmission method. When the user login to our system, the agent could provide the newest recommendation to them.

Finally, we have an information center in the server. That includes the identification database, the site topology database, the user profile database and the recommendation pool. The identification database stores the basic information and login information about each user. The site topology database stores the product information and presentation structure in the site. The user profile database stores the user behavior as the individual profile and transactions. The recommendation pool stores the recommendation material and targets that would not be delivered, at once.

5.3 Community Recommendation

One the advantage of the electronic commerce environment is that all participants can exchange information with each other. In the paper we also apply this advantage to the designed recommendation system in the nature way. We proposed the community recommendation mechanism based on the mobile agent technology. We consider that the electronic commerce sites or portals would be arranged different location. But these electronic commerce sites wanted to get more profit and sell more product or service. So in the paper we proposed mechanism based on the design of the generic agent framework to achieve to community recommendation. The community recommendation idea can be showed in the figure. We consider the community recommendation mechanism based on the peer-to-peer computing architecture that is the essential network computing architecture in the past. The basic idea of this architecture is that all participants in the architecture can interact with each other directly. Based on this idea we consider the current system, the users must know the complete location of the electronic commerce and they can begin to browse the favorite goods. If the site had the better goods or products, the user cannot know and browse the better one. So in the design of the community recommendation mechanism we can solve the disadvantage. All electronic commerce site can send the recommendation events to others and the other site would recommend the products to the suitable users for the sending site. In the mechanism all electronic commerce can get more profit or sell more products or service. Therefore we proposed an environment based on the design of the section 5.1 and achieve the community recommendation goals. Figure 9 showed the community recommendation mechanism.

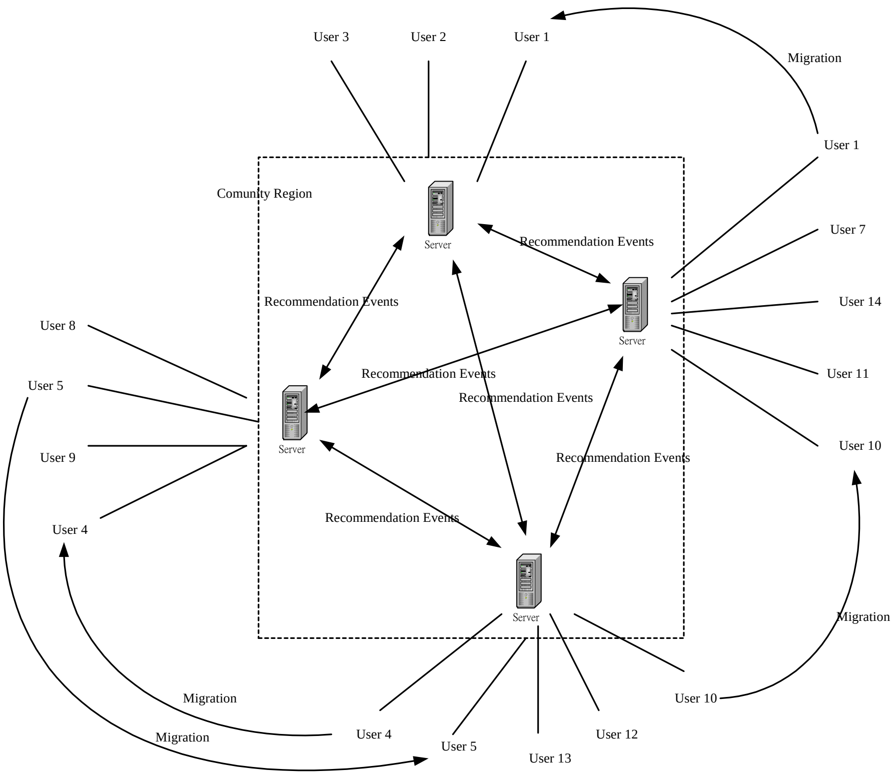

Figure 9. The community recommendation mechanism

In order to make recommendation to other site the recommendation agent must migrate to another sites and make recommendation to the users. We proposed two approaches as the community recommendation mechanism. First the recommendation events can be announced to another sites in the community. And the other is that user's agent can migrate to another site to survey the recommend product or service. Therefore we proposed a mechanism for transferring the recommendation message from one site to another. Figure 10 is the message flow of the real-time recommendation across the electronic commerce sites.

