Knowledge Acquisition

for Dynamic Personalization in E-Commerce

Alina Andreevskaia

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

May 2003

Your file Votre référence
ISBN: 0-612-83895-1
Our file Notre référence
ISBN: 0-612-83895-1

ABSTRACT

Knowledge Acquisition for Dynamic Personalization in E-Commerce

Alina Andreevskaia

Information technology is playing an increasingly important role in today's world. Commerce through Internet is not an exception to this phenomenon. Currently the focus in the retailer e-commerce is shifting toward catering to the needs of repeat customers by offering them more personalized services. One of the barriers to such an individualized approach to each customer is the difficulty of collecting information about individual users. This thesis addresses this *knowledge acquisition* problem.

Based on a thorough analysis of different kinds of knowledge acquisition tools and techniques, we propose an architecture that allows the use of a combination of different approaches for knowledge acquisition about users in e-commerce. This architecture is designed to support dynamic adaptation of the user profile to changes in the user interests as well as in the store. The architecture is based on two core concepts, namely dynamic personalization and software agent-support. To reduce the time and effort put in the process of knowledge acquisition by the user and by the knowledge engineer software agents in the proposed architecture assist in different aspects of the process, such as profile initialization, processing of results discovered by the web mining, making changes to user profile and tracking their effects, and in trust related issues and interaction with other agents and systems. A proof of concept prototype has been implemented to demonstrate the feasibility of the architecture.

ACKNOWLEDGMENTS

I would like to express my deepest respect and gratitude to my supervisor Dr. T. Radhakrishnan for his wise guidance and valuable support at all times.

I also would like to thank my colleague, Rony Abi-Aad for sharing with me his ideas on User Modeling and for fruitful collaboration on several publications.

Financial support for this work has been provided by the Canadian Institute for Telecommunication Research special project on Highly Qualified Personnel Training (HQP) through a research grant to Dr. T. Radhakrishnan. The support from CITR is greatly appreciated.

Thanks to my family and friends for their help and support.

And I am very grateful to all my professors who taught me in these years and to all the people who gave me their opinion and suggestions regarding this work.

TABLE OF CONTENTS

LIST OF FIGURES	**vii**
LIST OF TABLES	**viii**
1 INTRODUCTION	**1**
1.1 The context – Electronic Commerce	1
1.2 The challenge – Dynamic Personalization	5
1.3 Objective and Scope of the Thesis	8
1.4 Organization of the Thesis	9
2 KNOWLEDGE ACQUISITION TOOL AND TECHNIQUES	**10**
2.1 Knowledge Acquisition in Psychology and Market Research	10
2.1.1 *Some Popular Psychological Methods*	10
2.1.2 *Knowledge Acquisition: Market Analysis Perspective*	15
2.2 Knowledge Acquisition: the AI Perspective	19
2.2.1 *Classical Tools: Brief Historical Overview*	20
2.2.2 *Modern Knowledge Acquisition Tools*	24
2.2.2.1 Knowledge Elicitation Tools	24
2.2.2.2 Knowledge Capture Tools	38
2.3 Conclusions	45
3 AGENT-SUPPORTED DYNAMIC PERSONALIZATION: TWO CORE CONCEPTS	**51**
3.1 Dynamic User Profiling	52
3.2 Agent Support	56
3.3 Conclusion	67

4 THE PROPOSED KNOWLEDGE ACQUISITION TOOL FOR DYNAMIC USER PROFILING IN E-COMMERCE **68**

4.1 The User Model 68

4.1.1 *What are the User Needs Today?* 68

4.1.2 *Content and Structure of the User Model* 73

4.2 Knowledge Acquisition for User Model 80

4.3 Implemented Prototype of the Knowledge Acquisition Tool for E-Commerce 96

4.4 Conclusion 101

5 CONCLUSIONS **104**

5.1 Contributions 104

5.2 Future Work 105

REFERENCES **107**

Appendix A. MAIN COMPONENTS AND FUNCTIONALITIES OF THE PROOF-OF-CONCEPT PROTOTYPE **118**

LIST OF FIGURES

Figure 1.1. Typical architecture of a personalization system (from [8]) 4

Figure 2.1. Examples of knowledge acquisition tools linking application tasks and problem-solving methods (from [30, p. 259]) 21

Figure 2.2. Personalization Framework utilizing Domain Knowledge (from [55]) 40

Figure 2.3. WUM architecture (from [53]) 44

Figure 2.4. Types of KA tools in historical perspective 48

Figure 4.1. Positioning of the PUMe in the 3D classification space of [80] 72

Figure 4.2. A fragment of a user model with frames representing product features and user preferences 77

Figure 4.3. Proposed architecture of the KA sub-system 85

Figure 4.4. Framework for Use Case 1 89

Figure 4.5. Framework for Use Case 2 92

Figure 4.6. Framework for Use Case 3 94

Figure 4.7. Framework for Use Case 4 96

Figure 4.8. Registration page 97

Figure 4.9. Personalized product selection at login 98

Figure 4.10. Advice for an unknown user 99

Figure 4.11. Customized advice for a parent 99

Figure 4.12 Validation question 101

Figure A.1. Home Page 120

Figure A.2. Search Results Window 121

Figure A.3. Checkout Page 122

LIST OF TABLES

Table 2.1 Some Knowledge Acquisition tools 50

Chapter 1

INTRODUCTION

"If I have 3 million customers on the web, I should have 3 million stores on Web"

Jeff Bezos, CEO of Amazon.com ™

"A new conception of markets emerges, one that recognizes the obvious: that every individual customer is a market of one"

James H. Gilmore and B. Joseph Pine II

1.1 The Context – Electronic Commerce

Information technology is playing an increasingly important role in today's world. Commerce through Internet is not an exception to this phenomenon. According to Statistics Canada "from January to December 2001, an estimated 4 million households, about one-third of all households in Canada, had at least one member that used the Internet to support purchasing decisions, either by window shopping or by placing online orders" [1].

Servicing the existing customer base of a retail website seems to be a strategy that is gaining popularity, as well as larger shares of online marketing budgets. According to a May, 2001 report from the Boston Consulting Group (BCG), successful retailers translate their brands to the Web in many ways, including "recreating the best parts of the in-store experience, including the services of a great salesperson or agent." Such tactics are aimed squarely at retention and generating incremental sales from existing customers [2].

The history of electronic commerce started in 1995 when the first shopping malls arrived on the Internet. The growth of the electronic commerce was impressive. December of 1998 showed how big the e-commerce potential was: Amazon made above $1 billion in annual profits, AOL generated more than $1.2 billion in just 10 weeks. Predictions were very optimistic, e-commerce became highly popular among businesses and consumers, having an on-line presence became a must for businesses. Even though, later on, the growth started to slow down - the $12 billion spending during the 2000 holiday season was followed by a relatively small increase in 2001 holiday season ($13.8 billion) - the electronic commerce is still very popular among retailers and consumers. At the same time the market becomes more aggressive and now the companies that sell on the net need to have more than a nice-looking web-site to attract and keep customers. In their quest for competitive edge Internet retailers are turning more and more towards the benefits of personalization of their interactions with customers, mostly in the form of customized presentation or recommender systems.

