A New User Model to Support Electronic Commerce

Rony Abi-Aad

A Thesis
in
The Department
of
Computer Science

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

July 2001

0-612-64073-6

ABSTRACT

A New User Model to Support Electronic Commerce

Rony Abi-Aad

User models incorporate the knowledge about the user so as to increase, in a general sense, the effectiveness of man-machine interaction. Several applications or software agents can use such model-based knowledge to give better and personalized services to end-users. The personalization can be in the form of filtering, presentation, resolving ambiguity or incompleteness, and proactive notification.

In electronic commerce applications, the user interface plays a key role in achieving user acceptance. This is more so, in the case of business-to-consumer type e-commerce. Based on the user needs, we classify the shopping behavior of consumers into three categories, which are called comparative shopping, planned shopping and browsing-based shopping. We propose a user model to suit these buying behaviors. The user model described in this thesis can be helpful in the development of personalized services and for building user interface and software-agents that can assist consumers both reactively and pro-actively. The proposed user model is applicable in the context of virtual or electronic malls that deal with broad as well as specific types of products and services.

Acknowledgements

Many people gave me different kinds of support throughout the course of this research work. I would like to mention a few of them.

Respect and gratitude go to my supervisor Dr. T. Radhakrishnan for his guidance and invaluable encouragement.

Best wishes go to Mashrur Mia. I thank him for the fruitful discussions we had and for implementing my ideas with elegance.

Special thanks go to my family for their moral support. Especially my father for being a role model, and my mother for being patient with me whenever I complain.

Financial support of this project provided by the Canadian Institute for Telecommunication Research through a research grant to Dr. T. Radhakrishnan is greatly acknowledged.

And finally thanks to all the people I met during my study, because they all inspired me in one way or another.

Table of Contents

List of Figures

1. Introduction

"People don't know what they want until it's proposed to them"

Sir Terence Conran

1.1. Overview of Electronic-Commerce

Commerce is one of the oldest activities that men have known. In fact, it has been known to cavemen. With the growing popularity of the Internet, it is only natural that commerce found its way into this medium. This kind of commerce where business is done using electronic means is referred to as "electronic commerce" or "e-commerce". It has been around for more than twenty years in the form of EDI (Electronic Data Interchange). But it is only when the Internet became increasingly popular that e-commerce started to grow rapidly [Bolin 1998] and hence several new issues are faced. In May 1997, global e-commerce generated $750 millions in sales; in May 1998 that figure had grown to $2.3 billion, a 205% increase. Consulting agencies like IDC, Active Media, and Forester Research all estimate that global e-commerce revenues will exceed one trillion dollars by 2003 [Rohn 1998]. Many reasons have helped to achieve this: the affordability of the Internet, the possibility to link people from all over the world and the convenience to exchange money electronically.

E-commerce includes trading of physical objects and intangibles such as services and information. This encompasses all the trading steps such as marketing, ordering, payment and support for delivery [Timmers 1998].

Three common types of e-commerce are known as business-to-business (B-to-B), consumer-to-consumer (C-to-C), and business-to-consumer (B-to-C). We focus our attention in this thesis on the B-to-C e-commerce model, which is also known as "retail-market". There are still many challenges to be tackled in this area. The main research focus of people in this area is to characterize and enhance the infrastructure needed for this type of e-commerce. The research efforts cover many areas of computer science, from databases and multimedia to networking and security, the quality of service, and user interfaces. Our interest is in the HCI (human-computer interaction) domain. Our aim is to provide a better service for the user by bridging the gap between the machine and the humans in information seeking and providing.

Electronic shopping is different from traditional shopping in several ways. First, in e-commerce, it is easy to go from one shop to another for the sake of price or quality comparison. In traditional shopping, one has to physically move from one place to another and maybe even travel for hours, which has its own limitations. On the other hand, in traditional shopping, companies spend large amounts of money on the store design and decoration to create a positive environment allowing for a pleasant shopping experience. This aspect is called "store front". In electronic shopping, this pleasant aspect has often been neglected: e-commerce sites have often been similar to ordinary web sites [Rohn 1998], and like most web sites, they suffer from bad usability [Nielsen 1994].

Since its very beginning, the Internet has been growing in popularity and complexity, and the largeness makes it difficult for the user to find the information he needs. Often, it is more difficult for many users to shop on the Internet than by conventional means. On the Internet, the user finds himself either "flooded" with irrelevant information mixed with some relevant information, or lacks relevant information. In this context, we note the following:

- User interfaces play an important role
- Queries usually return more matches than the user can consult, or fewer matches than expected

- The user is "flooded" by unwanted and sometimes unsolicited information
- The information is sometimes very badly organized making browsing difficult
- Finally, some of the pleasures of shopping in stores are missing when shopping through the Internet (communities, easier and reliable transactions, trust, human dialog, etc.)

One possible way to help in solving these problems is to personalize the interactions between the user and the e-commerce system based on appropriate user models. We believe that usability can be improved through personalization. The work in this thesis concentrates on this belief and we try to provide solutions to the above mentioned problems by means of user models.

1.2. Overview of Agents

Software agents have become very popular in the last six years or so, but the term "agent" in computer science area is neither new in itself nor well defined. Although definition and categorization of agents is hard, it is not reasonable to call any program, a software agent. In order to clarify what is a software agent, we compiled some of the properties of agents as viewed by different authors and this compilation highlights the differences between agents and ordinary software [Erdogan 2001] [Schroeder 2001] [Lu 1999] [Guttman 1998]:

- *Personalization*: agents can either learn or be explicitly taught what to do for each individual or group of users
- *Autonomy*: agents have some degree of decision-making capability. They are able to reason on their own and take decisions on behalf of the user. Of course the decisions made should be according to the user's preferences
- *Reactivity*: agents can perceive their environment and respond to external changes
- *Pro-activeness*: agents exhibit a goal-directed behavior by taking the initiative

- *Social ability*: agents communicate two-way with users and other software agents.
- *Persistence*: agents are continuously running processes
- *Trust*: agents are expected to accomplish tasks in a responsible manner

Software agents create interesting topics for research in both fields of Human-Computer Interaction and Artificial Intelligence [Lu 1999]. The general debate in agent studies is in agreeing on "which of the above mentioned characteristics are essential for a software system to be qualified as an agent". In addition, software agents are often referred to as "intelligent agents". Generally, agents are said to be intelligent if they incorporate some advanced behavior such as reasoning, learning or planning, usually by making use of a knowledge base and an inference engine. In addition to that, although some software agents have been anthropomorphized, anthropomorphism is not considered an essential requirement for a software agent [Desharnais 2000].

The application of agents can be in different domains. They can be used to filter information, match people with similar interest, automate repetitive behavior, etc. Lately a lot of research has been concentrated on the application of agents in the domain of e-commerce [Maes 1999].

There exist many categories of software agents. One of these categories that we are interested in, is known as the "Interface Agent"[1]. Interface agents are special kinds of agents that operate in a user interface environment. Their raison d'être is helping users in learning and effectively using the interface of a software. This is often done by personalizing the interface: filtering the information and displaying it in a manner that is most appropriate for the individual user. In order to do so, agents need information about the user. This information is what makes the "user model" or "user profile" as it was called by Ji Lu [Lu 1999]. The next section gives a brief overview of user models that are useful in B-to-C e-commerce context, and chapter 2 provides a general review of the literature on user models.

[1] Referred to also as "User Interface Agent", or "UIA".

1.3. Overview of User Models in B-to-C E-commerce Context

User modeling means incorporating certain knowledge about the user in a manner suitable for algorithmic processing. The goal is to give a better service based on this stored knowledge. The knowledge about the user can describe what the user "likes", or what the user "knows" [Chin 1986]. Modeling "what the user likes" can help us deduce what kind of information he is interested in. Modeling "what the user knows" can help us decide how to present this information.

In e-commerce, user modeling should help achieve these four tasks[2] [Lu 1999]:

1. Provide personalized service to a particular user by employing the user model
2. Present information in a way suitable to the user's needs
3. Provide proactive information and messages to assist the user
4. Disambiguate user's search input based on his user model

Existing user models in B-to-C e-commerce applications have thus far narrowly focused on the preferences of the user in one given domain. For example, using different techniques, the existing models can help predict if a person might prefer a certain book *A* to another book *B* [Amazon 2000]. In this case, the domain is narrow and it is about books. Our goal in this thesis is to be able to deal with multiple domains as well as fulfilling the four tasks mentioned above. The user model we propose, like other models can predict the preference of the user among similar items (e.g. books). But it can also predict that in some situations, the user is more interested in buying a TV set than a bicycle. Previously existing models are useful in specialized shops, like bookstores, or

[2] In fact this is achieved by the software that uses the data from the user model. An interface that integrates a user model is a UIA. We can use equally the terms "Intelligent Interface" and "Interface agents".

video stores. Our model is more useful in general departmental stores, like Wall Mart, or in virtual shopping malls where multiple vendors coexist.

1.4. Overview of the CITR Project

Established in 1990, the Canadian Institute for Telecommunication Research (CITR) is a federally incorporated, non-profit research institute and a member of the federal Networks of Centers of Excellence (NCE) Program. There are several major projects managed by CITR and "Enabling Technologies for Electronic Commerce" is one of them and the work done in this thesis is a partial contribution to this project. The objectives of this project are:

(1) to develop fundamental enabling technologies and infrastructure for e-commerce applications;
(2) to analyze the networks, computer systems and work-flow architectures supporting those applications;
(3) to develop architectures for distributed e-commerce applications;
(4) to model and understand the behaviors of those applications.

The focus of the research is the creation of a virtual mall. The key technologies under development are multimedia virtual catalogues, user interfaces and intelligent agents. Researchers are also exploring system and network management challenges, and performance issues [CITR1 2000].

The CITR project consists of the following five major components [CITR2 2001]:

- *Data Management Issues in Electronic Commerce*

  This includes, among others, the development of virtual catalogues, intelligent query across these catalogues, and support for the management of transactions between customers and vendors. These catalogues should have multimedia capabilities and should allow for the integration of different catalogues developed by each vendor. The objective of the project is to enable users to be able to access multiple, distributed, and potentially heterogeneous catalogues in a uniform and transparent manner.

- *Quality of Service and Distributed Systems Management*

  This project addresses the issues of the management of quality of service and the adaptation of the application within different system (from low end PCs to powerful workstations) and different network environments (from wireless link to high-speed attachment).

- *Security Issues*

  This project considers both the general security issues on the Internet (as they apply to electronic commerce) and the specific problems generated by this application.

- *System and Network Performance and Application Management*

  This project uses the deployed electronic commerce application as a test-bed to collect data and conduct measurements on application behavior.

- *User Interface and Intelligent Agents*

  This project investigates the design and integration in the major project of a user interface employing distributed virtual environment concepts for navigating the virtual electronic commerce mall, web-based tele-collaboration tools and also using appropriate intelligent agent technologies.

The e-commerce project of CITR involves nine professors and teams from seven universities in Canada. Both the University of Ottawa and Concordia University are responsible for the user interface and intelligent agents sub-project. Two challenging aspects of this work are: being able to integrate individual work within the global architecture of the main project (the virtual mall), and co-ordinating the work of different teams and individuals who meet each other only once or twice a year.

1.5. Objective and Scope of the Thesis

We started by analyzing the user's shopping behaviors. We analyzed the user's way of interaction with an online shopping system. Based on these findings, we tried to provide a user model that can capture the user's general "preferential tendencies". We broadened the domain of application of the user model (in contrast to narrow domains). We tried with a user friendly GUI to demonstrate that our user model can be useful in different ways of interaction with the system (browsing, querying, proactive notification).

In addition to describing the user model, we proposed ideas (and implemented some of them) for ways of acquiring and refining the model; ways to use it for personalization, and places to store it in a network taking care of privacy and security. The methodology used can be summarized with the five steps shown in table-1 below. In this thesis we address each of these steps in turn.

1) What are the HCI problems encountered by the user in the application?
2) What personalized service can we give the user to solve these problems?
3) What do we need to do it?
4) How do we do it if we have the required information?
5) How to acquire what we need?

Table 1. Methodology for designing the user model

1.6. Organization of the Thesis

Chapter 2 of the thesis provides a literature review of user models and mental models. Two classification schemes for user models are presented. These classification schemes serve to understand differences and commonalties among user models. This chapter also introduces three case studies of different user models. A critical evaluation of each of these user models follows each case study in order to understand the pros and cons for each approach.

Chapter 3 discusses the interaction problems and analyses the user behavior in e-commerce. A CBB (Consumer Buying Behavior) model is presented as a common framework for exploring user's shopping needs as well as the role of agents in electronic commerce applications. This is followed by a discussion concerning the usability of e-commerce sites. Three types of shopping behaviors are identified as well as the interaction problems associated with each of these shopping behaviors and suggestions to solutions.

Chapter 4 presents the proposed generic user model as a solution to the problems identified in chapter 3. The user model is described and examples are given. This chapter also explains how the user model can be helpful in different interaction modes and also presents a technique to acquire the user model.

Chapter 5 presents the global architecture of the CITR project. It also addresses the question of where to store the user model in such architecture. This question deals with security and privacy issues.

Chapter 6 describes the implementation of the prototype that incorporated the proposed user model. Apart from implementation issues, this chapter also discusses the objectives met (or to be met) by building a prototype.

Finally chapter 7 is the conclusion that summarizes the contribution of the thesis.

2. Review of Literature on User Models

The expectations of a user about a computer's behavior stem from mental models (figure-1); while the "expectations" a computer has of a user come from user models (figure-2). The fundamental distinction between them is that mental models are not explicit and they are inside the user's head while user models are representations inside a "computer". Thus, mental models can be modified only indirectly by training while user models can be examined and manipulated directly [Allen 1997].

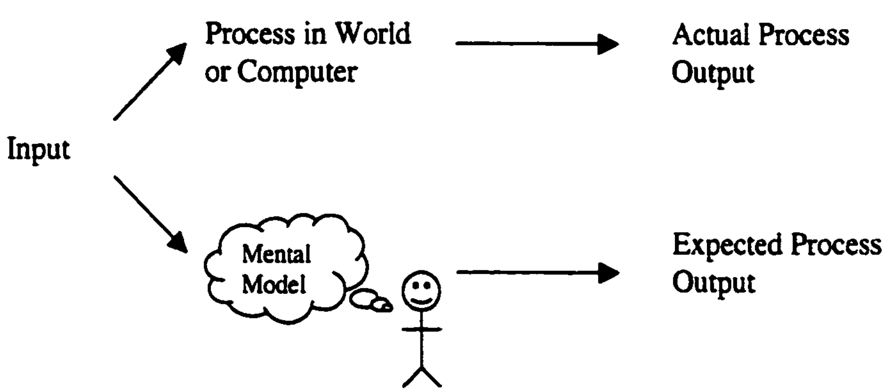


Figure 1. Process and mental model of that process

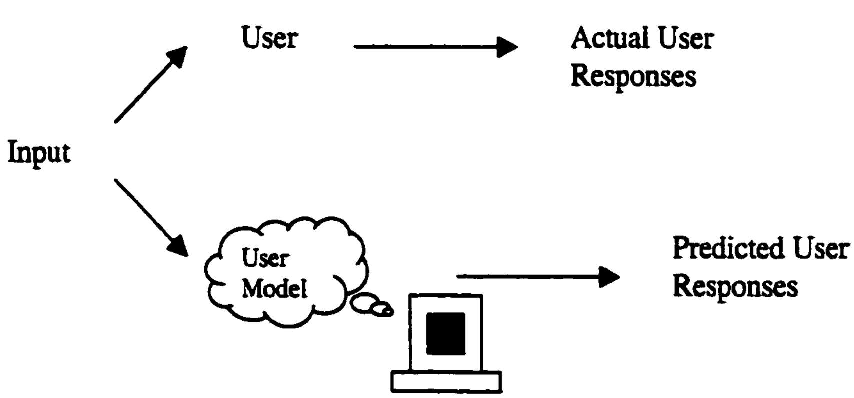


Figure 2. User responses and user model predicting those responses

2.1. Mental Models

The world in which we live is becoming more and more complex with regard to information management. In order to deal with this complexity, people create scalable mental models [Barker 2000]. Mental models are the conceptual and operational representations that humans develop while interacting with complex systems [Jonassen 2000]. The Scottish psychologist, Kenneth Craik, was the first to postulate mental models in 1943. He wrote that the mind constructs small-scale models of reality that it uses to anticipate events, to reason, and to create explanation [Johnson-Laird 2000]. Mental models provide a powerful tool for storing the knowledge in the human mind. People create them by putting together sets of perceived rules and patterns in a way that explains a situation. A typical person cannot draw or describe his or her mental models and in many situations the person is not even aware that these mental models exist [IBM 2001]. Yet mental models influence the behavior of people with regard to every aspect of social life. A mental model does not necessarily reflect a real situation and its components accurately. In fact, they are only the approximations that humans have in their mind of important aspects of the operation of a process or an object. The mental model synthesizes several steps of an actual process and organizes them as a unit, without necessarily representing all the steps of the process. That is why they are often incomplete and inconsistent [Allen 1997].

In order to create, improve and discuss about mental models, it is helpful to have "a model of" mental models. These models of mental models are called conceptual models. Conceptual models are categorized by [Allen 1997]:

- *Metaphors*, which are powerful verbal and semantic tools for conveying both superficial and deep similarities between familiar and novel situations [Mountford 1990].

- *Surrogates*, which are descriptions of the underlying mechanisms of a process [Allen 1997].

- *Mappings*, which are a way to describe the relationship between controls and their effects on a system [Baecker 1997].

- *Task-action grammar* (TAG), which is a formal notation to describe the knowledge that users need to have in order to perform tasks with a computer system or other complex devices [Howes 1991].