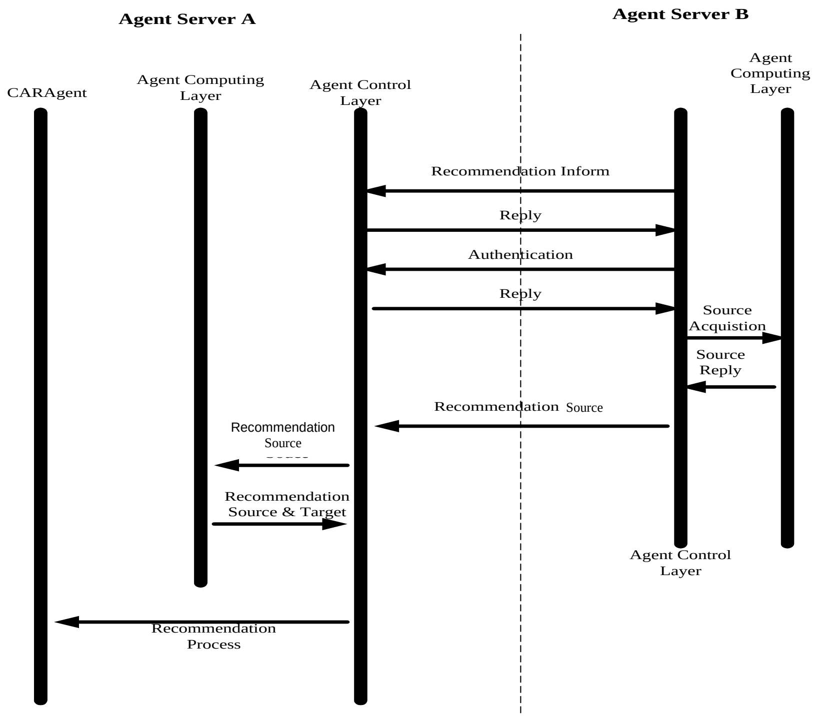

Figure 10. Mobile recommendation mechanism & message flow

The following steps are the recommendation flow between the sites in the community.

Step 1: Source site send the recommendation inform action.

Step 2: The sites must authenticate the source site. If authentication process is success, then the site would reply the source site. Otherwise the destination site would reject the recommendation inform.

Step 3: Source site send recommendation material and recommendation messages.

Step 4: The destination site accepts the event.

Step 5: The destination site calculates the suitable users who are on line and send the recommendation to the users.

On the other hand if the recommended user wanted to navigate the recommended product or service, the user can migrate to the electronic commerce site. We also proposed the simple migration mechanism for the agents that can move to another site. Figure 11 is the message flow of a simple agent migration protocol.