In early 1990s the mass customization was understood as “variety and customization” instead of standardization by Joe Pine [3]. He suggested a few years later to apply mass customization not to the products themselves but to their presentation to the individual user, customizing the *consumer experience* [4].

With the advent of e-commerce a new discipline called “Customer Relationship Management” (CRM) has appeared and quickly moved to the forefront. CRM covers methodologies, strategies, software and other web-based capabilities that are used for managing enterprise relationships with customers. CRM "allows companies to gather and access information about customers' buying histories, preferences, complaints, and other data so they <the companies> can better anticipate what customers will want. The goal is to instill greater customer loyalty." [5]

As stated by Thomas H. Davenport (Boston University; director of the Accenture's Institute for Strategic Change) two trends have made CRM especially important in the recent years:

- With the increase of the competition and increasing difficulty of differentiating between products, "companies have begun moving from a product-centric view of the world to a customer-centric one".
- Advances in the technology have made it possible to put customer information from all over the enterprise into a single system. "Until recently, we didn't have the ability to manage the complex information about customers, because information was stored in 20 different systems," says Davenport. But as network and Internet technology has matured, CRM software has found its place in the world "[6].

Another stimulus for personalized services came from the need to manage the information overload. The expansion of the World Wide Web made finding and sorting all the available information beyond human ability. Understanding of this phenomenon gave rise to the idea of pre-selecting, pre-processing of the information delivered to the user. To ensure better precision of the information retrieval it became necessary to better understand the customer/user for whom the information was intended. This is the motivation behind many non-commercial recommender systems.

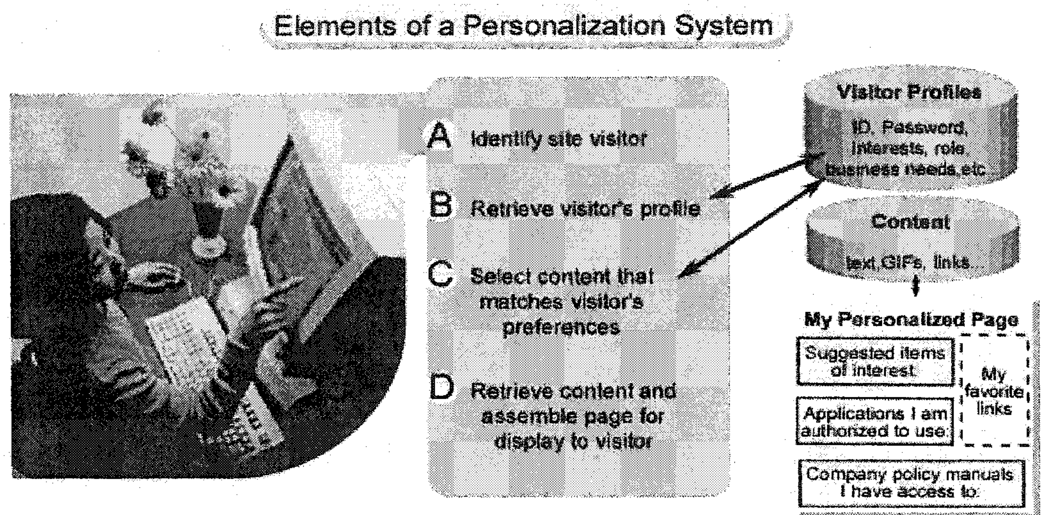

Figure 1.1. Typical architecture of a personalization system (from [8])

Many Internet companies outside e-commerce introduced personalization as part of their services: My Yahoo! (the customizable Yahoo! launched in 1996) was the first large scale personalization effort [7]. Since the mid-90s, a lot of work has been done to encompass the information about a consumer, or rather, about consumer groups, into virtual shopping malls. Figure 1.1 (from [8]) shows a typical architecture of a personalization system. The architecture is built around the user information that is usually interpreted as interests of personal kind, business needs, access rights, and user's role in the organization. What a typical architecture, such as the one in Figure 1.1., do not show, is how this information has been collected, from what sources the system acquires the knowledge about and keeps track of changes in the user's needs and interests. In brief, such an architecture lacks a major component – a knowledge acquisition module for personalizing the user model (or PUMe). This thesis will show how PUMe can be included in the system and what role it plays in dynamic user profiling and how software agents could support an end-user and a Knowledge Engineer in the process of continuous knowledge acquisition.

1.2 The Challenge – Dynamic Personalization

The idea of personalizing the consumer experience has become very popular in recent years. Many e-commerce sites already incorporate user-modeling capabilities for the purpose of personalizing the interactions, targeting advertisements and recommending products [amazon.com], [garden.com]. Major commercial software packages for e-commerce sites and portal developers often provide personalization capabilities as a standard feature (e.g. IBM Net Commerce [http://www-3.ibm.com/software/webservers/commerce/wcs_pro/], ATG Dynamo [www.atg.com], BroadVision [www.broadvision.com], and others). Older versions of these technologies (e.g., those introduced by engage.com and personify.com in 1999-2000) built only on factual information (age, gender, shoe size, maximum amount of purchase, etc.), later other companies (e.g., Art Technology Group and BroadVision) started to include rules describing user's on-line behavior that are developed by experts in the specific area of marketing[1]. The main purpose of personalization adopted by most on-line stores has been

[1] [9] contains a good overview of these trends.

to keep customers longer in the 'virtual' or electronic store, or to attract more visitors to the site. As shown in [8] and [10], we notice a shift in emphasis on personalization towards catering to the needs of repeat customers as well. Encouraging the shoppers to return to an e-commerce web site, is beneficial and challenging. Special promotions and discounts or customized contents (information about specific products deemed to be interesting for this particular customer, different level of details for different users, etc.) and presentations can be used to achieve this goal.

There are many commercial products that allow achieving some degree of personalization of an e-commerce site but they lack the following features:

- <u>address the individual customer's needs.</u> Standard recommender systems employed in most e-commerce web-sites are trying to match the current user to a set of group profiles and then customize the presentation based on the systems knowledge about these general groups of customers. This approach of grouping customers into classes was satisfactory for the original intentions, namely for recommending books and movies where lack of accuracy is not very crucial. However, it is not satisfactory when taken out of this context to products that do not allow for straightforward clustering (e.g., grocery shopping). In addition, not everyone is a "follower" and some people need to feel special and be treated as individuals rather than members of some indeterminate group.

- <u>adapt dynamically to the changes in the user's behavior.</u> Most user models are static and do not reflect changes in the user's situation (e.g., having children, kids growing up, moving to a bigger house, weight loss/gain, changes in health status or in financial situation, etc.) that may occur over time. In some cases such changes may lead to dramatic changes in the shopping behavior, choices that are made, etc. Some products are more sensitive to these changes than others – e.g., choice of toys will follow the children's growth pattern; contents of the grocery bag will be even more dynamic and will depend on special occasions, visitors at home, health, etc. Ideally, the system should be able to extract (if not predict) such changes and react to them appropriately. Some research in the area of web data mining related to this will be reviewed later.