- *Planning models*, which describe computer-based tasks using plans. An example of a planning model is the widely known GOMS model. GOMS is a technique for modeling and describing human task performance. GOMS is an acronym that stands for Goals, Operators, Methods, and Selection Rules, the components of which are used as the building blocks for a GOMS model. Goals represent the goals that a user is trying to accomplish, usually specified in a hierarchical manner. Operators are the set of atomic-level operations with which a user composes a solution to a goal. Methods represent sequences of operators, grouped together to accomplish a single goal. Selection Rules are used to decide which method to use for solving a goal when several are applicable [Wood 1998].

- *Propositional knowledge*, which is conceptual model based on logic and predicates.

Since mental models are intangible and not directly observable, several types of evidence have been used to infer characteristics about mental models [Allen 1997]:

1. *Predictions*: humans can predict the next action that a system will take. They can predict as well, how a change in the system in one part of the system will be reflected in another part.

2. *Explanations and diagnosis*: explaining the cause of an event, or reasoning about the cause of a failure of the system, can also help deduce the mental model.

3. *Training*: mental model is also reflected in training, where we realize clearly that people who are trained to perform a task with a coherent conceptual model, usually do it better than people who never had any training with the model.

4. *Other*: Evidence about mental models, is also obtained from eye movement, time for answering questions about processes, or time taken to perform a task.

2.2. User Models

User models incorporate the knowledge about the user so as to increase, in a general sense, the quality of man-machine interaction. Several applications or agents can use this knowledge to give a better and personalized service to users. The personalization can be in the form of predictions of the user's behavior (behavior includes actions, reactions, preferences, needs, etc...). These model-based predictions will be better than random prediction.

Many schemes to classify user models have been suggested. Here, we will present two of them. The first one is the empirical quantitative model that contrasts with the analytical cognitive model. This scheme was proposed by Carbonnel [Brajnik 1987]. The second one is the three-dimensional-space classification proposed by Elaine Rich [Rich2 1983]. Classification schemes generally are useful because they help characterize the common features and peculiarities of each approach and clarify the usefulness of the user modeling activity.

2.2.1. Empirical Quantitative versus Analytical Cognitive

Carbonnel suggested two broad categories to classify user models. One is the empirical quantitative category of user models, and the other is the analytical cognitive category of user models. Empirical quantitative models are based on information that is derived from an abstract formalization of general classes of users. These models contain only surface knowledge about the user, and no internal reasoning takes place. The knowledge about the user is usually taken into consideration explicitly only during the design of the system, and is then hardwired into the system. The design is done based on a target class of typical users, without trying to adapt the design specifically to each user. Most conventional help systems follow this approach.

Analytical cognitive models, on the other hand, try to simulate the cognitive user processes that are taking place during permanent interaction with the system. These models incorporate an explicit representation of the user knowledge. The integration of a knowledge base that stores user modeling information allows for the consideration of specific traits of various users [Brajnik 1987].

2.2.2. Three-dimensional Classification

According to Elaine Rich, user models can be classified along three criteria that form a three-dimensional space. As she claimed, any user model will necessarily be situated somewhere in this three-dimensional space.

The first dimension is the canonical versus individual models (this dimension is also called the granularity [Lu 1999]). That is one single model of a typical user versus a collection of individual models of many users. These are two extremes and in between we can have user models that apply for groups of people or stereotypes [Rich2 1983]. Individual models provide better personalization than canonical model but are more difficult to implement. We can assume that any application has at least a canonical user model built-in. For example, a chess game has the assumption that the user will always make the best move to win. This assumption is the knowledge we have about the user and therefore this assumption constitutes the canonical user model for the chess game application. In the case of group or individual user models, we can't always have the model built in. We have to find a way to build the user model using some techniques. This leads us to the next dimension.

The second dimension of the space of user models is explicit versus implicit modeling [Rich2 1983][Allen 1997]. Explicit models are in systems that allow users to create their own profile of interest. In this approach, the users create explicitly their own environments within the system. Entries in an explicit user model are a kind of self-categorization and therefore they can have the inconvenience of being biased sometimes, because a person is not objective when it comes to categorizing himself. Besides, people sometimes do not have adequate knowledge about themselves [Nisbett 1977]. Another argument against this approach is the fact that people, especially casual users, might get annoyed if they have to stop and answer a large number of questions for the sake of model building before they can get on with whatever task they are trying to use the system for. The alternative and more sophisticated approach is to build and update the user model on the fly. That is, as long as the user interacts with the system, his model gets built, updated, and refined. Machine learning techniques are used here to infer facts about the user, based on his actions while using the system. In many cases, this approach is combined with a canonical model that gets refined to become an individual model at the end. New users are assigned a canonical model, that will become later a stereotypical model, and in the end an individual model.

The last dimension we consider in classifying user models is short-term versus long-term modeling. This dimension is also called the temporal extent [Lu 1999]. This has to do with the type of information in the model. Some information is relatively permanent while other information will change more often. For example the user's level of education is a long-term fact, whereas what the user needs to buy online is a short-term fact. Thus the model can be a short-term model or a long-term model. As Elaine Rich suggested [Rich2 1983], it is reasonable to demand that the amount of effort spent to decide on a particular fact about a user be roughly proportional to the amount of time that fact will be used. This means, that it may be important to spend a lot of time, spread over many sessions to form an accurate model of some permanent characteristics. Whereas, it is a waste of time spending a lot of energy inferring facts about the user that will change in a short time.

It is useful to point out some of the differences between user modeling techniques adopted in intelligent tutoring systems and those utilized in man-machine interfaces [Brajnik 1987]. In fact, the purpose of tutoring systems is to increase the knowledge level of the user in the short term and hence short-term modeling techniques are used. Whereas in man-machine interfaces, educating and training the user is only a secondary goal. The emphasis here is on performing specific domain dependant tasks. So, long term modeling techniques are used.

In both short-term and long-term modeling, a dynamic evolution of the system can be present. This evolution will aim at repetitively refining the model to adapt to the user and become more accurate irrespective of the possible change in the user himself.

2.3. Three Case Studies

In the remaining part of this chapter we will examine three different existing user models. The first one is the clique based model, (also known as collaborative filtering technique) [Alspector 1998]. We chose to present this model because it is widely used in e-commerce applications. It is an implicit, individual and long-term model. Even though it is based on ratings supplied by the user, we still consider the model to be implicit since the user is not directly manipulating his model.

The second one is the stereotype-based technique used in IR-NLI II system [Brajnik 1987]. First we present this model, then a global architecture of an information filtering system, and finally we examine how a user model can fit within this architecture. We also see how stereotypes and statistical analysis can be used to deduce facts about a person to produce an individual user model. Besides being individual, the model is also long-term and implicit.

And finally, the third one is the rule-based model used in Scribe [Rich2 1983] [Reid 1980]. This model belongs to another family of user models that are useful for online help or tutoring systems. This model is an implicit, individual, and short-term model. We chose to present it because it is rule-based.

These three cases are presented to give a general and practical idea about user models and their uses. By studying all three user models given here, we are exposed to different kinds of long-term or short-term model. All the models are individual and implicit type because we find that spending time to study canonical and explicit models is more straightforward and not very interesting. We also describe rule-based, stereotype-based and rating-based models, and consider different domains of applications. With the IR-NLI II, we see how we can fit a user model within an existing architecture of a system.

2.3.1. Clique-Based User Model for Selecting a Movie

Also known as collaborative filtering, this popular method of filtering presumes that if a user's stated preferences are similar to a group or a clique of others in some aspects, it will be similar in some other aspects also. Because, the Internet makes it easy to collect preference information for large groups of users, collaborative filtering has become the basis for many recommendation services. The following section presents the details of the technique in the context of an agent that recommends movies.

2.3.1.1. *The User Modeling Technique and Algorithm*

This model is used to filter information by predicting the rating a person might give to a movie. Specifically, this approach is based on the notion that the average rating of a clique of users is the best indicator of an individual's future rating. A set of users forms a clique if their ratings are closely related. Each user has a unique clique composed of other users whose ratings are similar. To measure the similarity between two users, researchers have used the Pearson correlation coefficient.

$$C = a \cdot b / \left(\sqrt{a \cdot a} \cdot \sqrt{b \cdot b} \right) \qquad (1)$$

Where a and b are two vectors of the ratings given by two users. The vector of size n contains n ratings for n movies. The two vectors a and b should have ratings for the same movies in order to be able to do the comparison. The values of C range from -1.0 to $+1.0$. A positive value for C indicates similarity. In order to decide whether a user belongs to a certain clique of another target user, we define C_{min} and S_{min}. C_{min} defines the minimum correlation required, and S_{min} defines a lower limit on the number of movies that a user must have seen (in common with the target user) and have rated those movies. In a typical implementation S_{min} was assigned a constant value of 10 and C_{min} was a positive variable such that the number of users in a clique was held constant at 40.

The movie recommendation algorithm for clique-based approach involves the following steps:

1. Initialize parameters C_{min} and S_{min}
2. Identify a clique for the target user
3. Estimate ratings for the movie that the target user wants to see by considering the ratings by the members of the clique who have already rated that movie

Two approaches can be used to come up with the predicted rating (which is the same as the rating of the clique). One is the simple arithmetic mean of the ratings of the members of the clique.

$$r(m)=\frac{\sum_{i=1}^{N} c_i(m)}{N} \tag{2}$$

Where $r(m)$ represents the clique rating of a particular movie m, N is the size of the clique, and $c_i(m)$ is the rating of movie m by the ith clique member.

The other strategy consists of assigning the rating of each clique member a weight corresponding to the average correlation between ratings of the clique member and the target user. The formula is:

$$r(m)=\sum_{i=1}^{N} w_i \cdot c_i(m) \tag{3}$$

Where $w_i(m)$ is the correlation between the clique member and the target user.

2.3.1.2. *Critical Evaluation of the Case Study*

The above method is one kind of collaborative filtering. It is known to be an efficient method for movies and novels. The main drawback of this method, is that it requires active participation by the users in model creation. A variant of this method does not rely on ratings and it is used in Amazon.com [Amazon 2000]. They track who buys what, and when a customer buys a book, Amazon.com can suggest other books based on what other books people who have purchased 'this' one, have bought. This requires no extra effort or ratings by the users, and is found helpful.

Another limitation to this method is that it is applicable only in specific domains. While it is very useful in bookstores, video stores and music stores, it can be of little help when we want to compare different kind of items, or general preferences. For example, the method cannot allow us to suggest to the user a restaurant since each person might prefer a restaurant for different reasons. Some people will like a restaurant for the food, while some others for the service, and some others for the ambiance, etc. In such cases, the rating of the clique will be a poor indicator of the individual's preference. In other words, the model does not take account of the features and the options associated to an object and which might influence the preferences.

2.3.2. Stereotype-based User Model of IR-NLI II

IR-NLI II (Information Retrieval – Natural Language Interface II) system is an expert interface that allows casual users to access online information and encompasses user modeling capabilities. The architecture of the systems is shown in figure-3. It has two major subsystems. The first one is the information retrieval expert subsystem, which takes care of conducting the search session, interacting with the user and interrogating the information storage system. The second subsystem is the user modeling subsystem, which carries out the user modeling activity. It extracts information relevant to user modeling from the dialogue between the user and the system, and it constructs and updates the user model.

The information retrieval subsystem comprises three modules and two knowledge bases. The modules are:

1) *The understanding and dialog module*, which interacts with the user understanding his input and generating questions.

2) *The reasoning module*, which refines the user input and formulates the query.

3) *The formalizer module*, which constructs the search strategy from the search formulation and connects to the information retrieval subsystem.

The user modeling subsystem comprises two modules: the model builder and the history manager. The history manager records details of sessions into a database called search session history. The model builder uses this search session history to produce and update the user model through statistical analysis, devoted to extracting the most common features and traits of each individual user. The model builder is connected to two other databases besides the sessions' history. These are the stereotype database and the user model database.

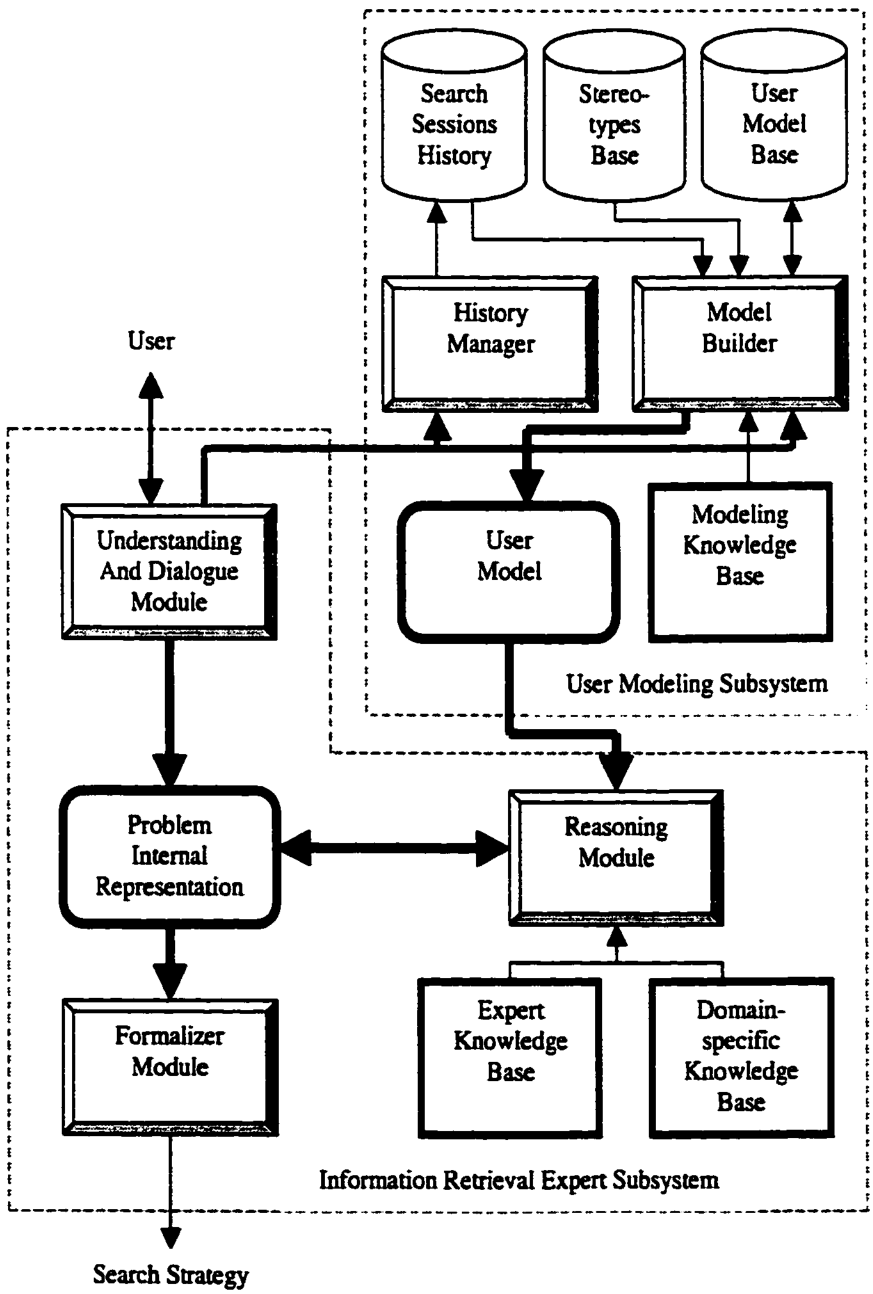


Figure 3. Architecture of IR-NLI II

The stereotype database contains stereotypes and is accessed whenever a user logs in for the first time to the system. With the stereotypes, the system tries to produce a first tentative and partial model for the user.

The database of user models contains the actual individual model for each user and is accessed once at the beginning of each session to retrieve the model parameters of the current user. At the end of the session, the possibly refined model is stored back to the database.

2.3.2.1. *Content and Usage of the User Model*

The user model is structured in the following way:

1. <user identification name> which identifies the user to whom it relates
2. <model history> which contains information about the history of the model such as creation date and successive refinements and updating
3. <user profile> which encompasses knowledge about specific features, attitudes and traits of an individual user
4. <user knowledge> which is devoted to storing information describing what the user knows about the environment of IR-NLI II operation

The understanding and reasoning modules use the information contained in the user model, mainly the user profile and the user knowledge, to perform the following tasks:

1. Tuning the user-system dialogue. This means tailoring the level and content of system generated utterances to each individual user.
2. Translating user's answers and utterances to suitable database queries.
3. Obtaining relevant information from the user model when that information cannot be acquired from the user.

2.3.2.2. Basic Operation of the Model Builder

At first, the model builder starts by identifying the user accessing the system and looking up his model in the user model database. If the user is new to the system, then no model exits and model building has to start from scratch. The process of building the user model is described in the five phases below [Brajnik 1987]:

Phase 1. *Preliminary interview.* This phase is devoted to obtaining basic information about the user.

Phase 2. *Stereotype activation.* Based on the information gathered in the previous phase, stereotypes will be selected. The stereotypes from the stereotypes-base whose activation method is satisfied become active and might be considered later as starting points for building the individual model of the current user.

Phase 3. *Stereotype discrimination.* In this step the model builder identifies one stereotype among the active stereotypes to be used as the kernel of the model construction.

Phase 4. *Model refinement.* In this phase the model builder extends and refines the individual user model during a single search session. Refinement starts on the discriminated stereotype identified in the previous phase and is achieved through information acquisition and information validation. The model refinement phase ends with the end of the session.

Phase 5. *Closing operations.* At the end of the search session the model history slot is updated and the model is stored back to the user model database. The history manager also writes a summary of the search session to the search sessions history.

In the case where the user is already known to the system, the operation of the model builder is slightly different. And it is described in this case in the four steps below.

Step 1. *Model retrieval.* After identifying the current user, his model is retrieved from the database and made available to the system.

Step 2. *Historical information processing.* In this step statistical processing of the summaries of the past search sessions is done in order to identify meaningful patterns denoting specific features of the user. This aims at refining typical values of some slots of the model.

Step 3. *Model refinement.* See phase 4 above.