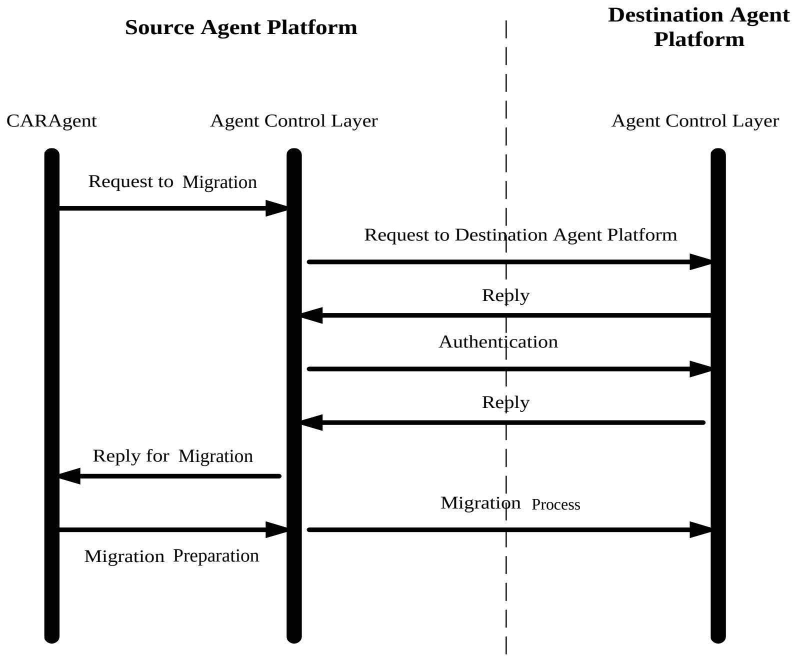

Figure 11. Simple agent migration mechanism

In the migration flow the first step is to send migration request to the destination site. And then the site must authenticate with each other. If the above processes were correct, the actual agent would be transferred to the destination site and to be executed.

According to the design of the community recommendation mechanism we can conclude the agent life cycle as the following. The user agent would start at the initial state that denoted the user begin to browse the electronic commerce.

5.4 Implementation

Our system is based on the Microsoft Windows platform. The implementation of the system follows the popular software development trend – distributed computing. Hence we develop a mobile agent server platform that can host agents from the client's machine. We use the agent control proposed by Microsoft as the agent program interface. When the users navigate the E-commerce site after login to the system, the agent is launched from the client. Wherever the user goes the agent would follow. In the mobile agent server, we program a platform based on the MS-Windows environment. The agent communication mechanism with the server is built based on the Windows SDK. The platform also provides agent naming service, so that each user can be identified.

Three servers are included in the basic runtime environment. That includes the web server, the agent server and the database server. The IIS is used as the web server that hosts all web documents and materials of the E-commerce site. And the agent server is programmed based on the Windows DLL library. All data are stored in the database server that is the MS-SQL server in our implementation. Figure 12 shows our implementation architecture.

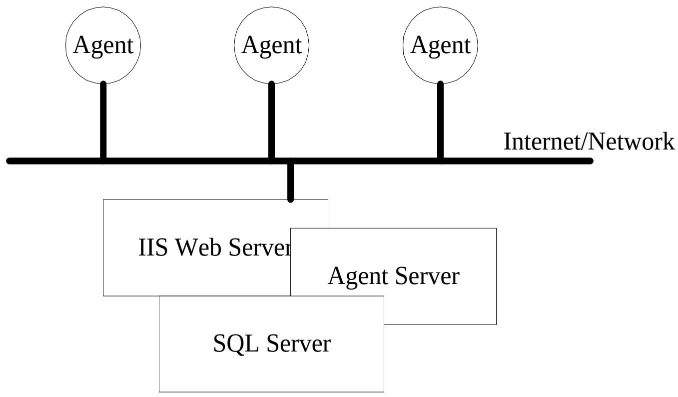

Figure 12. Implementation architecture

6. Conclusion

In the paper we present some works about our proposed recommendation system. We propose a browsing-oriented approach to represent the on-line behavior of the user. And the method is a popular navigation function based on user behavior on the Internet and the World Wide Web. We can discover relationship among customers based on user classification and buying patterns. We also propose a mobile agent platform for mobile recommendation over the Internet and other electronic commerce applications. Finally, a collaborative agent computing architecture is designed for our recommendation application based on the generic agent framework.

In the proposed integrated approach, first, we can find the relationship between the customers, products, and features and based on the result we could make better business strategies. The integrated approach got better response from customers and predicted customer's behavior precisely. However, the mobile computing is another important issue in recent year, the proposed mobile agent framework is the core value of the mobile computing environment. We can develop several electronic commerce applications upon the mobile agent framework and reduce the network latency and traffic. And we develop the recommendation system based on the framework and the analysis strategy. The collaborative recommendation system, integrated analysis approach, and the generic mobile agent framework are the main contributions in our paper. On the other hand the community recommendation is another value in our research.

In the future we will enhance our classification algorithm and make multimedia recommendation. Customer profile security and privacy is an important issue for the customer in the electronic commerce environment. We will add the security and protection mechanism into our recommendation system, especially about the customer's privacy. Finally, we believe the generic agent framework will be a useful prototype for electronic commerce environment and the proposed recommendation system is a useful approach to maintain good customer's relationship.