One of the reasons the personalization of e-commerce customer experience is not satisfactory is the challenge of collecting the information about the customer that is relevant to dynamic changes without

1) compromising his/her sense of security/privacy;
2) asking for too much direct input;
3) overloading the user with unnecessary recommendations;
4) slowing down the interaction;
5) taking too much resources on the server.

This thesis will address the first three problems: how to get maximum information about the user while keeping his/her active participation to a minimum and using only information, the user has already agreed to share. The technical problems of managing time and space constraints are out of the scope of this work.

1.3 Objective and Scope of the Thesis

In this thesis, we

- present an overview of classical and modern tools and methods for (semi-) automatic knowledge acquisition;
- propose an integrated architecture for PUMe (personalizing the user model) module, that will combine different methods and techniques;
- describe a proof of concept prototype of the system implemented as a feasibility study.

The goal of the work is to explore better means of acquiring knowledge about users on an ongoing basis, to be able to individualize recommendations, presentation, advertisement and proactive notification.

Throughout the thesis, a running example of a small wellness boutique will be used to illustrate the ideas and their implementation.

1.4 Organization of the Thesis

Chapter 1 outlined the goals of the work, its motivations and the context of electronic commerce in which the thesis is written[2].

Chapter 2 will present a survey of knowledge acquisition tools and techniques, from a brief historical overview to most recent trends. We classify the tools into two groups: "classical" and "modern". This chapter will explain the rationale behind this classification and give examples of tools belonging to the two categories.

Chapter 3 presents the two core concepts considered relevant for the proposed model: dynamic personalization and agent supported knowledge acquisition.

Chapter 4 will describe the architecture of the tool based on the analysis performed in chapter 3, tool's components, as well as methods and algorithms used by the system.

Chapter 5 will summarize the work and outline future directions of research.

---

[2] Thesis work was supported by CITR special project on Highly Qualified Personnel Training (HQP).

Chapter 2

KNOWLEDGE ACQUISITION TOOLS AND TECHNIQUES: AN OVERVIEW

Acquiring knowledge about customers is a well-known problem that has been explored by marketers working for brick-and-mortar retailers for many years[3]. With the advent of electronic commerce in the mid 1990s, new tools and techniques have been invented or new variations of traditional approaches have been developed. This chapter will start with a very brief survey of traditional market research techniques that in some form are already used or can be used in electronic commerce. The second part of the chapter will concentrate on existing knowledge acquisition tools that have been developed in the area of expert systems and in the domain of web-based services.

2.1 Knowledge Acquisition in Psychology and Market Research

2.1.1 Some Popular Psychological Methods

Multiple methods (naturalistic observation, surveys including questionnaires, case studies with interviews and tests, different experimental methods) have been developed in psychology for collecting information about people either as individuals or as members of social groups. Many of these methods have been applied in knowledge acquisition in the area of AI (a good detailed overview of KA methods can be found in [11] and [12]).

---

[3] First marketing studies were conducted in the 19th century, since 1930th market research was relying on formalized theory.

Some of these techniques (e.g., protocol analysis or structured interviews) are well-known and do not require further explanation here. The card sorting method and methods based on personal construct psychology are less known but are quite popular; therefore they are briefly described below.

Personal Construct Psychology (PCP)[4] was developed in 1950-s by George Kelly, a psychologist and a geometer. The system is very popular and is used in different domains such as management studies, artificial intelligence, and other disciplines.

PCP is based on a very subjective approach to human psychology. A person is considered to have a completely free choice of what she likes or prefers. The person is free to reinterpret events of the past and is a "personal scientist" in events anticipation. An individual elaborates his own ways of seeing the world through constructs. He creates constructs and tries them. Constructs are organized into systems. More than one system can be applied to the same event. Despite the subjective nature of this method, Kelly did not completely deny social factors and differentiated between individuality ("persons differ from each other in their construction of events" [18]), communality ("to the extent one person employs a construction of experience which is similar to that employed by another, his psychological processes are similar to those of the other person" [18]), sociality ("to the extent that one person constitutes the construction processes of another, he may play a role in a social process involving another person" [18]). The last two categories can serve as a foundation for comparative studies of individualities.

[4] Mildred L.G. Shaw and Brian R. Gaines maintain a web site on PCP: http://repgrid.com/pcp/

The main concept of Kelly's PCP is a construct – a verbal label that represents a conceptual distinction made by an individual. The idea of a construct is based on the understanding that humans think in terms of contrasts and dichotomies. Constructs serve as tools to replace events in individual's imagination and they are used as building blocks in construction of a view of the world. Constructs have limited applicability ("construct's range of convenience" which "comprises all those things to which the user would find its application useful" [18]) and an area where they fit the best ("construct's focus of convenience" which "comprises those particular things to which the user would find its application maximally useful" [18]).

Kelly developed a new technique for capturing and exploring constructs – a *repertory grid* (repgrid) technique. This technique invented by Kelly to bypass cognitive differences allows the analyst to obtain individual information using standardized tools. This combination of a uniform representation and a standard acquisition method with ability to capture individual characteristics can be very helpful when it is necessary to acquire and compare individual profiles. The efficiency of repgrid technique was proved by more than 50 years experience in psychology and other disciplines. WebGrid and PC PACK tools described below (in Section 2.2.2.1) use a computerized version of PCP repgrid for eliciting knowledge.

Laddering is a variation of the PCP method where the elicited concepts are organized into a hierarchy such as class hierarchy or ontology [19]. It is employed in PC PACK.

Card sorting is a technique applied in different variations in marketing research, web design and artificial intelligence. Traditionally the card sorting exercise is conducted using paper cards or sticky (post-it) notes; rubber bands are used to hold the cards together. In recent years some software products were developed to conduct the entire card sorting exercise on a computer.

The card sorting exercise can be subdivided in several steps:

- First, the organizers have to decide upon items to be characterized and choose their labels. The labels are usually written or printed on individual cards in large print.
- Participants are recruited from the intended user population. The size of the subject group depends on the kind of project as well as on the budget of the developer organization. It is usually suggested that there are at least 6 participants to make results of the exercise representative. The upper bound may vary from 10 to 50 users depending on the complexity of the structure and vagueness of the concepts: very complex structures or vague concepts as well as multiple user segments necessitate relatively larger number of samples.
- Each participant is given a set of cards that he/she is asked to group according to his/her understanding. Groups have to be labeled by each participant. Usually each participant works individually, however, some discussions between the subjects may be allowed.

The results of the card sorting exercises are analyzed using different techniques. If the number of participants is small and differences in groupings are rare simple eyeballing may be enough. In most cases more sophisticated techniques are applied: the data is entered in a spreadsheet that is analyzed using statistical methods and cluster analysis techniques. Various software packages can help in this process[5].

The results of the card-sorting exercise include one or more hierarchical representations of the set of concepts that will be used to develop a structure from this set, a set of labels for groupings, as well as possible changes to the original labels. In some variations of this technique the users may also be asked to answer some simple questions or to make suggestions.

A detailed descriptions of how to conduct a card sorting exercise can be found in [13], [14], [15] and others.

There are two major software packages available to perform the card-sorting exercise on computers: IBM EZSort [16] and National Institute of Standards and Technology WebCAT [17]. The card sorting technique is also included as a part of the PC PACK knowledge acquisition tool described in Section 2.2.2.1.