Step 4. *Closing operations.* See phase 5 above.

2.3.2.3. Critical Evaluation of the Case Study

The architecture and the high level operations of the IR-NLI II system are considered to be a good design, since each module has a well defined role and since the user model is retrieved once and updated every session. Besides, the content of the user model is well organized since the user's knowledge (what the user knows) and his preferences (what the user likes) are well separated. These two types of information usually serve different purposes in such systems. The user's knowledge helps deciding in which format and language should be used to display the information. The user's preferences help deciding what information should be shown.

However, the user modeling technique presented in this case study and used in the IR-NLI II system has one weakness. It is based on the assumption that whenever the model builder has to make a critical decision, all the information necessary for this decision is available. For example, whenever the choice is to be made between alternative values for some slots, like the one provided by the user and the one inferred by the system. Unfortunately, most of the time, necessary information for the model builder is lacking at the crucial decision moment. Furthermore, it is impossible to undo most of the actions performed by the model builder [Brajnik 1987].

Also, the model as described does not take full advantage of the stereotypes, in the sense that a user is associated to only one stereotype, which serves as the backbone of his individual model. All the facts inferred later about the user, are individual facts that will refine his model. Associating the user to a stereotype is an efficient way to deduce several facts at a time. The system Grundy has the possibility to activate many stereotypes to a single user, taking maximum advantage of the stereotype database [Rich1 1979]. In the case of Grundy, the user model is a combination of all activated stereotypes.

2.3.3. Scribe Rule-based Model

Scribe is a document formatting system that employs a rule-based user modeling technique to personalize the interaction of its online help facility. The model is built by inferring individual facts about the user, which is unlike stereotypes-based models where multiple facts are inferred at the same time. The system learns on the fly, facts about the user by looking at the way the user is interacting with the system. The knowledge about the user in the model can help in producing messages at the right level of description.

One way to do so is, depending on the sophistication of the commands utilized by the user, the system can learn information that is associated with the individual's expertise with the system. This is done by constructing a dictionary of system commands, options, etc. and associating with each item, an indication of what information the use of that item provides about its user.

An alternative way to infer facts is via patterns of the user's commands. For example, when the user gets a response from the system to a question that he has asked, he might either be satisfied with the response or not. If he is, he will probably leave or proceed with another task. If he is not, then he is likely to restructure his request and make another attempt to get what he wanted. This second attempt should signal to the system that the user was not satisfied with the first response.

To illustrate this further, we describe briefly the operations of the online help facility of Scribe.

2.3.3.1. The Rule-based Modeling Technique

Knowledge is stored in the system as a set of rules. A rule is a "condition-action" pair. A condition is usually a command plus certain values of some internal system variables. This means that depending on the values of these variables, we can have many rules describing the same command. This description is hierarchical. Actions are often some other Scribe commands that will trigger some other rules. The actions of the rules will often set the system variables, which will later affect the operation of other rules.

Because of the hierarchical organization of the knowledge, the system can answer a question at different levels of detail. For example, if the system is asked a "why" question, it can answer by simply stating the conditions in the rule that led to this action, or it can chain back through the rules to determine how these conditions came to be true. To decide at which level to answer a question, the system looks at the level of the question itself and at the level of knowledge possessed by the user. To do that, the system maintains a dictionary of entries of all the "concepts" that are found in a rule (commands and variables). With each concept, a rating is associated denoting how expert a person would have to be in Scribe, in order to understand an explanation in terms of it. Thus, many of the internal system variables have very high ratings while the simple user commands have very low ratings. Each concept also has another rating that describes the level of sophistication of the user with computer systems to understand it. For example, some concepts, like block structures, are easily understood by a programmer, even if he were a novice Scribe user, whereas even a more Scribe experienced non-programmer might not understand it.

So when the system tries to answer a question, it first finds the appropriate rule (or rules) that apply to the particular situation, and then it looks at the concepts mentioned in those rules and see if their ratings match the rating of the user. If they do, then it generates the answer immediately, if they don't, then it chains up or down through the hierarchy of rules until it finds an explanation at the correct level.

To construct and update the user model, the same dictionary of concepts is used. When a user asks a question for the first time, he formulates it in terms of Scribe actions, Scribe commands and Scribe variables. The system then uses the dictionary of concepts to evaluate the user's level of expertise and the first model is built. This model of the user's level of expertise can be raised as the user asks other questions.

As stated above, the model can also be updated via patterns of user's questions. If the user did not understand the response to a question, then he is likely to ask the same question again in another format, or ask another question to clarify some concepts that he did not understand in the original response. This should signal to the system that it misjudged the user's knowledge in the first place. The system will then try to correct the model when it discovers the level of explanation with which the user is satisfied. In the same way if the system underestimates the user's level of expertise, it will be giving him general and broad answers. If the user is not satisfied, he will be asking for specific information. Such patterns should help the system update the model.

2.3.3.2. Critical Evaluation of the Case Study

This kind of user modeling technique is very simple and can be used mainly in applications where we need to model the user's knowledge (e.g. online-help or tutoring systems). Its weakness in those cases is that it makes the assumption that there is a fixed order in which people learn things about a system. Most of the times this is not true.

Applying this method to systems where we need to model the user's preferences is more difficult to imagine (e.g. e-commerce applications, or information filtering agents). In fact, to apply such a technique, we need to have an ordered hierarchy, where the user model of a given user has to be somewhere along this hierarchy.

2.4. Summary of the Chapter

This chapter provided a review of the literature on user models. We presented mental models and what are user models. A mental model is the model that a person has in his head about a complex object or process. A user model is the model that is incorporated in the machine about the understanding of the user's behavior. Mental models can in fact help in analyzing the specifications for user models.

We also saw in this chapter how to classify user models according to two suggested schemes. The first one is the empirical quantitative versus analytical cognitive. The second one is the three-dimensional classification: canonical versus individual, explicit versus implicit, and long-term versus short-term.

Three different user models were presented. We looked first at the widely used clique-based technique useful for narrow domains such as books and movies suggesting agents. We saw also the IR-NLI II information retrieval system. The user modeling technique used in this system is based on stereotypes. Such techniques are successfully used in information filtering systems. By studying this system, we were also introduced to its architecture, thus learning how a user model can fit within the global architecture of a system (information retrieval, and natural language interface in this case). We finally examined the rule-based user modeling technique employed in Scribe and which is used in the context of online help and tutoring systems. By looking closely at these three case studies, we realize the pros and cons for each of the above mentioned techniques.

3. Interaction Problems and Analysis of User Behavior in E-commerce

3.1. The CBB Model

Business to consumer type e-commerce encompasses a wide set of user-centered issues such as dealing with intermediaries between the producer and the consumer, ensuring security in the payment mechanism, providing a good after-sales service, emulating the pleasurable aspects of shopping, and catering to the needs of the wide variety of end users or consumers [Lu 1999].

There exist many models that attempt to capture the buying behavior of consumers in the marketing and psychology literature. Based on this previous literature, Guttman et. al proposed a Consumer Buying Behavior model (CBB model) that captures the decision processes that people undergo when purchasing a product [Ripper 2000]. In addition to that, the model defines a common framework to explore users' needs as well as the role of agents in e-commerce. The CBB model is a descriptive model and comprises the six fundamental stages [Roeland 1999] given below:

1. *Need Identification.* This stage characterizes the consumer becoming aware of some unmet need. Within this stage, the consumer can be stimulated through product information.

2. *Product Brokering.* This stage comprises the retrieval of information to help determine what to buy. This encompasses the evaluation of product alternatives based on consumer-provided criteria. The result of this stage is called the "consideration set" of products.

3. *Merchant Brokering.* This stage combines the "consideration set" from the previous stage with merchant-specific information to help determine who to buy from. This includes the evaluation of merchant alternatives based on consumer-selected criteria (e.g., price, warranty, availability, delivery time, reputation, etc).

4. *Negotiation.* This stage is about how to determine the terms of the transaction. Negotiation varies in duration and complexity depending on the market. In traditional retail markets, prices and other aspects of the transaction are often fixed leaving no room for negotiation. In other markets (e.g., stocks, automobile, fine art, local markets, etc.), the negotiation of price or other aspects of the deal are integral to product and merchant brokering.

5. *Purchase and Delivery.* The purchase and delivery of a product can either signal the termination of the negotiation stage or occur sometime afterwards (in either order). In some cases, the available payment options (e.g., cash only) or delivery options may influence product and merchant brokering.

6. *Service and Evaluation.* This post-purchase stage involves product service, customer service, and an evaluation of the satisfaction of the overall buying experience and decision. The nature of this stage, as well as other stages depends upon for whom the product was purchased, and on the type of the product.

In this thesis, we focus our interest on the first two stages of the CBB model. We try to tackle usability issues and interaction problems in the need identification and product brokering stages. We suggest ways to improve the usability and to solve the interaction problems through personalization, which is based on user modeling.

3.2. Usability of E-commerce Sites

Usability is a prerequisite for e-commerce success. If people cannot shop, then the site won't sell a thing. It doesn't matter how cheap the products are if people can't find them [Nielsen1 2001]. Shopping on the Internet is not very easy. A survey showed that 8% of the people interviewed don't shop online because the sites are too difficult to use [Herschlag 1998]. This reveals serious usability problems with e-commerce sites, which are no less than usability problems of web sites in general. For the purpose of making useful analogies, let us compare physical shopping to online shopping.

Since it is easier going from one site to another on the web than going from one physical shop to another, the merchant brokering becomes easier with online shopping. The same cannot always be said about product brokering (price, features, and options comparison) since products are not always easy to find on a web site. Once found, it is not always easy finding the relevant information that is of interest concerning the product. Part of the reason for this is that there is no consistency in displaying the information related to a product.

In traditional shopping, companies spend some energy and money on the physical store design, and on creating a positive environment and experience for the customer (storefront). Since the environment is limited to the computer screen in electronic shopping, companies should try to find alternative ways to create the storefront. Three things become important: product perception, shopping experience and customer service [Rohn 1998]. The use of sales agents, multimedia and virtual reality can help achieve these goals to some extent. A good and successful example of the use of sales agents combined with virtual reality is the Virtual Mannequin developed for San Francisco boutique [BSF 2001]. The site allows the user to create a three-dimensional mannequin that will have her physical characteristics: dimensions, shape, hair color, hairstyle, skin color... etc., as chosen by the user to mimic herself. After creating the mannequin, the site[3] will suggest to the customer, fashion clothes based on her appearance. The user is able to try many clothes without having to leave home (figure-4).

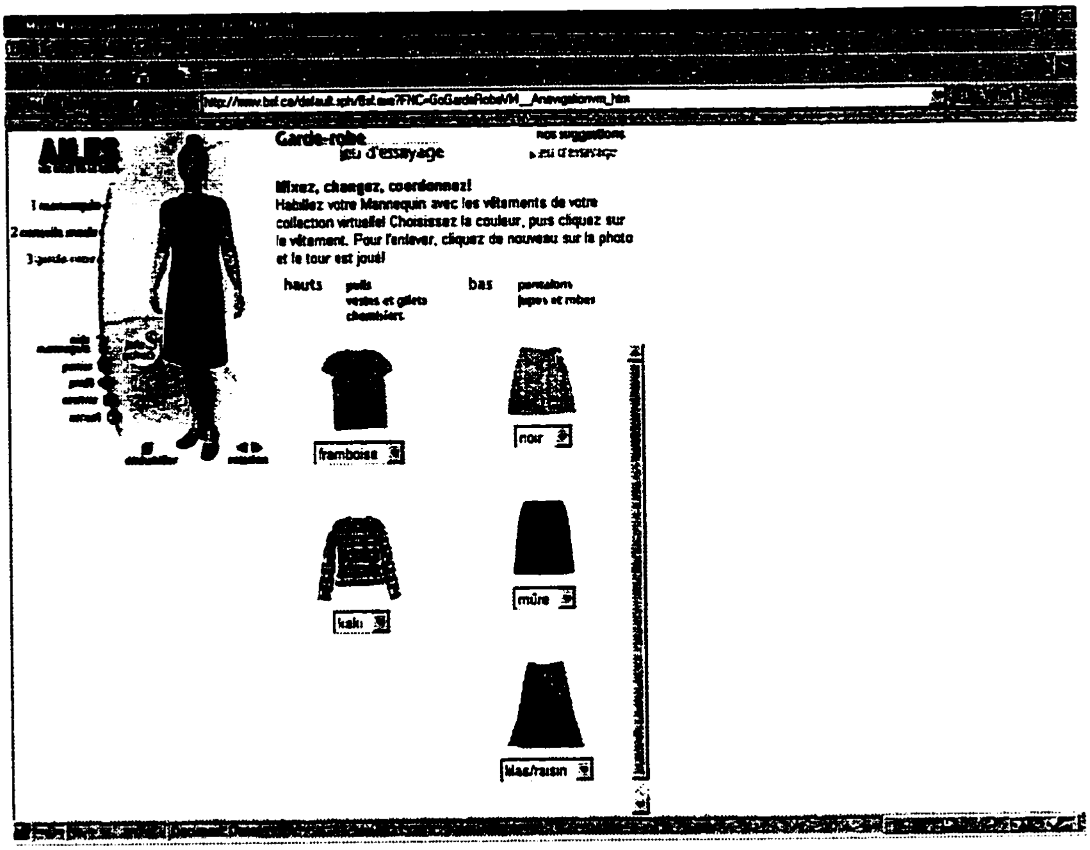


Figure 4. The Virtual Mannequin

Another example about the use of virtual reality combined with agents and multimedia, is the demo we[4] developed as part of the CITR project and that was presented in CASCON 1999 and CASCON 2000 [Georganas 2000]. In this prototype a Java applet allowed the user to query a database for some items. Before adding an item to his electronic shopping cart, the user is given the possibility to view this item in a 3-D VRML representation, that reflected the reality as close as possible, or to see a video demonstration of the item (figure-5, figure-6). After adding the item to his shopping cart, and before the final purchase, the user can negotiate the price with an automated sales agent [Desharnais 2000].

---

[3] In fact, this is done by a software agent expert in fashion.

[4] Rony Abi-Aad, Patrick Desharnais, Xiaojun Shen and Paul Iglinski.

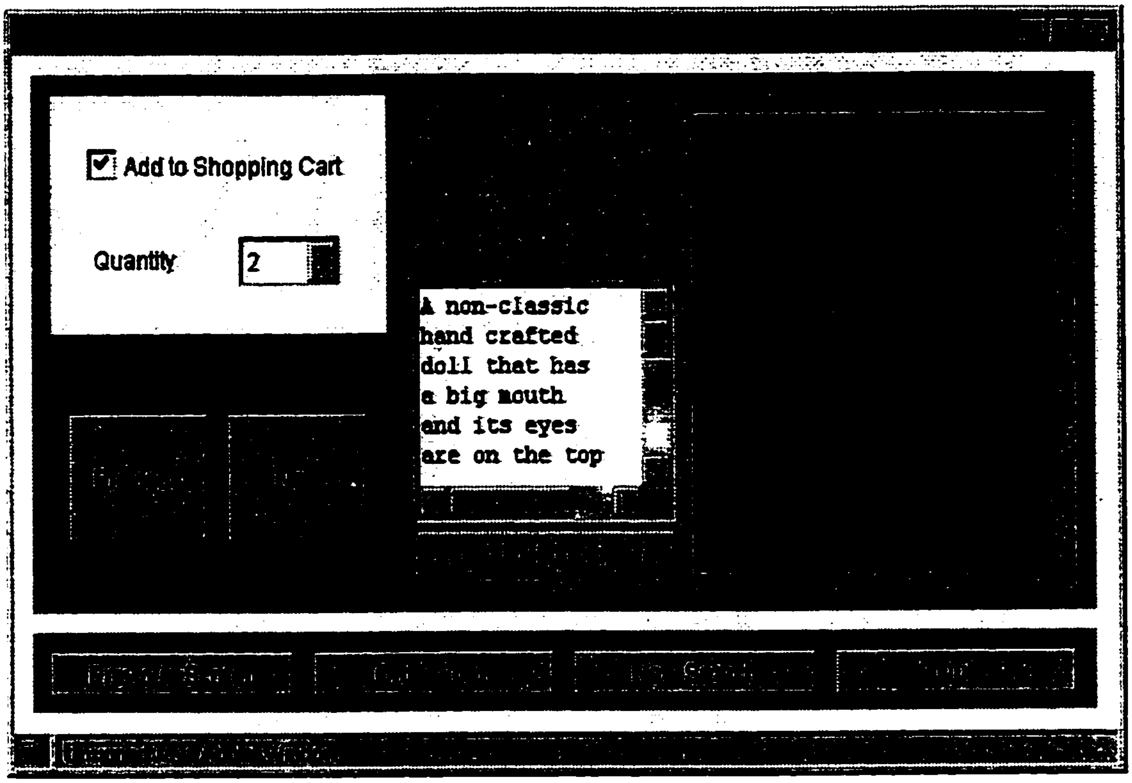


Figure 5. Result of the Query

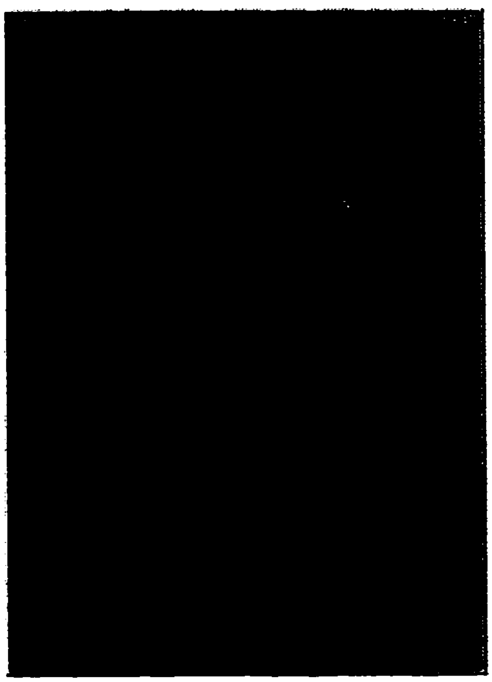


Figure 6. 3-D Representation of a Toy

The raison d'être of a salesman in a shop is to give service and advice for the customer. This is always given in a personalized way. For example in a city like Montreal, all salesmen speak French and English because people in Montreal are either French speaking or English speaking. On the Internet, shops that are targeting a global clientele should support multiple languages since the customers can be anywhere in the world. Those shops should also support multiple currencies [Nielsen2 2001]. Many sites are unable to sell outside the US because they offer the information only in English [Herschlag 1998].