References

[1] Guttman R., Moukas A., and Maes P.," Agent-mediated electronic commerce: A survey," *Knowledge Engineering Review*, Vol. 13, pp.147-1594 (1996).

[2] Paul R. and Hal R. Varian. "Recommendation Systems," *Communication of the ACM*, Vol. 40, pp. 56-58 (1997).

[3] Marko B. and Yoav S. "Fab: Content-based Collaborative Recommendation," *Communication of the ACM*, Vol. 40, pp. 64-72 (1997).

[4] Basu, C., Hirsh, H. and Cohen. W. W. "Recommendation as Classification: Using Social and Content-based Information in Recommendation," *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, WI, U.S.A., pp. 714-720 (1998).

[5] David G., David N., Brian M. O. and Douglas T., "Using Collaborative Filtering to Weave an Information Tapestry," *Communication of the ACM*, Vol. 35, pp. 61-70 (1992).

[6] Fisk, D., "An Application of Social Filtering to Movie Recommendation on the Web," *Software Agent and Software Computing. Springer- Verlag*, pp. 116-131 (1997).

[7] Paul R. N. I., Mitesh S., Peter B. and John R., "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," *Proceedings of the 1994 Computer Supported*

Cooperative Work Conference, N.Y., U.S.A., pp. 175-186(1994).
[8] Sarwar, B. M., Karypis, G., Konstan, J. A. and Riedl, J. “Analysis of Recommender Algorithms for E-Commerce,” *Proceedings of the ACM E-Commerce 2000 Conference*, pp. 158-167 (2000).
[9] Uprendra S. and Pattie M. “Social Information Filtering: Algorithms for Automating Word of Mouth.” *Proceedings of the Conferemce on Human Factors in Computing Systems (CHI’95)*, CO, U.S.A., pp. 210-217 (1995).
[10] Good, N., Schafer, J. B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., and Riedl, j. “Combining Collaborative Filtering with Personal Agents for Better Recommendation,” *Proceedings of the Conference of the American Association of Artifical Intelligence (AAAI-99*), pp. 439-446 (1999).
[11] Agrawal, R. and Srikant, R., “Fast Algorithm for Mining Association Rules,” *Proceedings of the 20th VLDB Conference*, Santiago, Chile, pp. 487-499 (1994).
[12] Agrawal, R. Imielinski, T. and Swami, A. “Mining Association Rules Between Sets of Items in Large Database,” *Proceedings of the ACM SIGMOD Conference on Management of Data*, Washington D.C., U.S.A., pp. 207-216 (1993).
[13] Genesereth, M. R. “Software Agents,” *Communication of the ACM*, Vol. 37, pp. 48-54 (1994).
[14] Christoph, G., Thomas and Gerhard F., “Using Agents to Personalize the Web.” *Proceedings of the 1997 ACM IUI Conference (IUI’97)*, Orlando, FL, U.S.A., pp. 53-60 (1997).
[15] Chanda, D. and Martin, F., “Information Agents for Automated Browsing,” *Proceedings of the 1996 ACM CIKM Conference(CIKM’96)*, Rockville, MD, U.S. A., pp. 296-305 (1996).
[16] Kiniry, J. and Zimmerman, D., “A Hand-on look at Java Mobile Agents.” *IEEE Internet Computing*, Vol. 1, pp. 21-30 (1997).
[17] Nwana, H. S., “Software Agents: an Overview,” *Knowledge Engineering Review*, Vol. 11, pp. 205-244 (1996).
[18] Pham, V. A. and Karmouch, A., “Mobile Software Agents: an Overview,” *IEEE Communication Manazine*, Vol. 36, pp. 26-37 (1998).
[19] http://www.tri.ibm.co.jp/aglets
[20] http://www.objectspace.com/voyager
[21] http://www.meitca.com/HSL/Projects/Concordia
[22] Krause, S. and Magedanz, T., “Mobile Service Agents Enabling Intelligence on Demand in Telecommunications,” *Proceedings of the 1996 IEEE Global Telecommunications Conference*, London, U.K., pp. 78-84 (1996).

Manuscript Received: Jun. 22, 2001
Revision Received: Oct. 19, 2001
and Accepted: Aug. 13, 2002