[5] Statistical Package for the Social Sciences – SPSS – seems to be one of the most popular.

2.1.2 Knowledge Acquisition: Market Analysis Perspective

The research on consumer preferences, perceptions and, more generally, buying behavior encompasses different methodologies that can be briefly described as follows [20].

The two major components of modern market research present in practically every major market research project are known as secondary and primary research. Contrary to the intuitive order (first primary then secondary), the secondary research that involves search of information through all available public sources and industry reports ("secondary" data sources) is usually conducted first as an exploratory phase preceding much more expensive primary research. The primary research involves a contact with a consumer (usually sampled from a target population) in order to elicit from him/her information that the researcher could not obtain through the secondary data search. This direct contact with the consumer can take a number of forms – from Internet or mail-in questionnaire to individual interviews or focus groups, where a consumer or group of consumers has a discussion on a certain issue, product or consumption pattern. This is conducted by a specially trained interviewer / mediator. The primary research is usually associated with significant costs and requires special considerations with respect to balancing between accuracy vs. level of detail vs. cost.

The following is a list of main methods of primary market research, followed by a brief discussion of their advantages and disadvantages.

Questionnaires

While the traditional **mail-in questionnaire** technique is not very suitable for e-commerce, a more modern version of this approach – the **Internet questionnaire** – is often applied by on-line retailers. The advantages of this techniques are:

- it involves no printing and mailing costs,
- it is easy to change if needed even after the study has started,
- it has no costs associated with data entry,
- it provides a fast response.

In all other respects, Internet market research shares the drawbacks of mail-in questionnaire: low response rate and selective bias. As the use of the Internet expands and more households are linked to the Internet, the selective bias becomes a bit less of a problem, but still the researcher does not have control over who within a household or business, responds to his questions: is it a person in charge of buying decisions, his "influencer" (e.g., a child who wants a toy) or just someone who does not belong to the targeted consumer group but is attracted by a free incentive? Some special techniques are used in questionnaire design to provide for a way to identify respondents who do not belong to the targeted consumer group or did not respond to the questions with the necessary diligence.

Interviews with a consumer

Interviews with consumers can supply valuable information that can then be processed using knowledge engineering software tools described in Section 2.2.

Interviews with a consumer are performed in a variety of different ways: telephone interviews, door-to-door and executive interviews, mall intercept interviews, purchase intercept interviews. The last technique has been applied not only in the brick-and-mortar setting, but also in virtual shopping environment. **Purchase Intercept interviews** are administered at the time of observable product selection by a consumer, which provides for better consumer targeting and can give good insights in respect to stimuli in consumer's buying behavior. In the case of Internet shopping this technique can be replaced by "observation" of consumer actions where a record of shopper's choices while browsing can serve as data for building his/her user profile. At the same time, customers can be asked a few additional questions before they check out from the virtual store.

Focus Groups

Focus group is a group discussion focused on a particular topic or issue introduced by a mediator; the group members are encouraged to express their own views on the topic and elaborate or react to views of each other. This is the ultimate "qualitative" method of market research that does not provide any statistically significant results but can give a researcher a good insight into consumers' reactions, motives, thought processes, and to provide the researcher with new ideas in respect to the possible potential improvements to the product, its image or advertisement. Sometimes a limited number of focus groups is conducted (from two to five) as a part of exploratory primary research that precedes more expensive large-scale surveys with face-to-face interviews. This method is also actively used when researchers are interested in in-depth discussion of a product or an issue with consumers rather than assessment of what percent of the target market would buy product X.

This technique can be most easily combined with sophisticated computer tools for knowledge elicitation because focus group members will be more ready to commit their time to learn and use a new computer tool such as WebGrid.

As a way to cut costs of their study, many researchers favor **omnibus studies** that are conducted on a regular basis by market research companies and include questions from different clients. The method of data collection may vary: it can be a questionnaire or an interview. Given its obvious cost advantage, omnibus market research studies and multi-client reports are usually preferred over custom-designed studies. In the case of electronic commerce, omnibus studies can be conducted for a virtual mall that offers diverse products at the same Internet location.

Thus, the usual algorithm of data collection in market research proceeds from least expensive secondary data search to omnibus studies, if secondary data is not available. Alternatively, it can proceed directly to custom designed primary research studies that can provide for greater breadth and depth of research than omnibus studies but are much more expensive. The choice of a particular method of research depends heavily on the objectives of the study, target audience, budget constraints and required accuracy/statistical significance of the results. The selected method and study design is usually a tradeoff that takes into account many conflicting requirements.

The major downside of this kind of knowledge acquisition method is the amount of time and human effort. For example, EnviroSell Inc. analyses 14,000 hours of store videotapes a year to do their marketing research [21]. This means, that while marketing science can supply the general methodologies for behavioral research necessary for user profiling in e-commerce, it is not possible to fully depend on these methods due to their inefficiency, inaccuracy, and high cost both in terms of time and money.

A similar problem is well known in the area of Artificial Intelligence and is known as "knowledge acquisition (KA) bottleneck".

2.2 Knowledge Acquisition: the AI Perspective

The term "Knowledge acquisition bottleneck" was coined by Feigenbaum in 1984 [22] but first attempts on building knowledge acquisition tools that would replace an intermediary in the KA process were made since late 1970s. This effort, however, was mostly driven by the needs of knowledge engineers. Their aim was to reduce the cost of building knowledge bases rather than to make the knowledge elicitation process user-friendly for non-programmers. Recently, though, the situation has changed and several successful attempts were made to shift the focus toward end user's convenience. Based on this observation, we divide the KA tools into two major types:

- Classical KA tools treat the computer as a support tool that "serves" the needs of specially trained specialists (knowledge engineers or domain experts) by facilitating data entry and performing some error checking which we call "low-level operations".

- Modern KA software is more proactive, it is more process and logic oriented and often provides enough guidance and advice to allow a naïve user to perform necessary knowledge acquisition activities.

The following sub-sections will present several tools from both classical and modern types of KA tools with more emphasis on the latter. The scope of the analysis is limited to systems that are described in publicly available literature with sufficient level of detail[6].

2.2.1 Classical Tools: Brief Historical Overview

Knowledge Acquisition tools that help a knowledge engineer in acquiring the knowledge from experts and transferring it to the knowledge base have been created since late 1970s: TEIRESIAS [23], [24], OPAL [25], SALT [26], MOLE [27], KSSn [28] and many others. The best way to present these tools, most of which are well known and well described in the literature, is to show what groupings they form and how they can be classified. To do so, two classifications of knowledge acquisition tools will be shown

[6] There exist also several commercial KA tools on the market that claim the ability to capture knowledge in interaction with the end-user (for example: KAT, AKAS, ACQUIRE®). However, it is difficult to understand what functions these software products actually perform because information about them is coming only from advertisements and no demos are available.

here – one by John H. Boose dating back to 1989, and a recent one by Yolanda Gil and Jihie Kim done in 2002.

In 1989-1990 John. H. Boose ([29], [30]) suggested a classification of the KA tools that existed by that time, based on the tasks and problem solving methods they were using. This classification gives a good picture of the "classical" knowledge acquisition tools (Figure 2.1).