Big companies spend substantial amounts of money on branding, with the purpose of associating a product or a logo with a positive emotional experience, thus creating customer loyalty. Unlike passive forms of communication such as print and television advertising, positive branding experiences with web sites are dependent on the site being usable. When someone has a negative experience with a web site, being unable to find a product or navigate through the site, they associate that negative experience with the brand. First-hand experience is much more powerful in determining whether a customer will remain loyal to a brand, and no amount of marketing can overcome a negative experience such as being unable to use or find information on a web site [Rohn 1998].

To focus on usability, we should decide who is the population of users; we should understand what the needs, goals, tasks, and priorities of the user are.

In an earlier research in our team, Ji Lu [Lu 1999] identified three different shopping behaviors: "comparative shopping", "planned shopping" and "browsing-based shopping". The user model we propose in this thesis, is beneficial in these three types of shopping behaviors. The following sub-sections, are analysis of the interaction problems and the ways to solve these problems through personalization in the three cases of shopping.

3.3. Comparative Shopping

In the case of comparative shopping, the user knows what products or product category he wants. He probably will compare similar products to find out which one is the best from his perspective. The simple scenario could be that the user queries the database to find products of a particular category like toys, clothes, kitchenware, etc. He may get many matches of what he looked for. He will then choose the one that satisfies the most of his need. The user's task in this case is the selection of one item out of many. The user model can assist the user in the selection process. Filtering the results of the query posed to the database is a possible task that could be automated. Many techniques exist for information filtering. Clique-based filtering, also known as collaborative filtering[5] is a popular method, but it relies on ratings given by users, so it requires active participation from the user. Feature-based filtering technique relies on the information that the user prefers certain options in the product category to other options [Alspector 1998]. Let us suppose the user is looking for a motorcycle and we know from his profile that he prefers Honda products, then we can eliminate all other motorcycles from other manufacturers.

Another assistance that the user can be provided is in refining his "tastes" based on extra information provided by the system or the user in a dialog mode. This we call "step by step spiralling in" to one item that matches the user's needs. For example, the user might always prefer Honda motorcycles, but when he realizes that Honda motorcycles have one year warranty whereas Suzuki motorcycles have three years, he may then start preferring Suzuki motorcycles, in other words his preferences are not static but information-dependent.

[5] Examined earlier in chapter 2 of this thesis.

3.4. Planned Shopping

In the case of planned shopping, the user plans to buy something in the future. He may have a "rough idea" of what he wants. Consider the following example that shows the difference between comparative shopping and planned shopping:

> **Planned shopping:** Miss *X* needs her car to go to work everyday. Her car is old and it is O.K. for the summer time. For the coming winter, she plans to buy a new car. This is an example of planned shopping and there is "adequate time" to shop.
>
> **Comparative shopping:** Unfortunately, one day *X* meets with an accident while driving and her car becomes unusable. She needs to buy the car "right away". She shops around for a good deal. The time available to perform the task of shopping is considerably smaller in this case than in planned shopping.

In the case of planned shopping, *X* knows that she wants a car. She probably has not decided yet what kind of cars she wants. She considers certain alternatives but her choice is not made yet. She wants to collect more information and wait before making a decision. Having more time is usually a good thing but that can also result in loosing a good opportunity or a special sale.

In the case of planned shopping, we need to do almost the same thing as in the case of comparative shopping, namely "narrow down". The filtering should be more flexible than in the case of comparative shopping because the user has less focus on the product type. So the filtering can be either strictly narrowing from generic to specific, or oscillating (expand to be generic or contract to be specific). Also, in this kind of shopping we need to identify, in a personalized way, when the opportunity is good and when it is not, for the user. For example, we may suggest to a user who follows the fashion to shop for winter clothes in September, before the beginning of the winter, because that's the best time for him to find nice fashionable selections. However, for a person to whom fashion is not a concern and who would like to get good prices on clothing, we might suggest to shop for winter clothes in January. Proactive notification can be used in this case.

3.5. Browsing-based Shopping

This denotes shopping with a casual objective (window-shopping). The main objective of the user here is not merely shopping. It may be information gathering or socializing. When someone goes to a mall, he is not alone walking around. He may talk to people and look for some products or watch the storefronts, switching between activities freely. If he finds something interesting, he may ask his friends for their opinion before purchasing it, thus seeking social acceptance. He may find another person with a product in her shopping cart and ask her some information about the product because it looks interesting for him.

In electronic shopping "chat rooms" are provided to address some of the issues in virtual communities [Georganas 2000]. In the demo we developed that was presented in CASCON 1999 and CASCON 2000 as part of the CITR project, we provided chat facilities and user-to-user interaction possibilities in the virtual mall[6] (figure-7). But we should also allow easy switching of modes from the chat room to browsing. This shifting of focus could be based on the "context". For example when a person in a chat room encounters an interesting new product, it becomes a "trigger" and he may decide to leave the room and go look at this product. One major problem here is to be able to personalize this shifting of focus or determine what constitutes a trigger. In real life we can detect the degree of enthusiasm and interest by eye tracking, speech variations, speech cues and facial expressions. In the context of computers, these methods are currently being explored [Mann 1996].

[6] Apart from the chat facility the user-to-user interaction was possible through body language: waving a hand, smiling, shaking the head, etc.

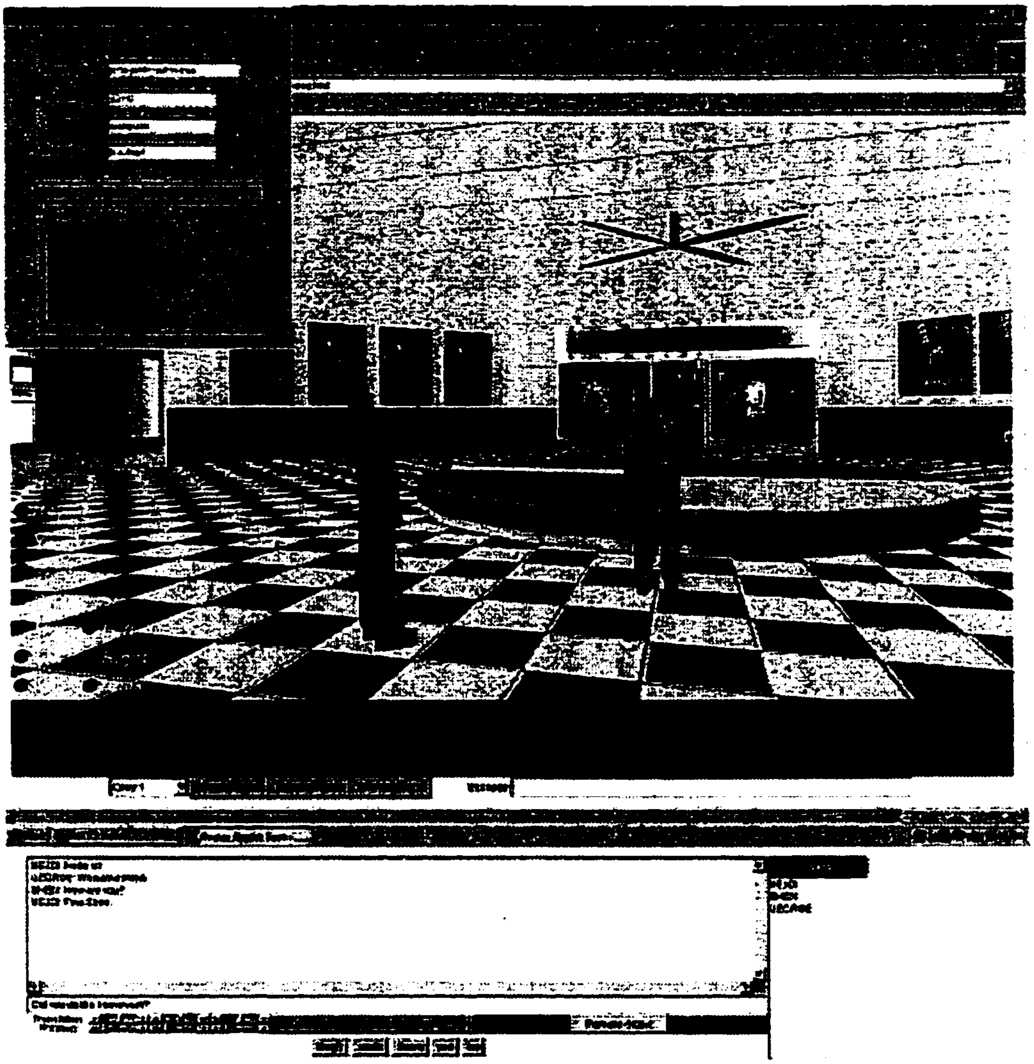


Figure 7. The shared environment of the Virtual Mall

Another problem in the case of browsing based shopping is the limited space of the computer screen. Since the user is looking for a panoramic view, one cannot show all the details on the screen. Thus, we have to be careful in making use of the screen space to achieve a good degree of user satisfaction in browsing.

4. Proposed Generic User Model

4.1. Introduction

The user model we propose, according to the classification described in chapter 2, is a long term, individual, implicit user model that gets refined with time. On the first login, a first version of the user model is built based on the registration form that the user fills in. Later, based on the interaction with the user, the model is completed, corrected and refined.

The user model we propose is suitable for e-commerce applications. It is referred to as a generic[7] user model for three main reasons. First, it is useful in generic types of commerce, like a multi-commodity store or a departmental store (e.g. Wall Mart). It can be used also in a Virtual Mall where all the vendors share one database. Existing e-commerce user models have so far been focused on specific domains of application, like bookshops, music shops or toyshops [Amazon 2000] [Alspector 1998] [Lu 1999]. The second reason is that it can be of assistance to the user in all three types of behavior we have discussed in the previous section: comparative shopping, planned shopping, and window shopping. The third reason is that our model is useful in three types of user-machine interaction: querying, browsing and proactive notification. These three modes of interaction are discussed later in section 4.3 of the thesis.

[7] The term generic is relative. The model proposed in this thesis is applicable in a broader scope than the models used in current electronic commerce applications

4.2. Content and Structure of the User Model

From the analysis about shopping behaviors in the previous chapter, we deduce that basically we need to offer two services to the user: either filter a large amount of information to a more specific set of useful information, or provide extra useful information when it is missing. So our user model should be able to capture the "preferences" of the user, in order to guide in the decision of what is considered useful information and what is not.

Preferences of the user are classified into his "tastes" and "needs". Tastes are broader and generic whereas needs are relatively narrower and specific. In buying clothes, a particular user may have a taste for expensive and brand name clothes and in a specific shopping he might have a need to buy a "Tommy Hilfiger" shirt. Tastes and needs are not independent from each other and they can influence each other. The tastes can become needs, and the needs can modify the tastes. The relationships between them and the transitions are shown in figure-8 and deserve to be studied further.

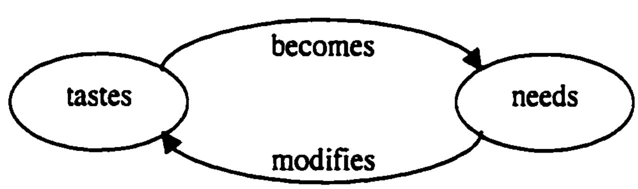


Figure 8. Relation between the user's tastes and needs

Figure-9 shows the abstract general block diagram of the system. The user model consists of user's tastes and needs and is enclosed into the dashed rectangle. The actions that should be provided by the system are filtering the information or giving extra information that the user has not asked for. The triggering process is the focus of the research and is discussed later in this thesis. In short, the triggering process should:

- Decide when to give less information and when to give more information based on external events and on the content of the user model.
- Update the user model, based on events.

The events are also discussed later in the thesis and they consist of actions performed by the user, changes in the database, or changes in the environment.

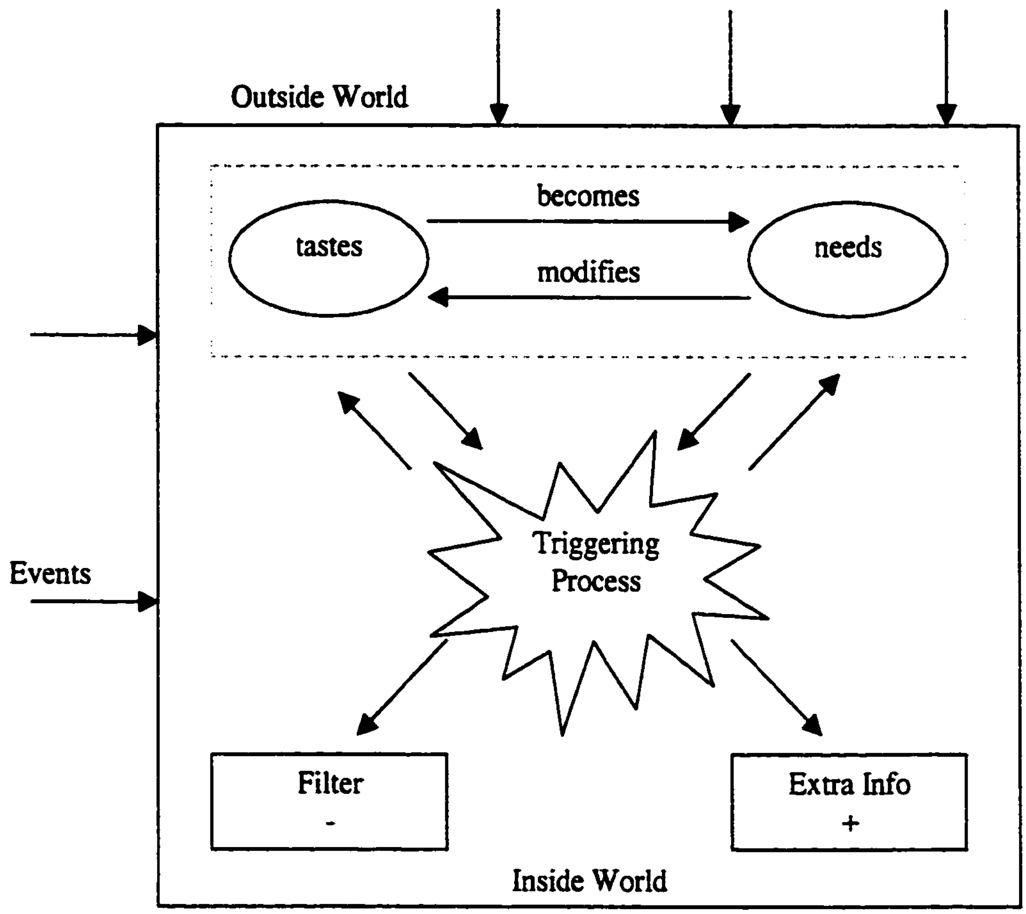


Figure 9. General Block Diagram

In order to capture the preferences of the user in a user model, we decided on including three main types of information in the user model:

- The kind of products that the user likes. This knowledge can help filtering information, or showing extra information on a general level. For example to predict if that the user is more interested in buying a TV set than a bicycle. These are described in section 4.2.1 about PIEs (Preference Indication by Example)[8]

[8] A term introduced in this thesis.

- The options of various products and the features of those products. This knowledge can help compare items and predict user's interests on a more specific level. Like predicting what kind of TVs the user wants. Features are associated to products and are described in section 4.2.2.
- And any "additional information" about the user concerning these products, such as the reason for the user's interest in the product or his expertise in the domain. This knowledge about the user can help decide how to present the information. And it can also help detect when an opportunity is interesting for the user. The user centred "additional information" is also domain dependent, therefore it is associated to products. It is described in section 4.2.3.

In the next four sub-sections, we will describe each of these three components of the user model, and the relationships among these three types of information.

4.2.1. PIEs (Preference Indication by Example)

Elaine Rich in her early work [Rich1 1979] on the Grundy system represented the user's interest in novels by means of a user model. She represented user's "personality characteristics" as a list of facets where each facet has a value and a rating associated to it. For example: <seriousness, 5, 800>, where 5 is the value given to seriousness indicating how serious the person is, and 800 (maximum value is 1000) is the rating indicating the degree of certitude of the information. We follow a similar approach and we refer to our representation as "preference indication by example" (PIE). For example, consider the triplet <car, 5, 700>. This is an abstracted representation that implies that this user is interested in cars, and <5> indicates the degree to which he is interested in this item. We use a 5-point scale where the value 1 stands for the lowest, the value 5 stands for the highest and the value 0 denotes indifference. The second integer 700 indicates how certain we are of this information. In the case of Elaine Rich's Grundy, the filtering of the information was semantic based, and therefore the value associated to a facet ranged from –5 to 5, because sometimes she had to censure some information that might be irrelevant to the user. In the case of e-commerce, we simply need to know whether the user is interested in the products or not. If he is not interested, we simply don't show him the products. Therefore, we use a range of 0 to 5.

4.2.2. Features

Features are the properties associated with certain products that the user likes to buy. For example, when buying clothes, a person may always prefer wrinkle-free material because he doesn't have time to iron. Thus, when showing clothes in an electronic shop, we will show him only material satisfying this property. In our model, features are represented by giving them values. So the features should be for example: 'color = red', 'height = 5 inches', 'wrinkle-free = yes'. Also features can be compared to values like 'height > 5 inches'. Or features are prohibited to have a certain specific value: 'color ≠ red'. We can sometimes assign approximate values to a feature, for example 'length ≅ 200 pages' for books. This indicates that a person usually buys books (novels) that have about 200 pages as length. Features are associated to the products and therefore, we associate a list of features with each PIE.

4.2.3. Additional Information about the User

The PIEs and the features are both about the user's interests in the products. There is another kind of knowledge that can help us in personalization. For example, knowing that a user is a comic book collector can help us identify that he will need missing items from his collection. When one wants to view a car, we can display details about the car engine to someone who is an expert in cars, and only show external details such as color, furniture, etc. to a person who doesn't know much about mechanics. Unlike the features that are product oriented, this information is user oriented similar to the traits of Elaine Rich. We call it "user additional information"[9].

4.2.4. Organization of the Data

These three types of information (i.e. PIEs, features and user addition information) are not independent. In fact, to each category of products there can be associated features and relevant domain dependent additional information. So after associating the list of features and the "additional information" to the PIE, the structure looks like:

Category	Value	Features List	Additional Info

Here are some examples of PIEs:

Car	5	Motor = good Color = blue, red Color ≠ black Mileage < 120 Brand = Honda	Expert

Comics	4	Writer = Dennis O'Neal Book = Batman, Detective Brand = DC Used = Yes	Collector

Comics	5	Book = Spiderman Used = Yes	Casual Reader

The user model we propose is in fact a collection of PIEs, just like the user model in Grundy is a collection of facets [Rich1 1979]. But, unlike the facets in Grundy, the PIEs

[9] This user additional information is the information related to each type of products as explained. It has nothing to do with the user's address, phone number, date of birth, etc. This other type of information is included also as part of any user model usually and we refer to it as "user personal information".

of a user form a certain hierarchy among themselves that is best represented as a DAG (Directed Acyclic Graph). It is not a tree because some products can belong to more than one category. The DAG (see figure-9) starts with general categories and refines downwards to sub-categories. We can take advantage of this hierarchical structure in two ways:

First, while building and refining the user model, this hierarchy can help propagate information up-and-down the DAG and thus vary the relative notions of generic and specific. The information (values, features and additional-info) can be inherited from general categories to their sub-categories. The information can be overridden as well in specific categories to correct inaccuracies inherited by general categories.

Second, since the user's preferences might vary in precision, this hierarchy can help filter/give extra information on different levels, thus adding flexibility to the concept of information filtering. It is this kind of flexibility that is needed to fulfill the need of different types of shopping behaviors. For example, consider the following PIEs hierarchy: "store → department → category → shelf → item". Each of these levels is in fact a sub-category of its parent category. When we are relatively sure of a "category" of product that the user would be interested in, then we compare PIEs at the following lower level in the hierarchy and that is the "shelf" level, in order to filter the information. For example, this is done if we know that the user is looking for a summer sport item, that can range from a bicycle to a roller blade (e.g. during planned shopping, or browsing based-shopping). On the other hand, if the user goes to the multi-commodity store (e.g. Wall Mart) and we are not sure which department within the store he is heading for, then PIEs comparison is done at the department level. This is done to predict if the customer is heading towards the computers department or the stationary department (e. g. during browsing-based shopping). At the lowest level in the hierarchy, if the user is hesitating between similar products, then features can be used to compare different items. This is the case when a customer is hesitating between two roller-blades that have minor differences in the possible features and options (e. g. during comparative shopping).

This is the kind of flexibility in information filtering that was mentioned in chapter 3. Usually, during comparative shopping, item-wise comparison is made at a very low level based on the features. During planned shopping, comparison is made at an intermediate level based on the PIEs. And during browsing-based shopping, comparison based on the PIEs is made at high and low levels, switching appropriately between these levels.

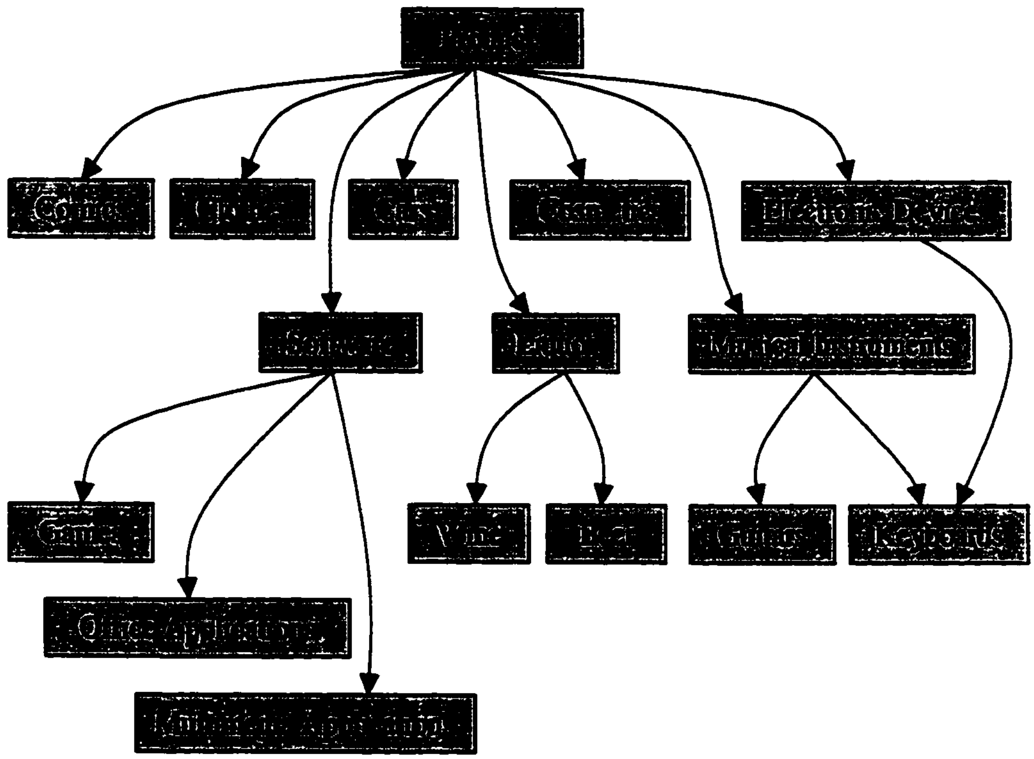


Figure 10. Hierarchy of the PIEs

4.2.5. Example of a User Model

The figure-9 shows part of a DAG representing the hierarchy among the products, their categories, and sub-categories. This same DAG can represent an "empty" model for a user. Let's say we don't have any knowledge about the user, then all the nodes in the DAG are turned "off" (colored gray). This means that the value of the PIE is zero. And zero denotes the indifference or the lack of information. Whenever we learn a fact about the user, this fact will be a trigger that will turn "on" the corresponding node in that user model (colored white). This means that the PIE will have a value associated to it that is between 1 and 5.

For example, if we are given or we have deduced about the user that he is a musician interested in New Age music, prefers Yamaha instruments, interested in electronics, lives in Europe and likes to have imported alcohol regularly, then we turn on the nodes "Music Instruments", "Liquor" and "Electronic Devices". The DAG will then look like figure-10[10].

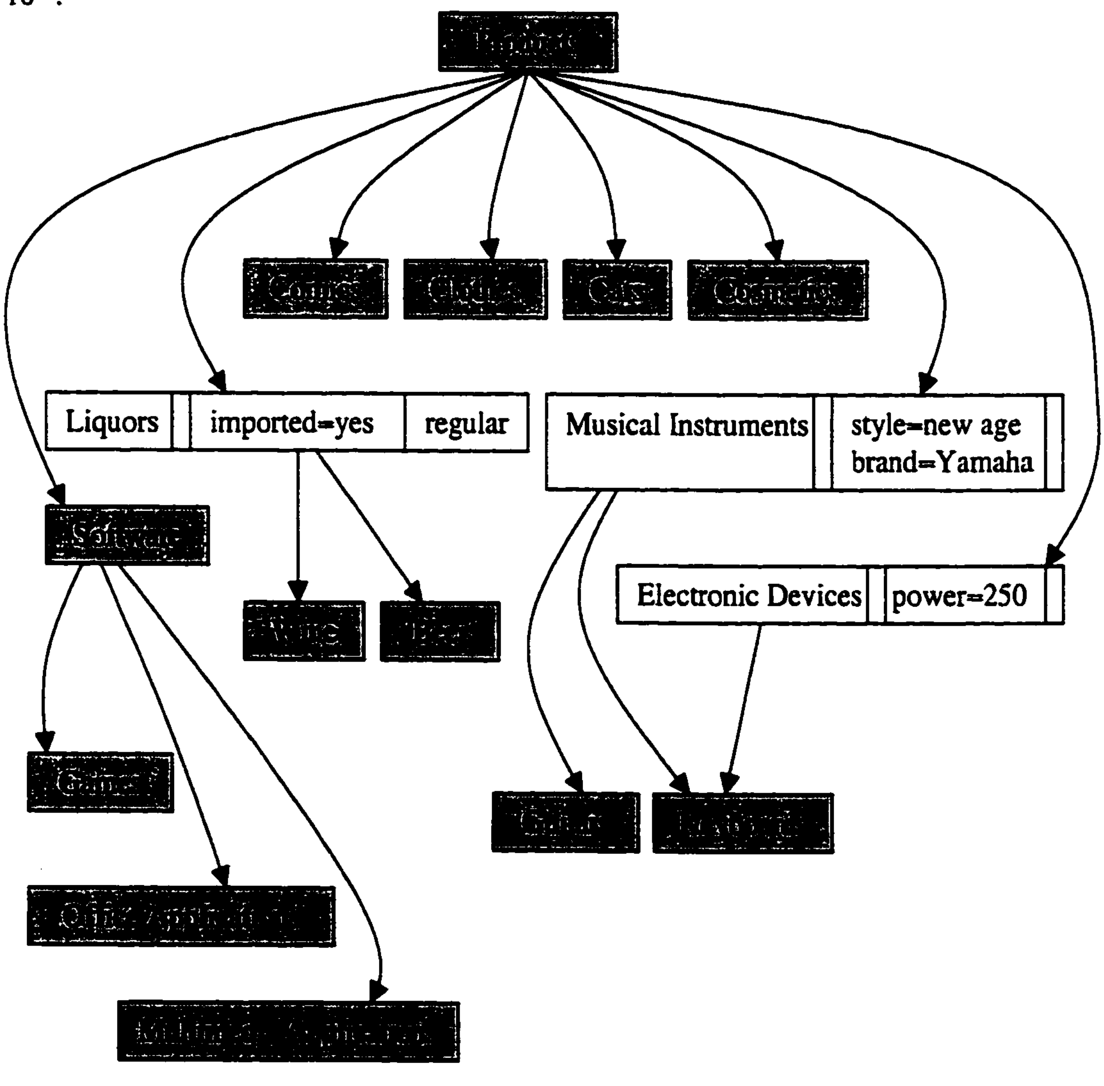


Figure 11. First modification to user model

[10] For the sake of simplifying the diagram, and because of the lack of space, the values for each node (PIE) are not shown. The values get inherited the same way as the features and user info. The way the values get computed and decided is discussed later in the thesis.

From this information, we were able to form three PIEs as our user model (which are "Liquors", "Musical Instruments", and "Electronic Devices"). We also were able to associate some feature and additional user information to the PIEs. The "regular" user information in the "Liquors" PIE tells us with which frequency we can suggest drinks to the user. We suppose that a regular drinker will have a drink every week. Whereas a casual drinker will have a drink only on special occasions, like Christmas and the New Year. The "power = 220" feature is another information that was deduced based on the fact that the user lives in Europe where the power of electric current at homes is 220 volts instead of the 120 volts in north America.

Later if we learn about the user that he's interested in multimedia applications, plays the keyboard and likes beer, then we turn on the appropriate nodes. The nodes in the DAG inherit information from their predecessor (generalization) nodes. So, we can deduce that the user is a regular drinker of imported beer, and that he's interested in playing new age music preferably on Yamaha keyboards as long as the keyboard supports the 220V power. The DAG will then look like figure-11.

Apart from that, we are also able to deduce with some uncertainty that he will also prefer Yamaha multimedia software since the brand Yamaha can also be associated with multimedia software. This is because, some features, like brand names and colors, are very strong preferences for people and they don't need to be inherited from the generalization nodes. Even if the user later wants to buy a motorcycle, he might have a strong inclination for Yamaha motorcycles. The difference between inherited features and non-inherited features is the confidence we can have in this information. For inherited features, we must be sure that the feature will have the same degree of certitude as the predecessor node. This cannot be said about non-inherited features. For example, a person who dresses with a rocker style will certainly look for a rocker style shirt when shopping for shirts. He will only probably, but not surely, listen to rocker music. This is denoted by the node "shirt" inheriting the style rocker directly from its generalization node, which is "clothes". The node "music" is not in the same branch as the other nodes. This is illustrated in figure-12.

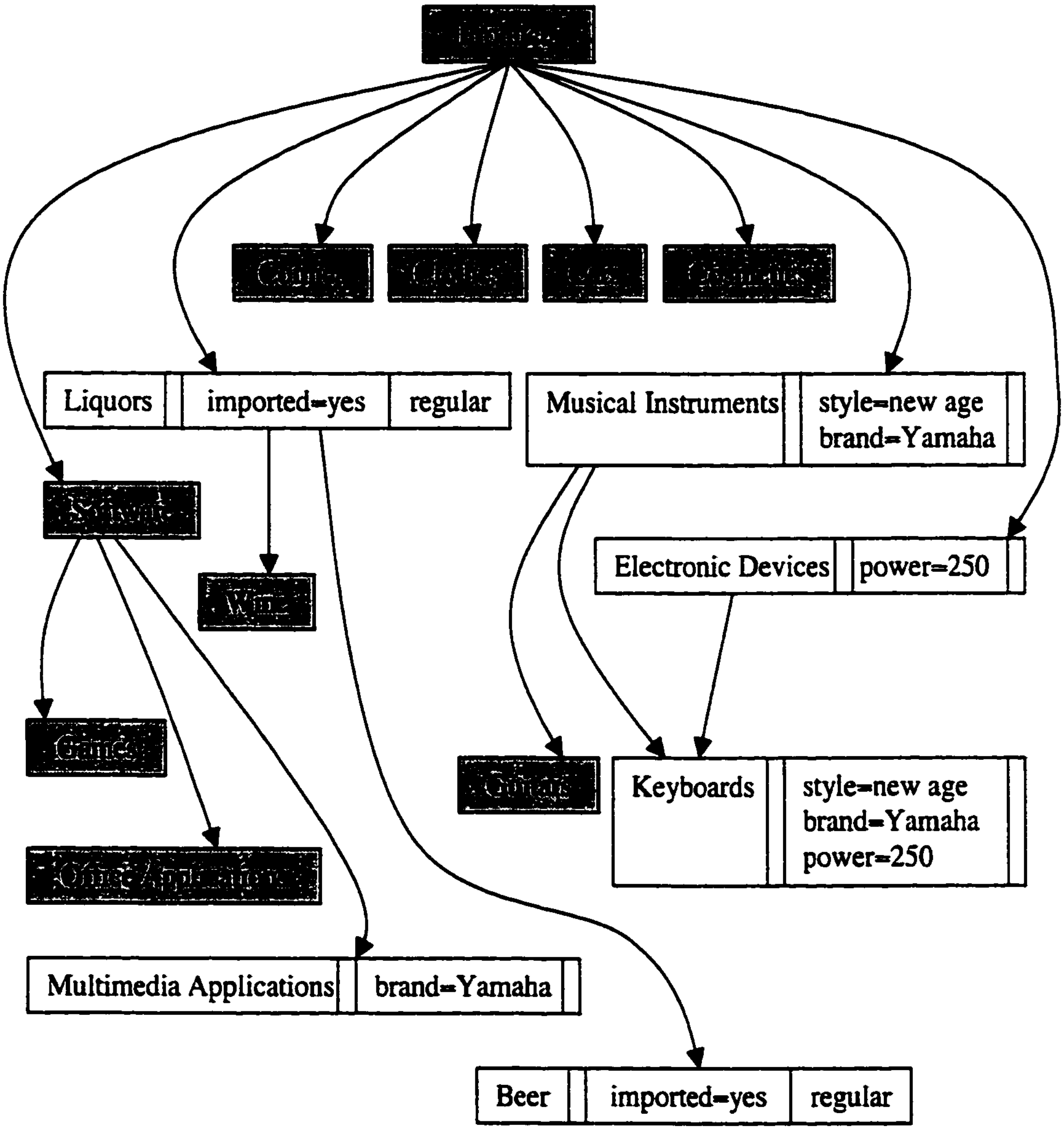


Figure 12. Second modification to user model

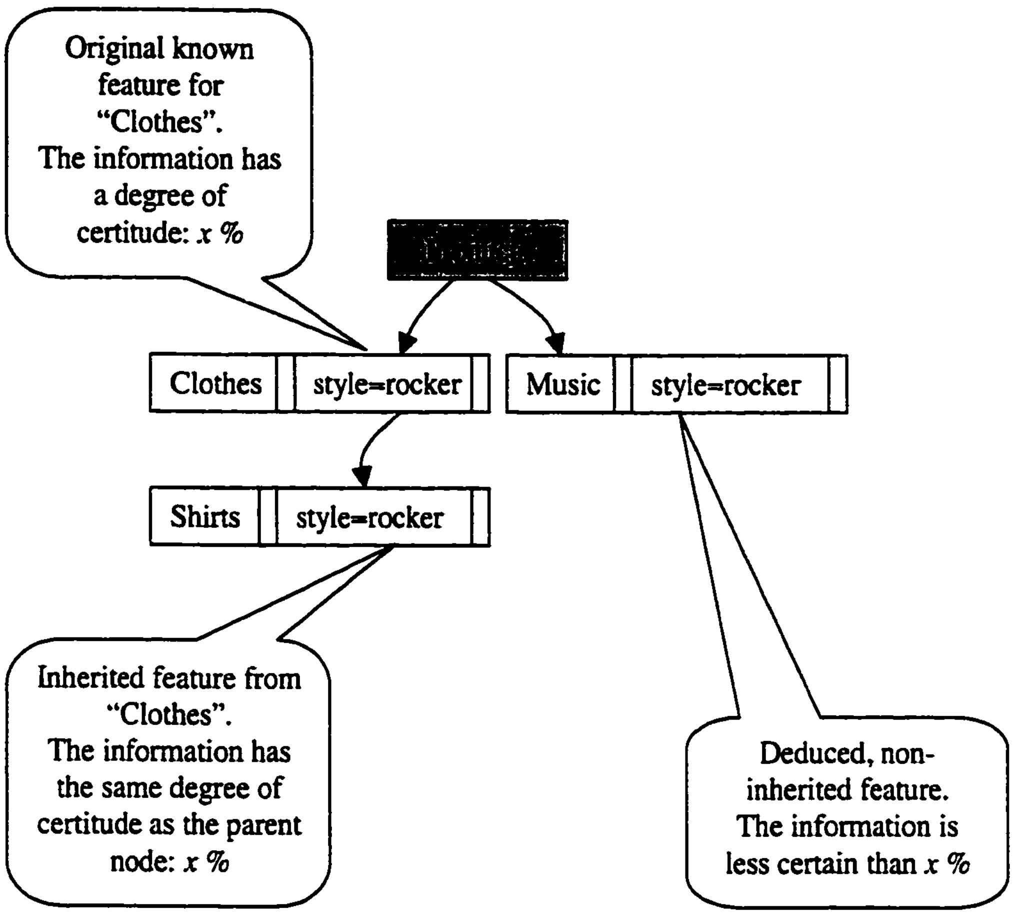


Figure 13. Inherited and non-inherited features

4.3. How the User model can be helpful in Three Modes of User-System Interaction

The suggested user model can be integrated in many kinds of interfaces and software. We believe that it can be integrated efficiently in a natural language interface, a personalized sales agent, a user shopping-agent, and an interface agent, as well as in a classical GUI. We classify the user's initiative to interact with a system into three groups, two of them are active and the third one is passive. They are querying, browsing and being informed (proactive notification). These interaction modes can be used in any of the shopping behaviors[11] explained in chapter 3. We will consider each one separately and examine how the suggested user model can be useful in each case.