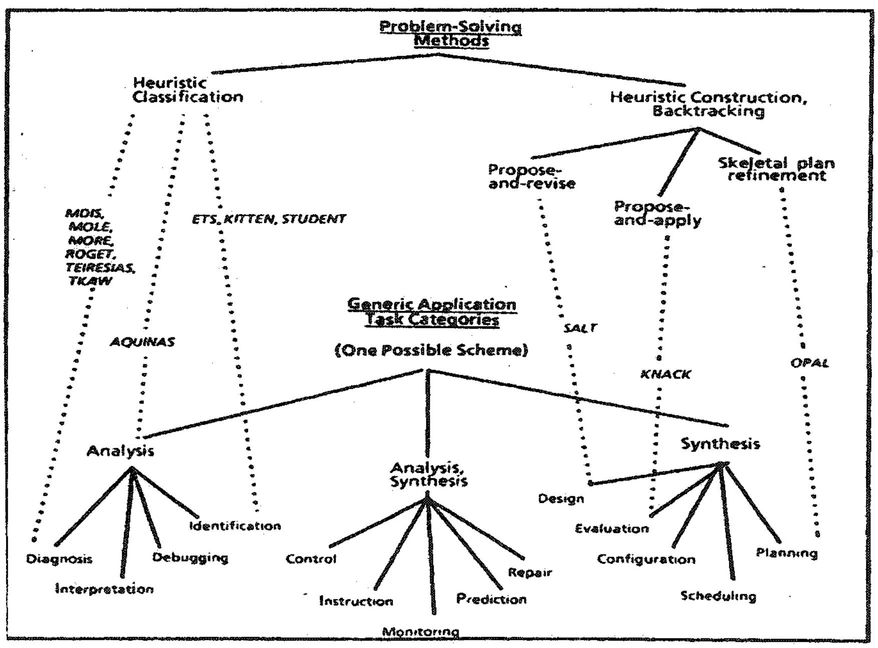

Figure 2.1. Examples of knowledge acquisition tools linking application tasks and problem-solving methods (from [30, p. 259])

The tools have been differentiated based on the problem-solving methods and they are divided into KA tools supporting systems that use heuristic classification (such as TEREISIAS or MYCIN) and that use heuristic construction (such as OPAL for

ONCOCIN). This classification is further refined by applying task categories such as analysis, synthesis or the combination of the two. These KA tools were designed to facilitate the task of the knowledge engineer and to speed up the knowledge acquisition process by automating some parts of the processing, e.g. using GUI for constraining expert's input in OPAL [25], implementing consistency checks in TEREISIAS [24], taking advantage of domain knowledge for guiding the expert in MORE [31] for MUD, and using other techniques.

A more recent overview of selected interactive knowledge acquisition tools can be found in [11]. The set of tools covered in this study includes some most popular classical tools such as TEIRESIAS and SALT as well as a number of more recent systems: two Soar systems - TAQL [32] and Instructo-Soar [33], [34], CHIMAERA [35], PROTOS [36], SHAKEN [37], SEEK2 [38], KSSn [28]. These tools are analyzed from the point of view of the types of knowledge and meta-knowledge they use. This classification breaks the set of selected KA tools into six overlapping categories:

- General problem solving and task knowledge (SALT, TAQL)
- Prior domain knowledge (EXPECT, INSTRUCTO-SOAR)
- General background knowledge (SHAKEN)
- Example cases (INSTRUCTO-SOAR, PROTOS, SEEK2, SHAKEN, TEIRESIAS)
- Underlying knowledge representation (KSSn, SEEK2, TAQL, TEIRESIAS, CHIMAERA)
- Diagnosis and debugging knowledge (CHIMAERA, EXPECT, TEIRESIAS).

The classical knowledge acquisition tools were an important step in the process of development of knowledge acquisition software and they have advantages as well as disadvantages.

Advantages:

1. Providing support for professional knowledge engineers and specially trained domain experts.
2. Speeding up significantly the process of knowledge acquisition.
3. Providing background for development of more sophisticated software.

Disadvantages:

1. Need for specially trained users.
2. Imperfect user interface – command-line for older tools or simple GUI for newer applications.
3. Support limited to providing forms for faster input and to checking new input against existing knowledge (e.g. using meta-rules in TEIRESIAS; based on domain model in MORE) to minimize errors and to providing forms for faster input.
4. Inflexibility due to the fact that such systems were usually created as additional modules for already existing systems. These tools were domain and system-dependent and had very limited flexibility and portability.

In the following part some of the most recent of the above-mentioned tools (KSSn and EXPECT) will be reviewed from a different point of view; namely how they can be used for knowledge acquisition in electronic commerce.

2.2.2 Modern Knowledge Acquisition Tools

Currently the automated knowledge acquisition based on machine learning and data mining techniques is gaining popularity [39] and should be considered along with the more traditional work on interactive semi-automated tools for knowledge acquisition. To clearly differentiate between the two approaches we use the following terminology: knowledge elicitation tools are the tools that mainly focus on acquiring information from people while knowledge capturing tools rely on machine learning and data mining.

2.2.2.1 Knowledge Elicitation Tools

In this thesis a number of most recent (semi-)automated knowledge acquisition tools is analyzed. They are summarized in Table 2.1 (p. 50).

[40] states that "In recent years researchers have investigated a variety of promising approaches to knowledge acquisition (KA), but they have often been driven by the needs of knowledge engineers rather than by end users." The KA tools reviewed in this section include both types: those supporting the knowledge engineer (e.g. PROTÉGÉ 2000 or PC PACK) and those supporting the "naïve" user (EXPECT, WebGrid).

The tools under discussion include: the KSSn tool WebGrid; PROTÉGÉ 2000; PC PACK; and EXPECT. These tools have been selected out of many others as most promising in terms of their potential applicability for knowledge elicitation for e-commerce. The following analysis will show to what extent this original assumption was true and if any of these tools can be embedded in PUMe. Each tool will be briefly summarized, then described in more detail, and finally, its usefulness for our purposes will be discussed. Particular requirements of electronic commerce will be taken into consideration, among them the need to allow a naïve user without any special training or experience to use the tool, to minimize the amount of work required from knowledge engineer or expert after the tool has been installed, to provide fast and inexpensive service, and to allow easy interface with other programs including web applications.

PROTÉGÉ 2000[7]

Assists a Knowledge Engineer or a domain expert in knowledge acquisition and representation (Object Oriented, allows for other representations via multiple plug-ins).

Protégé 2000 is a tool for knowledge base development created at Stanford Medical Informatics. It is the latest version of the Protégé project. Originally, Protégé was created in 1989 and was based on ONCOCIN. This first version had very limited applicability and was restricted to the medical domain. The next version of the project – PROTÉGÉ II – appeared in 1991. It became one of the most well-known knowledge acquisition

[7] Major sources of information about PROTÉGÉ 2000 are http://www.smi.stanford.edu/courses/mis301/protégé/ and http://smi-web.stanford.edu/projects/protege and works by Marc Musen and his colleagues (e.g. [41])

systems and was used for different medical domains all over the world. In 1995 it was adapted for Windows (PROTÉGÉ/Win). The newest version of the system is PROTÉGÉ 2000. [8]

PROTÉGÉ 2000 is a set of tools for knowledge-based systems development. It has a very user-friendly interface which allows a user, who can be a knowledge engineer or, more often, a domain expert with high degree of computer literacy, to create a knowledge based system starting from ontology and ending with a sophisticated knowledge acquisition system. The use of the interface is intuitive and straightforward. Excellent technical support is provided via protégé-discussion mailing list. The tool was designed in conformance with OKBC (Open Knowledge Base Connectivity) protocol[9]. PROTÉGÉ 2000 consists of a set of knowledge-base components joined together by a default configuration. It uses a layered approach with three layers: Widget layer (user interface), Control layer (to handle standard actions and connections between the other two layers), and Knowledge-base server (a wrapper around the actual knowledge-base server).

PROTÉGÉ 2000 allows to work with classes, slots and instances at the same time. There is no strict hierarchy of these concepts in the current version of PROTÉGÉ: instances can

[8]The history of the PROTÉGÉ project is described in detail in [41] available at http://smi-web.stanford.edu/pubs/SMI_Abstracts/SMI-1999-0801.html

[9] "Open Knowledge Base Connectivity (OKBC) is an application programming interface for accessing knowledge bases stored in knowledge representation systems (KRSs). OKBC is being developed under the sponsorship of DARPA's High Performance Knowledge Base program (HPKB), where it is being used as an initial protocol for the integration of various technology components." [42]

be used to represent classes, classes can be stored as instances if needed. PROTÉGÉ 2000 works with projects that incorporate everything from ontologies to forms layout.

The system has a uniform "tabbed" GUI: each screen is presented as a number of overlapping tabs. It was assumed that a user goes in cycles while creating a knowledge-based system and a typical cycle is as follows: an ontology is built, then a KA tool is generated, this tool is used to build the initial knowledge-base, then the ontology may be revised, the KA tool updated, and the initial knowledge-base is changed until the user is ready to build a test-case knowledge-base, test it, and is satisfied with the knowledge base.

The interfaces consist of

- The classes tab – a window for viewing, creating, and editing classes and their relationships, which model the domain concepts.
- The slots tab – a window for viewing, creating, and editing slots[10]. In this version of PROTÉGÉ slots are independent of classes.
- The forms tab – a window for viewing and editing prototype forms. The finished forms will be then employed by end users for entering instances into the knowledge base.
- The instances tab – a window for viewing, creating, and editing instances (i.e. actual data). The user can see the class where the instance belongs and a list of other instances of the same class.

[10] In the Protégé 2000 project slots are understood as attributes of a class, but at the same time " they can be defined and manipulated independently, and can exist without any relationship to classes" [42].

PROTÉGÉ 2000 has become very popular among knowledge engineers and other specialists working in the field of expert systems. It is a well designed easy-to-use tool for the purpose it is meant for that is to help a computer literate domain expert in knowledge acquisition. The tool was created under the assumption that its user is familiar with such concepts as ontology, class, slot, instance of a class, etc. If a person has the background in artificial intelligence it is easy to learn to use PROTÉGÉ, though it is still necessary to go through the tutorials and user guides to understand how the tool works. (The tool looks much easier to use than it really is).

A domain expert can be relatively easily taught to use the tool by somebody who would walk him/her through a number of examples. This allows a knowledge engineer to be completely excluded from or at least to have his involvement minimized in the process of knowledge base creation.

Trying to adapt PROTÉGÉ for an entirely different user category – naïve end-users – does not seem a good idea. First of all, a naïve end-user will not want to learn about all the theoretical basis of the application (one cannot expect that an elderly housewife shopping for a toy for her newborn grandson would be at ease with ontology or class concepts). Second, the interface that is nice and easy-to-use for a professional who has seen many programming tools and knows how such interfaces are usually organized is not intuitive for a person who knows only how to key in a web address or check boxes in simple forms. PROTÉGÉ can be used for the purposes of knowledge base creation in e-commerce by processing the data collected and stored in other sources such as questionnaires, focus groups, etc. If the input is of a standard form it might be possible to create a separate application that translates the input into a form that might be processed automatically by PROTÉGÉ without human intermediary. However, this will mean changes to the software and will also require a large amount of preliminary work in order to create meaningful templates for the input. To a great extent, PROTÉGÉ is a continuation of the classical approach to knowledge acquisition but taken to the next level of convenience and flexibility. Same comments hold for the PC PACK tool described next.

PC PACK[11]

Helps to organize knowledge (repertory grids, hierarchies of objects, attribute matrices, text annotation, Database creation)

PC PACK was created by Niget Shadbolt and others and is now a property of Epistemics company. The version described below was issued in 2000[12]. PC PACK is a set of integrated tools for requirements and knowledge engineering. PC PACK is based on the KADS methodology[13]. It uses several different methodologies, among them RepGrid with laddering and Card Sorting.

---

[11] [43] and http://www.epistemics.co.uk/products/pcpack/

[12] The latest version – PC PACK 4 – (http://www.epistemics.co.uk/staff/psmart/software/CommonKADS_PCPACK_Ontology/index.htm) – appeared in august 2002 and is conceptually very similar to the previous release.

There are eleven tools in the package:

1. **Protocol Editor.** Text can be marked up – the GUI is an intuitive marker metaphor: objects can be highlighted in different colors. After a set of objects is thus created they can be annotated (using hypertext tool) and organized into classes (using laddering tool).
2. **Hypertext Tool**. Objects can be linked and annotated using this tool. This tool also has an integrated help.
3. **Laddering Tool** serves to organize objects into hierarchies.
4. **Card-Sort Tool** allows objects to be sorted into categories. This tool elicits attributes from user for each sort. It is based on a technique well known in psychology (see Section 2.1.1).
5. **Matrix Tool.** Using this tool user can assign values to attributes. Matrix can be sorted. The interface is designed as a spreadsheet.
6. **Repertory Grid.** PC PACK uses the repertory grid technique to allow users to detect similarity and difference between concepts in a set. It can also be used for finding new attributes.
7. **Rule Editor** allows the user to modify and add rules.
8. **Control Editor** is employed for working with processes. It allows the user to decompose tasks into subtasks.
9. **GDM Workbench** is used for problem-solving models construction following well defined grammar rules. It includes libraries of predefined models. GDM Workbench supports KADS.

---

[13] KADS methodology is a de facto standard in European Knowledge Acquisition. It is described in detail in [44]. A brief overview can be found in [45] and at http://issco-www.unige.ch/ewg95/node103.html.

10. **Entity-Relationship Tool** was designed to create databases (conventional or object-oriented).
11. **Dependency Editor** allows to construct relations and populate database tables. Dependencies can be imported from other components like repertory grid or created by the user.

The usability of PC PACK is very good, the use of intuitive interfaces (e.g., marker metaphor) and traditional techniques (such as card sorting) makes it relatively easy to understand and use. On the other hand, as most other interactive (semi-) automated knowledge elicitation tools, PC PACK is geared for its use by a knowledge engineer or a domain expert with some computer experience (and some help from a knowledge engineer), rather than by a naïve end-user. Despite a convenient interface, this tool is too complex and time-consuming for use as means to collect the information about a user in electronic commerce.