4.3.1. Querying

Querying is used in comparative shopping as well as in planned shopping. We can personalize this interaction mode in two ways: (a) either by filling some fields of a query-form in anticipation, or (b) by filtering the results of the query. In the case of (a), the user can see the anticipated filling before submitting the query. We filter the results of a query when the response-set is too large to present, in a transparent way for the user. In both cases, refining the information is based on the features in the user model. The choice of when to fill in blanks and when to filter the results of a query is dependent on the significance of the fields used in the query. The significance can be obtained in two ways. We can either assign a significance factor to each field, or we can assign a significance value to each feature in the individual user model. The latter method is more complicated because it needs a way of deducing what factors are important for each different user. In the current version of our prototype, we assigned a constant significance factor to each field. For example, we considered that the brand of a car is more significant than its color. Thus, if the brand field is left empty in the query, we fill it before submitting the query. But if the color field is left empty, we ignore it until we get our query results. If we have more than say 100 match, then we show the first ten with the color that is specified in the user model.

When displaying the results of the query, the "additional information" in the user model, is used to personalize the display of the information. For example, we can show the details of the car engine to an expert in cars, and we show only the car external options for a person who is not an expert.

[11] Not necessarily in a one-to-one match.

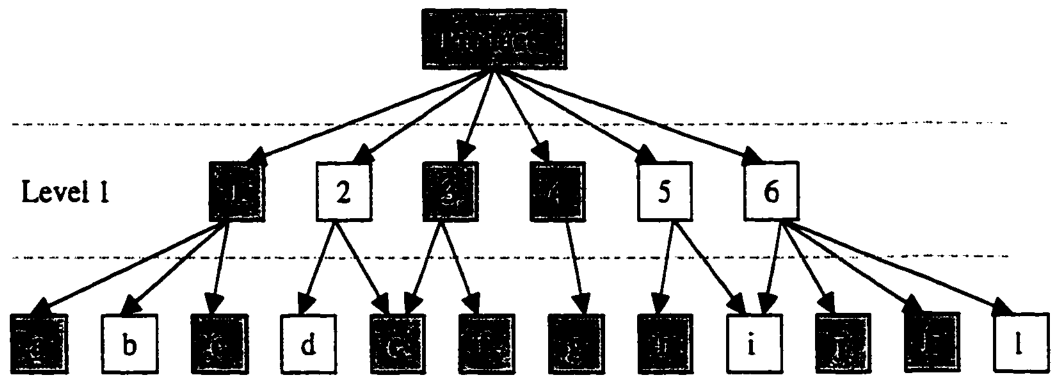
4.3.2. Browsing

Browsing activity may be used in planned shopping, and certainly in browsing-based shopping (window-shopping). The personalization here is at a different level. While browsing in a department store or a virtual mall, the user can go from one department to another or from one sub-department to another. We refer to this as navigating on the same level (or breadth-first traversing). He can also navigate up and down in the hierarchy starting from a department, to a category, to a sub-category, till he gets to the product level (this is depth-first traversing). The personalization possible here, is to show the departments, categories, or sub-categories that interest the user before showing him the others that are of no interest to him. The screen is limited in space, and not everything can be shown at the same time. To facilitate browsing, departments and sub-categories that have a "high priority" in the user model can be shown before others, thereby reducing mouse clicks and scrolling of windows[12]. The DAG of PIEs is useful here to make this kind of information filtering.

When navigating on the same level, or up and down the hierarchy, categories or departments that have high values in the user model are displayed first. So the mapping is direct from the DAG in the user model to the browsing interface. The way the user model is reflected in the browsing interface is illustrated in figure-13. The nodes in gray, designate nodes with low values, and the nodes in white are nodes with higher values. Figure-3 (a) shows the user model, which is composed of all the white nodes (PIEs) on the DAG. Suppose level 1 is the department level, and each of the nodes 1, 2, 3, 4, 5 and 6 designate different departments in a department store. The user is interested in departments 2, 5 and 6. Figure-13 (b) and (c) show a simple browsing interface. This interface has only enough space to display four images. In Figure-13 (b), the user model is not exploited and the first 4 departments in alphabetical order are shown first. To view the remaining departments, the user has to click on the right bottom arrow (or scroll, depending on the interface). In figure-13 (c), the user model is used to decide which

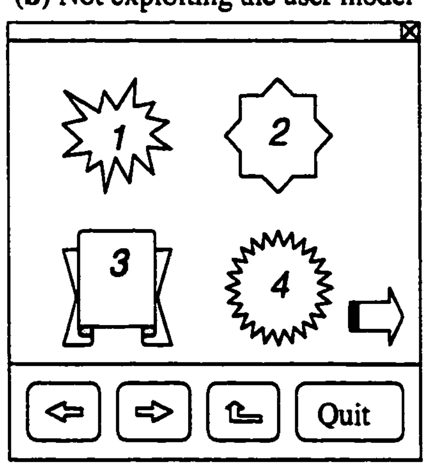
[12] This technique can be used in a virtual reality environment or in a classic GUI.

departments to display first. These are the departments that correspond to the PIEs that have high values in the DAG on the department level, starting with the one with the highest value. The user can still have the possibility to view the other departments. This method can be used on any level while browsing.



(a) The user model

(b) Not exploiting the user model



(c) Exploiting the user model

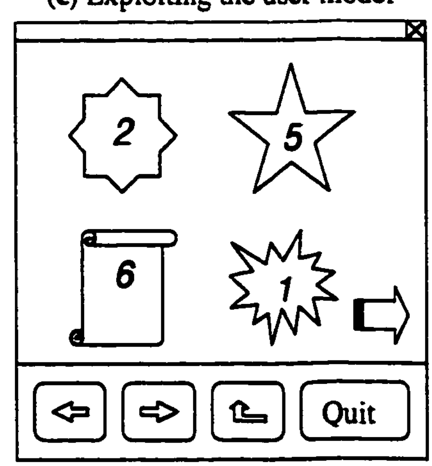


Figure 14. Showing how the user model is exploited in a browsing interaction

4.3.3. Proactive Notification

We use proactive notification whenever we find that the user might miss a good opportunity. One may act on the vendor's behalf or the user's behalf. We focus on the latter case. This kind of interaction could be in the form of a pop-up window that displays a message like: "Did you know that product X is available now at the price of Y?" or "Did you know that product X has a feature Y?" Proactive notification is used during any type of shopping behavior. It can be initiated by a change in the database, a change in the user model, or an external event, like an important date (Christmas, Valentine's Day, or a birthday) [Lu 1999]. In our prototype we included this kind of proactive notification. The message displayed is based on the knowledge stored in the user model, on the content of the database and on the user additional or personal information (like the address and the date of birth).

The additional or personal information about the user, combined with an external event can trigger a proactive notification message. For example, when a collector's item becomes available, the user who is a collector is notified. On Christmas, a casual drinker is notified of available and new brands of liquors. A student is notified when there is a price reduction on fashion clothes, etc.

4.4. Building User Models

As we noted earlier, user models can be explicit or implicit. Explicit user models are the ones where the user personalizes the system by setting environment variables to his needs. Implicit user models are the ones built by the systems based on the interaction with the user.

We prefer our model to be an implicit model for many reasons. First the user is often unable to judge what system preferences are better for him. Second, the user usually finds it annoying to be asked a lot of questions before being able to use the system. That's why we start with an initial user model that gets refined along the way. The more the user interacts with the system, the more information the system gathers about him to refine his model. This way the user will have the minimum possible work to do in order to build his model. For this we have to use some machine learning techniques. Techniques to build user models fall into two broad groups: methods to infer single facts at a time, and methods to infer many facts at once [Rich2 1983]. In our user model both techniques are employed. The following two sub-sections discuss these two kinds of techniques.

4.4.1. Inferring Individual Facts

One technique to build the user model is known as "look and learn". This means observe how the user is interacting with the machine and deduce relevant facts about him from this interaction. If a person is using advanced commands in a command oriented system, then we can deduce that he is an expert. Someone who performs commands, that are often rejected by the system, is probably a novice. There are often many ways to do the same task in a system. For example, to save a document in most word processor systems under windows, one can either, click on the diskette icon, click on the "save" item in the pull down menu, or simply press "Ctrl+C". Different commands are performed by people with different expertise. So from these man-machine interactions we can deduce facts about the user and form his model. This model helps then personalize response messages from the system.

4.4.2. Inferring Many Facts at a Time

Human traits are not distributed randomly throughout the population; rather they tend to occur in clusters. These clusters can arise for a variety of reasons, such as the existence of a single factor that causes several traits to be present at once, or the existence of a causal chain among the traits themselves. For example, a person who is very wealthy is likely to have traveled more than another person who is very poor. These co-occurring traits are called "stereotypes" [Rich2 1983].

Stereotypes provide a powerful technique that people use to build models of other people by evoking the cluster of characteristics associated to a certain stereotype in a specific situation. For example, one might know that if someone is a judge, he or she is probably over forty, well educated, reasonably pro-established, fairly affluent, honest and well respected in community [Rich1 1979].

In the same way, stereotypes can be used by computers to build user models. Furthermore, with computers it is possible to avoid some of the pitfalls that plague the human use of stereotypes. Computers don't develop any emotional attachment to their stereotypes and so are immune to whatever extent that contributes to the inadequacies of stereotypes [Rich1 1979].

4.4.3. Acquiring and Refining the User Model

The e-commerce user model described in section 4.1 can be incorporated in a natural language interface, a graphical user interface, as well as a virtual reality interface. Depending on the type of interface employed, the methods for acquiring the user model can vary. In this thesis, we describe the ways to build and refine the user model in a GUI type interface.

In the previous chapter we described three modes of user-system interaction. The user's actions can be monitored in querying and browsing to infer facts for his model. During the proactive notification, it is not possible to infer facts about the user, since in this mode the user is passive and only receives information. Before monitoring the user's actions during querying and browsing, we need an initial model to start with. The initial model is built based on a registration form. Further refinements are performed on this initial model.

To build the model and to do this refinement, we use triggers. Triggers are actions initiated by the user during registration, querying, or browsing that can imply certain information. There can be several triggers. For our design, we defined a small set of can be triggers that is expanded after implementation and experimentation.

We defined two kinds of triggers: triggers that allow us to infer a single fact at a time, and triggers that allow us to infer several facts at the same time. The idea of the latter kind of triggers is very similar to Elaine Rich's stereotypes: "whenever we know that the person belongs to a certain group of people, we can deduce more than one characteristic of this person that is commonly shared by the group". In the rest of the section, we will discuss the triggers in the registration, querying and browsing activities.

In the registration form we have several triggers. For example, checking the female box allows us to suggest that the user might be interested in perfume products, kitchen items and furniture. Checking the male box suggests possible interest in cars, electronics, and sports items. Of course, these deductions may not always be true. That's why these preferences have to be updated, refined, overridden or corrected during browsing and querying by the triggers that are defined in the browsing and querying interaction modes.

Another example of deduction from the registration information form is the apartment number field. If this field is filled with a number, then we can deduce that the user lives in an apartment and therefore he is not likely to be interested in garden tools and garden furniture. Leaving the apartment field empty does not necessarily mean that the user owns a house, but that is more likely.

In the registration form, there is also an occupation field where the users can select an occupation from a drop down list. From each entry in this list, we can deduce facts about the user. For example, being a homemaker suggest that the person is interested in home hardware and toilet products. Being self-employed suggests that the person is interested in computer related hardware and office software. Marking engineer as occupation suggests that the user will understand cars inside mechanism and advanced technology.

So we can personalize the display of the same information to adapt it to the understanding level of the user. Being a student suggests certain features associated to products (low prices, sports casual wear, sports car).

Some other facts can be deduced in a straightforward manner from browsing and querying behaviors of users. For example, if after logging in, the user starts browsing by clicking on the clothing department, we can deduce his interest is in fashion. This deduction or the belief becomes more evident if his action is repeated several times during subsequent sessions. That's why, the first time it occurs, we assign to the PIE "clothes" a value of 1. This value gets incremented every time the user performs another action that implies interest in fashion. If after a certain number of sessions the user has not done any action to reinforce this belief, then the value starts to decrement. In general, every time a trigger is invoked to imply interest in a category of items, the corresponding PIE gets its value incremented by 1 till it reaches 5 which is the maximum value that can be assigned to a PIE. After a certain number of sessions (a measure of time elapsed) if no interest was shown in this category, the value starts decreasing till it reaches 0, which denotes indifference.

This information and the information inferred during querying can override the information inferred from registration, if it is different.

From querying we can also deduce the tastes and needs. For example, searching for an item and not finding it makes us deduce that as a "need". A repetitive querying pattern can infer a user's "tastes". In fact the triggers associated with the querying activity can help us infer new knowledge that is more specific and accurate than the knowledge inferred from other triggers. This is due to the fact that while querying, the user specifies products as well as features.

5. The User Model within the Architecture of the CITR Project

5.1. The CITR Project

In order to talk about this issue, let us introduce first the global architecture of the CITR project. Figure-14 summarises the architecture of the demo we developed for CASCON 2000 conference held at Toronto and sponsored by IBM Canada.

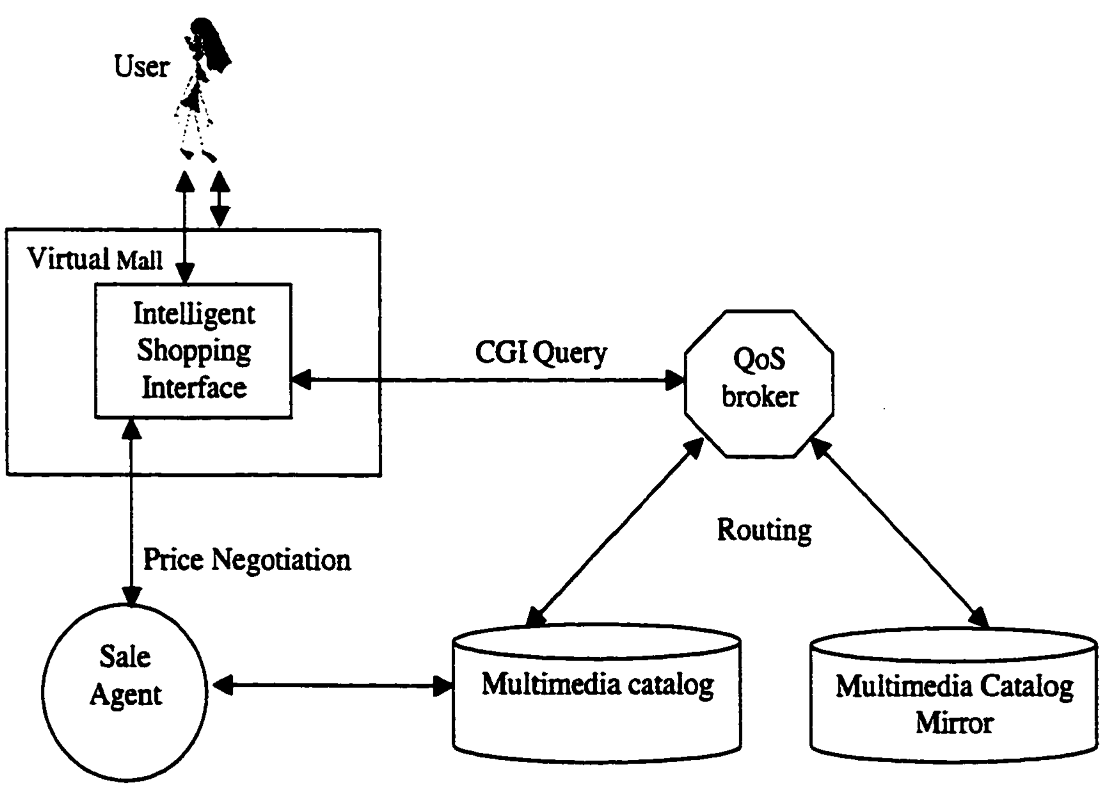


Figure 15. Global and top level Architecture of the CITR project

Conceptually many intelligent shopping interfaces are aggregated inside the Virtual Mall. Each vendor has his own interface. The user is able to query the database through this interface. Instead of querying the database directly, the query is sent to the QoS (Quality of service) broker which routs the message to one of the databases. The results are returned, displayed in the shopping interface and viewed by the user. The user chooses what he wants to buy, and before confirming the transaction, is given the choice to negotiate the price with the sales agent. For the Sales Agent to perform the negotiation, it needs the list of the items bought along with their sale prices and cost prices that he can retrieve from the multimedia catalog.

Since the user model is integrated with the shopping interface, let's take a closer look at this entity. Figure-15 shows further details.

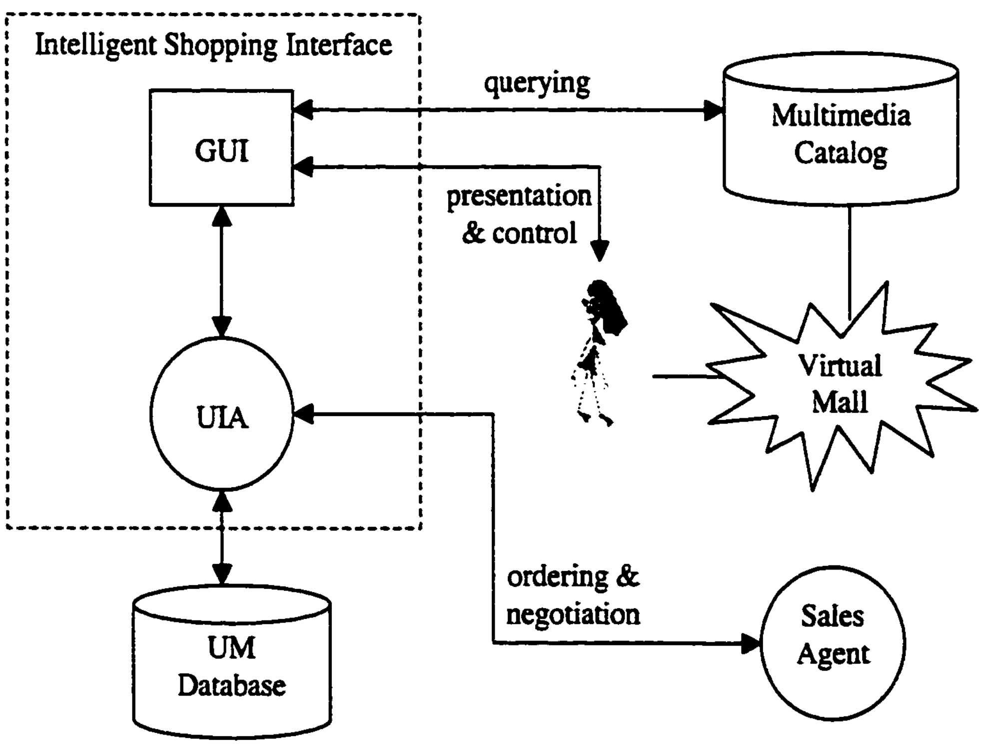


Figure 16. High level architecture of the Intelligent Shopping Interface

In the prototype built, a person is allowed to use the system as an anonymous user. He can decide to use the system without registering or logging in. Obviously there will be no personalization in this case. Figure-16 shows the registration and login process. At first the user is given the choice to use the system anonymously, login in the case he's known to the system, or register in the case he's a new user. After deciding on this, the user can see a list of links to many virtual malls, and the links to existing shops inside each virtual mall[13]. The user can decide then to click on the link of a mall or a specific shop. If he decides to go to a mall, he can then explore the 3-D environment of this mall, chat with other users inside the mall and go from one shop to another. If he decides to click on a specific shop link, he will have the proper browsing and searching interface for each shop. The personalization was done on the level of each shop. The user has also the possibility to alternate from the 3-D mall virtual environment to the specific environment of a certain vendor.

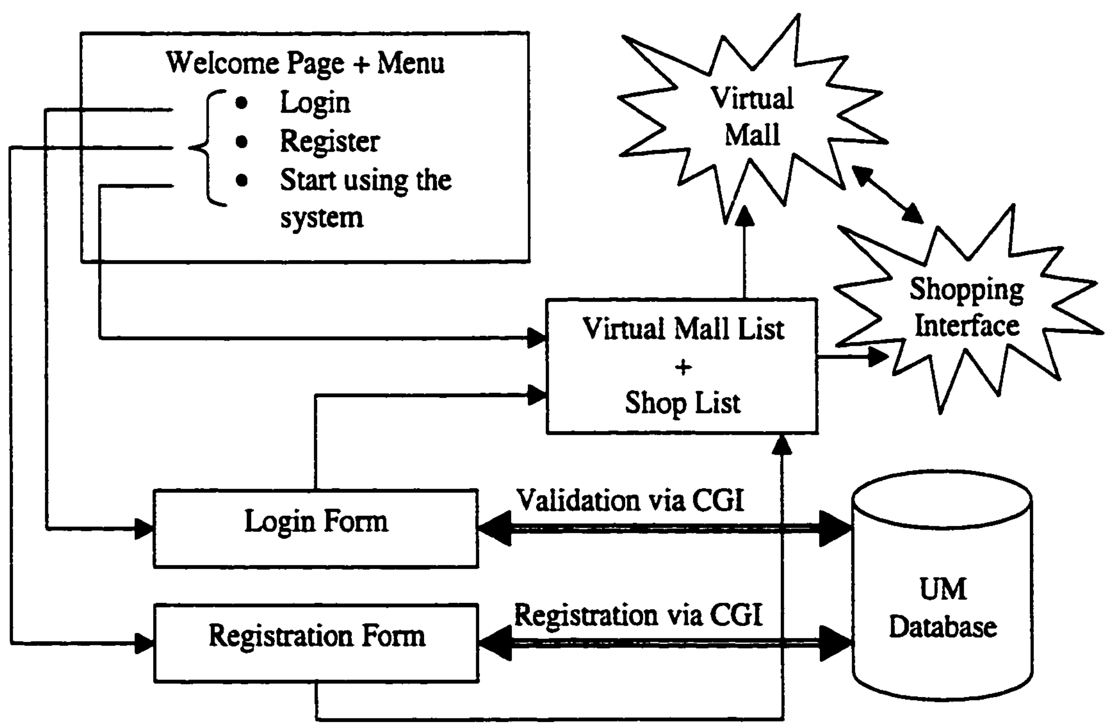


Figure 17. Login and Registration Process

[13] In the prototype built the list of malls had only one active link, and the list of shops had only three shops with active links.

5.2. Where to Store the UM, Security and Privacy Issues

In the context of B-to-C e-commerce, the user model can be stored either at the client side or at the server side. More specifically, in the context of the architecture of the CITR project presented in the previous section, the server side can designate the virtual mall, or each vendor's server[14]. In the prototype we built, the user model was stored in a separate auxiliary database on an *apache* server. In a real working application, a lot of consideration should be given to security and privacy issues upon deciding where to store the user model. We analyzed the advantages and disadvantages of each of the following three possibilities:

Storing the user model at the client side that is at the individual user's machine. This approach has two main advantages over the other approaches. The advantages are security and privacy. In this case, the user has maximum control over his information since it is stored locally on his home PC. He can view it, delete it or change it as he wishes. The inconvenience of this approach is the limitation in the accessibility of the user model away from home. If a person wants to use the system from his brother's house, he will not get any personalization. Another negative aspect of this approach has to do with the implementation. If the shopping system is built as an applet then the applet has to write cookies on the user's machine. Cookies are not always popular and some people feel bad about them, knowing that the applet is creating and writing files on the machine without their permission.

Storing the user model at the server side in the mall and this means that all vendors can have access to and will share the user model database. There are many advantages in this approach, mainly the possibility of using the system from any computer, the convenience of implementation, and the practicability of having one complete model instead of many partial models. This can eliminate redundancies. This can also consolidate the model as contradictory inferred information coming from different shopping sessions with

[14] each vendor has a separate product database, meaning vendors may have separate servers.

different vendors becomes useless and get to be discarded, and repeated information coming from different vendors can get a higher degree of certitude. In addition to that, some information coming from one domain can help deduce other information for another domain. For example, knowing what kind of music a teenager listens to, can suggest a style of clothing for that teenager.

On the other hand, there are some disadvantages for this approach. First, the vendors can misuse the private information of the user model. For example, knowing that the customer shops sometimes with a rival commerce, the vendor can go harder on price negotiation, or might even refuse to negotiate. Having access also to the sessions' history of the client might affect the price negotiation. For example, if a vendor can be aware that the customer bought last week a 15000$ car, the vendor might refuse to negotiate the price of a 100$ jeans. Finally, the last disadvantage has to do with trust: it's difficult for the customer to trust the whole mall, which means trusting all the vendors.

Storing the user model at the server side with each vendor and this means that each vendor will have a user model database for his customers. As in the previous option, here the advantage is the convenience from the implementation perspective, since the user model will reside on a database server and there will be no need to use cookies. The user can also login to his account and get the personalized service from anywhere. The disadvantage is the possibility of redundant information. In both options where we store the user model at the server side, there is the disadvantage of being susceptible to network conditions and network failures.

The trust issue in this third option is in between the other two approaches. Trust is at its lowest degree when storing the information in the mall, since the user may trust one particular vendor in the mall but not the whole mall. Trust is average when storing the information with one vendor, since the user may trust the vendor but still can't anticipate what the vendor might do with his information. For example, a vendor may, although he should not, sell his customer information to Equifax [Equifax 2001]. Finally, trust is maximum when keeping the information on the user's machine, since he'll feel he's got maximum control over his information in this case. It is to be remarked here, that even in this case, some of the information sent online during each session could be exposed, so the need for encryption and security protocols is always there.

None of these options is perfect; each one has its advantages and disadvantages. An alternative way will be also dividing the user model and distributing it among the three possibilities. Part of the model can be local to the client side, for example very personal information like a credit card number. Another part of the model can reside on the server side divided again between the mall and the individual vendors, in a way each vendor can have access only to the information that is relevant to his domain and that can help to give a better service.

A fourth and most ideal option has to wait for the future so that adequate hardware becomes available and cost effective. Ideally, the user model should be stored on a personal magnetic card, and the hardware should allow for a slot where this card can be inserted. The user model can then be loaded, used, updated, and written back to the card.

6. Implemented Prototype of the User Interface Agent

This chapter presents the demo that was implemented[15] in order to incorporate the user model discussed in this thesis. The prototype developed is a shopping interface for a "department store" that allows the three interaction modes discussed earlier in chapter 4, (i.e. querying, browsing and proactive-notification).

6.1. The Architectural Design of the Prototype

The architecture of the prototype includes two major components:

- The database design
- The interface and internal classes

The database design includes two distinct databases:

i. *User database*: responsible for storing the user model

ii. *Product database*: responsible for storing the different products in the department store

There are eight main objects in the application:

1. *Login Form*: to allow registered users to log in for the service.

[15] My role in this implementation was as analyst, top level designer and consultant. The prototype was coded and tested by Mashrur Mia. Acknowledgment is given at the beginning of the thesis.

2. *Registration Form*: to allow a new user register and take advantage of the personalized service.
3. *Browser*: an interface to allow browsing among different departments of a store, product categories and products.
4. *Query Form*: to query the database for specific products. Several forms were supplied for different categories of products.
5. *Proactive Notification*: a window that pops up to inform the user of good opportunities for buying.
6. *Product Order Form*: to allow the user to submit an order.
7. *Database*: an object that handles all interactions between other objects and the database.
8. *Update Entry*: an object that takes care of updating the user model based on messages from other GUI objects.

The demo is developed in Java Swing using the "Symantec Java! JustInTime" compiler version 3.10.093(i) for JDK 1.1.x. The user model and the products database were implemented in Access 2000[16].

6.2. The GUI of the Prototype

The application allows the users to query or browse the content of the database with or without registering. In order to take advantage of the user model, users must register. This is a way to illustrate the difference between personalized and non-personalized service. Figure-17 shows the login form. Registered users can login by typing the "id" and password. New users can register by clicking on the "Register" button at the top of the form. The "Query" and "Browse" buttons allow anonymous user to query the database or browse its content.

[16] For more details concerning the implementation, refer to the document "User Model" written by Mashrur Mia as part of Comp490 project course.

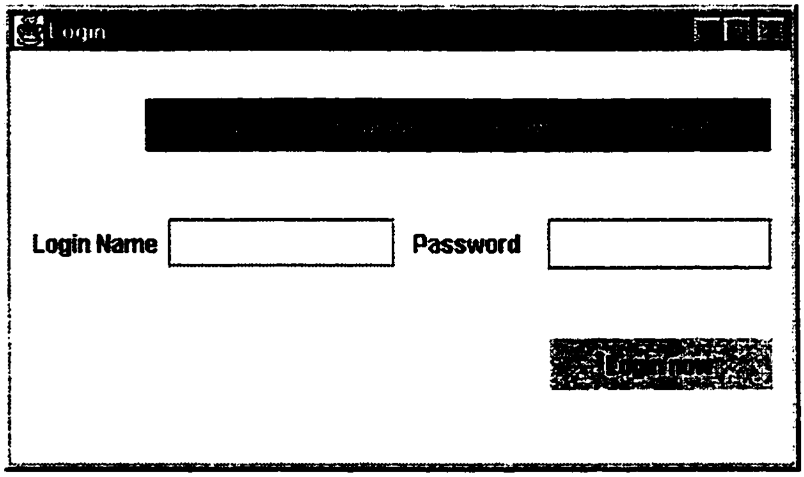


Figure 18. The login form

Figure-18 shows the registration form. Upon registration, the user is assigned an "id" and an initial model. The initial user model is created based on the information supplied using this form.

Figure-19 shows the browser window. The user can always browse up or down the hierarchy (represented by the DAG of PIEs) or stay on the same level. This means the user can go from a department to a shelf (or category) to a specific product, or from a product back up to the department level. He is also able to go from one department to another, or one product to another, or one category to another (staying on the same level). While browsing, the user model is constantly referred to, for determining the items of most interest to the user. This is decided based on the ratings given to the nodes of the DAG discussed earlier in this thesis (section 4.3.2). At the bottom level (leaves), the features are used to decide which products to show first. The interaction with the browser also helps update the user model in a way that repetitive clicks on the same graphical item causes the rating of the corresponding node to increase.

Figure-20 shows an example of a query form intended for car buyers. Several different query forms were implemented for different types of items based on their attributes. Here again the user model is exploited by either filtering the results of the query when the set is too large or by filling values for missing fields. Both services are based on the features in the user model. The query also helps to update the user model bye acquiring features that were explicitly specified by the user.