PROTÉGÉ and PC PACK demonstrate how the classical paradigm of knowledge engineer–oriented tool is enhanced and modernized. Other tools reviewed in this section belong to a different approach geared towards naïve end-users.

WebGrid[14]

Helps the expert to represent their (intuitive) knowledge. Not so useful for already structured domains (but can be applied there too)

Mildred L. G. Shaw and Brian R. Gaines have been working on knowledge elicitation tools for several years. Tools designed by them apply a repertory grid technique initially developed in personal construct psychology (PCP, described above in 2.1.1). The works of Brian R. Gaines and Mildred L.G. Shaw – WebGrid and WebGrid II – show how the PCP can be used on the World-Wide-Web. WebGrid is described in details in [47]. The article uses as an example a process of purchasing a home. The user (real estate agent) can customize a HTML screen to include his company's logo and name and pictures of homes under discussion. Then end-users (people who shop for a new house) are presented a series of triads of pictures which they are asked to distinguish using a single parameter. All choices are rated by users on a chosen scale. If two houses have similar ratings, users are asked to find a criterion which will differentiate between them. When a repertory grid, is created users have to describe an ideal home using a set of parameters and ratings. Eventually all houses are compared to the ideal home using FOCUS sorting analysis (grid is sorted so that similar elements are clustered together) and a map is produced by a specially designed ProCom software. This output allows the user to see in

[14] Information about WebGrid and WebGrid II can be found in works by Brian Gaines and Mildred Shaw [46], [47]. An on-line demo of WebGrid II is available at http://tiger.cpsc.ucalgary.ca. Other sources of information about WebGrid include http://tiger.cpsc.ucalgary.ca/WebGrid/WebGrid.html; http://www.brint.com/wwwboard/messages/186.html; http://ksi.cpsc.ucalgary.ca/KAW/KAW96/gaines/KM.html; http://ksi.cpsc.ucalgary.ca/articles/ (contains extensive bibliography)

what way the houses are similar to the ideal and in what they differ from it. Of course, such approach is hardly suitable for a regular e-commerce situation because customers have to spend large amounts of time and effort creating a repertory grid that might never become applicable to a real object (shopping is usually not about formulating the properties of an ideal purchase but about finding compromises between the ideal and reality).

WebGrid II is a variation of the WebGrid. It is a web server which offers remote users knowledge elicitation, modeling, comparison, analysis, and inference services. Users can create their own grids, cache them on the server for a limited time, exchange them and compare them. WebGrid II was part of the Sisyphus IV initiative[15]. The web based version of the repertory grid elicitation tool by Brian R. Gaines and Mildred L.G. Shaw has a user friendly interface and is easy to learn.

The repertory grid technique might seem a natural choice for a knowledge elicitation tool because it allows combining highly individual psychological profiles with standard representation. This allows different repertory grids to be compared automatically (as in WebGrid II). There is, however, one very important disadvantage to this approach if we consider applying it to the study of consumer's shopping behavior. Using this tool (or any other repertory grid based application) requires high commitment from the "subject". It takes quite a while to build a repertory grid, because it is an iterative process where a person has to fill many forms, often going back to refine the constructs. Special

[15] [48] give brief description of all four Sisyphus projects; detailed introduction into the Sisyphus IV can be found at http://ksi.cpsc.ucalgary.ca/KAW/Sisyphus/SisyphusIV.html.

knowledge of the technique is necessary to create a repertory grid, even using the helpful interface designed by Gaines and Shaw. A person has to understand what is required from him/her in this exercise and this means reading a long and detailed user manual – something most e-shoppers would not be happy about. Another very important factor that makes a repertory grid a bad choice for e-commerce is that it requires the user to think, to analyze his/her likes and dislikes, to open up in front of a third party (the fact that this observer is a computer makes it even worse, because the user does not know who is behind the machine and it is hard to create a trust relationship in this situation). Very few people would like to spend time on filling even a simple yes/no questionnaire. To ask them to spend hours trying to build their own psychological profile can scare consumers out of the e-mall altogether. In addition, the user might not want to have his/her further choices constrained by the results of a psychological experiment he/she does not fully understand.

Our conclusion is that the repertory grid technique in the way it is used by WebGrid is not suitable for regular consumer study in e-commerce. However, there are some possible applications for certain types of purchases. The first one was shown by B. R. Gaines and M.L.G. Shaw in the article summarized above. In case of very high involvement purchases and especially where there is some interaction with an agent, like in real estate, consumers are more likely to accept this overhead to help them to make a better choice. It can happen because the final decision is very important, the shopping process takes a lot of time anyway, buyers deal with a particular person and know who and will be using the acquired information and for what.

This technique can be employed for focus groups where people already agreed to commit their time and participate in a marketing study. Focus groups in this case can be conducted in the traditional way, where focus group members are interviewed by a market research specialist or by a psychologist. Internet shoppers are usually quite comfortable with computers and it might be possible to ask them to use WebGrid online after initial introduction by a company representative.

One more possibility to apply the repertory grid technique is to attach an application like WebGrid to a learning agent supplied to the end-user. Such an agent should not be related to a particular vendor but be part of a package the consumer can use for shopping wherever he/she likes. In this case the user will have a feeling that the agent works for him, not for the vendor and once the user becomes familiar with the application he/she can reuse it to build his/her consumer profiles for different kinds of products and occasions. The user will trust the application because it helps him and he knows how the information is used. He/she will be ready to commit more time and effort to this process because now it will be easier for him/her to see the outcome and how it eventually can give him/her a better shopping experience.

EXPECT[16]

Based on some initial knowledge base, EXPECT generates an Interdependency Model (IM) that captures the relationship between components of the knowledge base. It has an intuitive interface for incremental KA.

EXPECT system was developed at ISI (Information Sciences Institute, University of Southern California). It is a development of the Explainable Expert System framework.

The main aspect of the EXPECT system is that domain facts and problem solving principles are explicitly represented in two separate knowledge bases. Each of these two knowledge bases has its own acquisition tools. The Application Programmer is an automatic tool that extracts the relevant facts and problem solving methods from the two knowledge bases and based on this generates the code of a specific problem solver.

EXPECT's ontologies and factual knowledge includes concepts, instances, and the relations among them. Problem-solving knowledge is represented in a procedural-style language that is tightly integrated with the representations coded in Loom[17] - a state-of-the-art knowledge representation system based on description logic.

Given a knowledge base EXPECT automatically generates an Interdependency Model (IM). A basic type of IM is interdependency between ontologies and problem-solving

---

[16] EXPECT web site http://www.isi.edu/expect/link/publications.html. Publications on EXPECT include [49], [50] and [11].
[17] Loom project homepage: http://www.isi.edu/isd/LOOM/LOOM-HOME.html

knowledge. IM is then used to help the user in knowledge acquisition. It allows the user to detect inconsistencies, gaps and errors in the acquired knowledge as well as guides the user through the process of correcting these mistakes and adding missing data.

EXPECT is applied only when some initial knowledge base has been already acquired. But it allows to expand and correct the knowledge base and to update the problem-solving knowledge from an end-user.