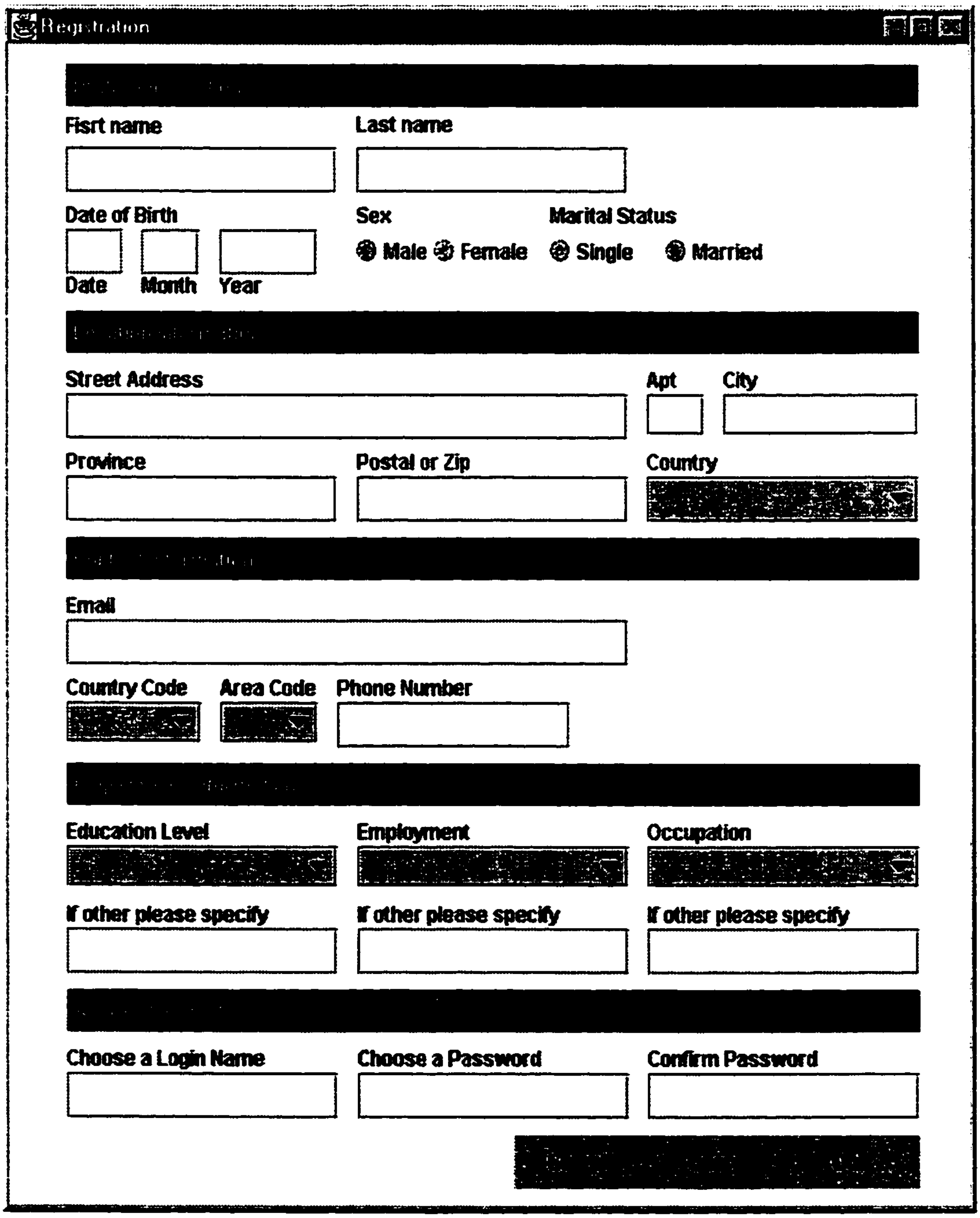


Figure 19. The registration form

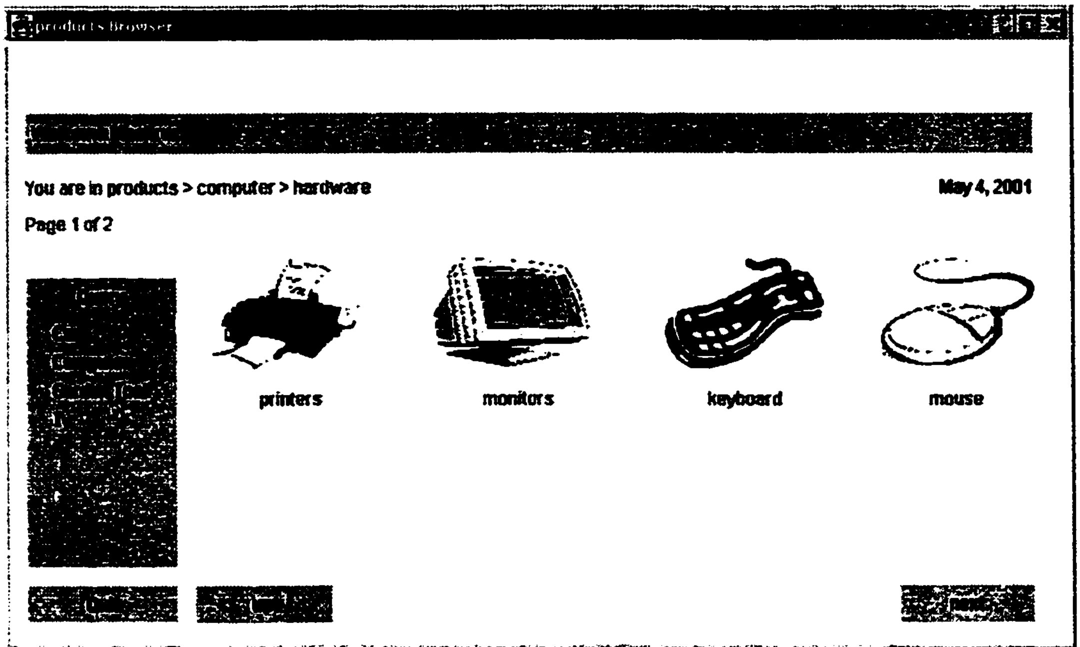


Figure 20. The browser

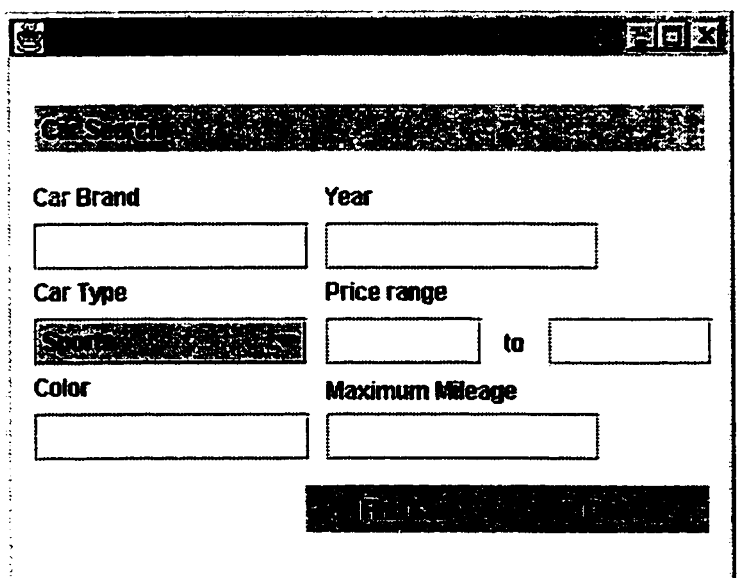


Figure 21. The query form for cars

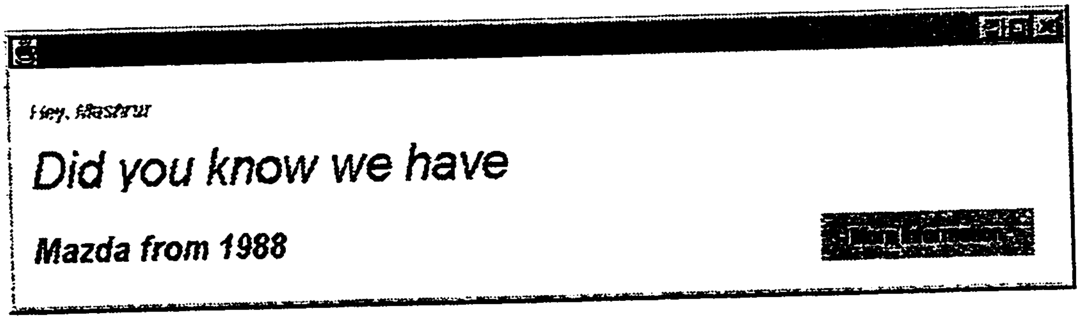


Figure 22. Proactive notification

Finally figure-21 shows the proactive notification window that notifies the user of important opportunities. Usually proactive notification is initiated when there is a match between the features of a bottom level node with a high rating in the user model (product node) and an existing item in the product database available for sale. Search for matches is done once at the beginning of each session, and after that the user model is updated so that the system knows that this user has been notified so as not to notify him again.

6.3. Objectives Met by Building the Prototype

After presenting an idea, the purpose of building a prototype incorporating this idea can be one or more of the following:

1) Studying the scalability
2) Studying the actual response time and not the estimated time
3) Testing and evaluating the usability by the targeted class of users
4) Proving the concept and demonstrating the feasibility of the idea
5) Illustrating the idea when it is difficult to explain it by other means

In our case, numbers 1, 2 and 3 constitute big projects by themselves, and demand a deeper study and bigger commitments. They can be pursued as a future study. The actual purpose of the prototype we developed is proving the concept, demonstrating the feasibility, and illustrating the idea. We believe we have met these goals. Also we would like to add that illustrating the idea, proving the concept and demonstrating the feasibility were achieved by building a prototype with a GUI type interface.

In order for us to get a feel for how the users might find the user interface based on our user model, we conducted a small non-exhaustive survey. We selected five users, and they were selected more on availability basis rather than random basis. They are graduate and undergraduate students in computer science, and in this sense they do not represent the real population of users. The five students are: Alina Andreevskaia, Mohammad Lokman Hossain, Daniel Sinnig, Helder Antunes and Asmaa Al-Sumait. To conduct the study, we started by giving them a demo. After that we created and account for each one of them. We asked them to use and explore the interface. Finally we interviewed the users and we organized their feedback in the tabular form presented below.

Pros	**Cons**
• Browsing organized in a tree structure • Multiple interaction modes to support different behaviors • The possibility to turn off the user model and use the system anonymously • Easy to use and learn interface • Seamless connection to database • Interesting mechanism for user profile updates • Update of user model is dynamic and transparent • Supports three shopping behaviors • Rules are extendible	• No field of interest in the registration form • Multiple widows • Not very aesthetics interface • Stereotypes are not scientifically proven • Database doesn't have a lot of items • It ignores cultural variations and difference of climates

A lot of the feedback had to do with the interface itself instead of the personalization. For example, the fact that the interface has multiple windows instead of a single window.

Since all the users were also computer science students, they were curious enough to look at the code and the mechanisms used. That's why we got feedback like "seamless connection to the database, "update of user profile is dynamic and transparent" and "rules are extendible".

Generally speaking, the selected users appreciated the multiple shopping behaviors and multiple interaction modes supported by the application. They appreciated equally the fact that we can turn off the user model and use the system anonymously. The interesting thing learnt from their feedback is realizing that we could have worked more on validating the stereotypes and on defining more rules based on cultural differences.

7. Conclusions and Future Work

7.1. Conclusions

Based on the needs of users, the three types of shopping behaviors are explored in detail. They are browsing, planned shopping and comparative shopping. There are commonalties and differences among them. Among the differences are the lengths of the time-window available for the user to complete the shopping activity, and the detail to which the user needs are known to himself/herself and to the system. The user model that is proposed in this thesis consists of a directed acyclic graph of PIEs (Preference Indication by Example) and a set of classified triggers or user initiated actions that refine the user model. Our user model is inspired by Elaine Rich's earlier work on "stereotypes" and our perceived need to broaden the coverage of the domain of products while dealing with a virtual or electronic shopping mall. The contributions of this thesis are:

- The new user model we proposed is generic in the sense that the domain of application is relatively broad (department stores, shopping malls)
- The user model is useful in the three shopping behaviors identified.
- Based on the user model, filtering of information can be done at different levels (departments, shelves, categories, products)
- Ways to exploit the user models are suggested in the three modes of interactions (querying, browsing, proactive notification)
- A prototype has been developed as a proof of the concept and it was tested by a small set of five users. Preliminary observations indicated the qualities and weaknesses of the user model, and suggested further enhancements.

The types of user interactions presented in this thesis and the user model were incorporated in an "Intelligent User Interface for a Department Store". This implementation will be integrated into the CITR project on Enabling Technologies for Electronic Commerce.

7.2. Future Work

- The user model proposed is based on a PIE hierarchy. This has the properties that would make the model usable not only in electronic commerce applications but also in information retrieval systems where the information can be semantically organized as a DAG hierarchy. This needs to be further explored.

- The current trend in user interfaces is personalization [Adomavicius 1999] [Riecken 2000]. The new user model proposed has the property of specializing to any degree that is needed (by going down in the PIE hierarchy). Such highly detailed user model can be helpful in personalization. This also needs to be explored.

- The usability of the prototype discussed in chapter 6 can be expanded into a full-scale study.

- A better and more robust prototype should be built for further exploration. This should take into account the feedback we got from users, mistakes that were made, and possible enhancements.

8. References

1. [Adomavicius 1999] Gediminas Adomavicius, Alexander Tuzhilin, "User Profiling in Personalization Applications through Rule Discovery and Validation", Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 377-381, 1999
2. [Allen 1997] Robert B. Allen, "Mental Models and User Models", Handbook of Human-Computer Interaction, chapter 3, pp. 49-63, 1997
3. [Alspector 1998] Joshua Alspector, Aleksander Kolcz, Nachimuthu Karunanithi, "Comparing Feature-based and Clique-based Models for Movie Selection", Proceedings of the third ACM conference on Digital libraries, pp. 11-18, 1998
4. [Amazon 2000] Amazon.com: http://www.amazon.com/, 2000
5. [Baecker 1997] Ronald M. Baecker, Jonathan Grudin, William A. S. Buxton, Saul Greenberg, "Introduction to Human-Computer Interaction", Readings in Human-Computer Interaction: Toward the Year 2000, Part 1, pp. 1-3, 1997
6. [Barker 2000] Philip Barker, "Mental Models and Network Pedagogy", University of Teesside, UK, URL: http://www.enable.evitech.fi/enable99/papers/barker/barker.html
7. [Bolin 1998] Sherrie Bolin, "E-commerce: a market analysis and prognostication", StandardView, Vol. 6, No 3, pp. 97-105, 1998
8. [Brajnik 1987] Giorgio Brajnik, Giovanni Guida, Carlo Tasso, "User Modelling in Intelligent Information Retrieval", Information Processing and Management, vol. 23, No 4, pp. 305-320, 1987
9. [BSF 2001] Le Groupe San Fransisco: http://www.bsf.ca/, 2001
10. [Chin 1986] David Chin, "User Modeling in UC, the UNIX Consultant", CHI'86 Proceedings, pp. 24-28, 1986
11. [CITR1 2000] Canadian Institute for Telecommunications Research (CITR): http://www.citr.ece.mcgill.ca/english/citr.html, 2000

12. [CITR2 2001] Enabling Technologies for Electronic Commerce: http://www.cs.ualberta.ca/~database/EC/ec_index.html
13. [Desharnais 2000] Patrick Desharnais, "Software Sales Agent for Price Negotiation", M. Comp. Sc. Thesis, Department of Computer Science, Concordia University, 2000
14. [Equifax 2001] Equifax Canada - Consumer and Commercial Information Solutions: http://www.equifax.ca/, 2001
15. [Erdogan 2001] Emre Erdogan, "Software Agents versus Programs", URL: http://www.cs.bilkent.edu.tr/~emree/ir/wesite/diff.html, 2001
16. [Georganas 2000] N. D. Georganas, "Distributed Virtual Environment User Interface and Intelligent Agents", University of Ottawa, URL: http://www.mcrlab.uottawa.ca/research/CITR_98.htm, 1998
17. [Guttman 1998] Robert H. Guttman, Alexandros G. Moukas, Pattie Maes, "Agents as Mediators in Electronis Commerce, Electronic Markets, Vol. 8, No. 1, pp. 22-27, URL: http://www.e-global.es/011_gutmann_agents.pdf, 1998
18. [Herschlag 1998] Miriam Hershlag, "Shopping the Net", Time Magazine, July/1998
19. [Howes 1991] Andrew Howes, Richard M. Young, "Predicting the Learnability of Task-Action Mappings", Human factors in computing systems conference proceedings on Reaching through technology, pp. 113 – 118, 1991
20. [IBM 2000] Understanding the user's perspective - the user's conceptual model, URL: http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/11, 2000
21. [Johnson-Laird 2000] Phil Johnson-Laird, Ruth Byrne, "Mental Models Website", URL: http://www.tcd.ie/Psychology/Ruth_Byrne/mental_models/, 2000
22. [Jonassen 2000] David H. Jonassen, "Operationalizing Mental Models: Strategies for Assessing Mental Models to Support Meaningful Learning and Design- Supportive Learning Environments", URL: http://www-cscl95.indiana.edu/cscl95/jonassen.html
23. [Lu 1999] Ji Lu, "User Interface Agents in Electronic Commerce Applications", M. Comp. Sc. Thesis, Department of Computer Science, Concordia University, 1999
24. [Maes 1999] Pattie Maes, Robert H. Guttman, Alexandros G. Moukas, "Agents that Buy and Sell: Transforming Commerce as We know it", Submitted to the Communication of the ACM, March 1999 Issue, URL: http://www.e-global.es/006_maes_ia.pdf

25. [Mann 1996] Steve Mann, "Smart Clothing: Wearable Multimedia Computing and Personal Imaging to Restore the Technological Balance between People and their Environments", Proceedings of the fourth ACM international conference on Multimedia, pp. 163-174, 1996

26. [Mountford 1990] S. Joy Mountford, "Tools and Techniques for Creative Design", The Art of Human-Computer Interface Design, Addison-Wesley, pp. 17-30, 1990

27. [Nielsen 1994] Jakob Nielsen, "Report From a 1994 Web Usability Study", URL: http://www.useit.com/papers/1994_web_usability_report.html

28. [Nielsen1 2001] Nielsen Norman Group, "E-commerce User Experience: High Level Strategy Guidelines", Report in the Collected E-Commerce Design Guidelines URL: http://www.nngroup.com/reports/ecommerce/strategy.html, 2001

29. [Nielsen2 2001], Nielsen Norman Group, "E-commerce User Experience: Design Guidelines for International Users", Report in the Collected E-Commerce Design Guidelines URL: http://www.nngroup.com/reports/ecommerce/international.html, 2001

30. [Nisbett 1977] R. Nisbett, T. Wilson, "Telling More Than We Can Know: Verbal Reports on Mental Processes", Psychological Review, vol. 84, pp. 231-259, 1977

31. [Reid 1980] B. K. Reid, "Scribe: a Document Specification Language and its Compiler", Ph.D. thesis, Carnegie-Mellon University, 1980

32. [Rich1 1979] Elaine A. Rich, "User Modeling via Stereotypes", Cognitive Science, 3, pp.329-341, 1979

33. [Rich2 1983] Elaine A. Rich, "Users Are Individuals: Individualizing User Models", International J. Man-Machine Studies, vol. 18, pp. 199-214, 1983

34. [Riecken 2000] Richard Riecken, "Growth in Personalization and Business", Communications of the ACM, vol. 43, issue 8, pp. 32, Aug. 2000

35. [Ripper 2000] Pedro S. Ripper, Marcus Felipe Fnatoura, Ayrton Maia Neto, Carlos Jose P. de Lucena, "V-Market: A Framework for e-Commerce Agent Systems", World Wide Web (WWW), Baltzer Science Publishers, vol. 3, No 1, pp. 43-52, URL: http://www.almaden.ibm.com/cs/people/fontoura/papers/www2000.pdf, 2000

36. [Roeland 1999] Dinand Roeland, "Trading Agents", Caput Technical Computer Science, URL : http://www.cs.rug.nl/~ben/agents/trading-agents.html, 1999

37. [Rohn 1998] Janice Anne Rohn, "Creating Usable E-commerce Sites", StandardView Vol. 6, No 3, pp. 110-115, September/1998
38. [Schroeder 2001] Michael Schroeder, "Software Agents – introduction", URL: http://www.soi.city.ac.uk/~msch/crs/mas/slides/1intro/index.htm
39. [Timmers 1998] Paul Timmers, "Business Models for Electronic Markets", Electronic Markets, vol. 8 No. 2, pp. 3-8, 1998
40. [Wood 1998] Scott D. Wood, "Usability First: GOMS", Diamond Bullet Design, URL: http://usabilityfirst.com/goms/, 1998