The highlights of EXPECT are that it has an ability

a) to isolate the end user from internal representation;
b) to help the user to get started by providing default values and by asking questions;
c) to ask the user follow-up questions;
d) to make intelligent guesses and to use already existing relevant background knowledge.

The system as a whole reinforces the intuitive belief that, since the knowledge acquisition consists of capturing different kinds of data and different types of data require different approaches, tools and techniques, an integrated approach that combines several techniques should work best.

EXPECT is one of the few knowledge acquisition tools that allow a naïve end-user to easily enter new knowledge into the system guided by the knowledge-acquisition program. But even in this case the amount of effort and time required from the user are too large to be usable in electronic commerce where users are not committed to helping the vendor.

2.2.2.2 Knowledge Capture Tools

In the recent years a new trend has started to develop – analysis of user's behavior based on data mining methods applied to web server logs. These methods first clean the web records to keep only relevant information and then use different methodologies such as clustering, association rule and sequential pattern discovery[18] to generate new knowledge about the user population. The results of such processing are being used for adapting the presentation and content of web pages for more targeted advertisement campaigns and for customized recommender systems. Currently, many companies employ some form of this technology to ensure quality and effectiveness of their web presence. When these techniques are applied to electronic malls, they would allow quick dynamic updates to user profiles.

A web log usually contains the following information: user's server, user ID, date and time, referring page, target page, time spent on the page. Based on this data various types of information can be extracted. For example, number of visits, how many visitors came more than once, what are top entry pages, who are the most active visitors, where they come from, what search keywords visitors used to get to the page, and many other useful facts.

---

[18] Detailed description and analysis of these techniques and methods is beyond of the scope of this work.

Some approaches and tools have been suggested specifically for electronic commerce. They range from very general architectures (e.g. [51]) to specialized tools. The number of applications that use mining techniques to analyze web server logs is considerable: [52] mentions 30 currently available commercial applications in 1998, a current Upsala University web page (http://www.uu.se/Software/Analyzers/Access-analyzers.html) lists more than 80 web log analyzing tools for different platforms. Most of these tools are however difficult to use and to understand [52], other drawbacks include various assumptions used by these tools to reduce the amount of the information to be processed, a need to clean the web logs before processing, the fixed format of the report and limited amount of information delivered to the user.

Below, only two tools will be described: WebLogMiner [52] and WUM [53]. These tools are typical to the area in several respects:

- They use mining techniques in combination with other methods which are applied to prepare data for analysis and mining;
- They are designed for use by management and by web designers who make decisions on web site content and design, and need feedback about page popularity, users' interests, success of advertising campaigns, etc.

Attempts have been made to use mining techniques for other aspects such as user profiling. Work by Dai and Mobasher [54] and [55] shows how ontologies can be used together with data mining to allow the designer to capture behavioral patterns and user profiles. Most work in this area concentrates in the field of recommender systems that make suggestions to users based on the choices the user makes during the current session and the mapping of these onto previously discovered typical patterns of user behavior (for example, if the user selects a certain book the system can suggest other books bought by people who also selected the same book; other, more sophisticated applications would adapt a dynamic web page content based on anticipated user's preferences and interests). [55] presents a personalization framework utilizing domain knowledge (Figure 2.2).

[55] provides two useful insights that can be applied in the design of knowledge acquisition tools:

- A combination of mining techniques and ontologies;
- A framework for personalization using domain knowledge (Figure 2.2).

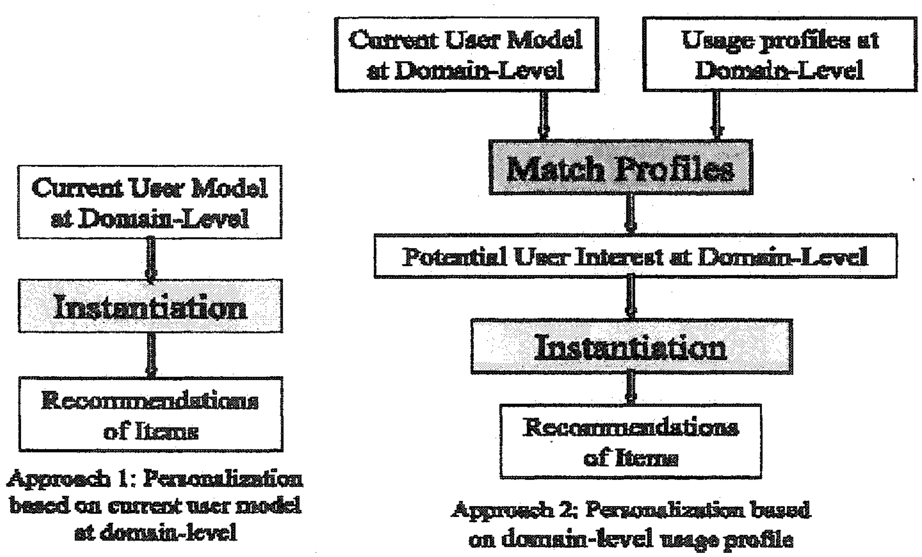

Figure 2.2. Personalization Framework utilizing Domain Knowledge (from [55])

However, the framework proposed in [55] does not solve some problems that user modeling for e-commerce may encounter: it works on the same principles as recommender systems where user's uniqueness is ignored and each customer is considered a part of a larger group, which is assumed to have the same tastes and interests.

Other approaches are limited to single applications – for example, [56] describes a personalized Web search based on data mining. Some recent research has been done in the area of web usage mining for individual user profiling [8], but its focus is more on the expert validation of the obtained results rather than on profile creation.

WebLogMiner[19]

Knowledge discovery tool for mining web server log files

The WebLogMiner tool processes web log data in four steps:

Step 1: First it filters data to remove "noise" (such as graphics downloads) to leave only relevant information. This information is stored in a database.

Step 2: Second, the system produces a "data cube". This is a multi-dimensional array where each dimension corresponds to some field with the entire range of values of its attributes. These attributes can be related to each other inside the dimension by a partial order. An n-dimensional data cube is a database. Each dimension of a cube represents an

[19] WebLogMiner is described in [52].

attribute and contains as many rows as there are different values to the attribute (data rows) plus one additional *sum* row. Possible dimensions include URL of the resource, time spent on page, agent that made the request, user, defined on a pre-built user hierarchy, etc.

Step 3: Third, OLAP (online analytical processing) operations are used on the data cube. These operations include drill-down, roll-up, slice, dice. They allow to view and analyze the data from different points of view.

Step 4: Forth, data mining techniques are applied to the data in the cube for predicting and classifying information and discovering correlations. For example, it is possible to identify network load and traffic patterns in relevance to time, including audience segmentation by the time and day of the week of browsing; to find out what are typical users browsing patterns and event sequences.

WebLogMiner shows the potential of data mining applications to web log mining. The tool is mostly designed for helping managers to improve the web site rather than to help the user to navigate it. Therefore, considerable attention was paid to the use of good visualization techniques (data cube) to allow humans to easily analyze data.

WUM[20]

WUM mines the log data on web traversals in a site and discovers navigation patterns in the form of graphs

[20] WUM is described in [53], a demo and a download are available at the project homepage http://wum.wiwi.hu-berlin.de/index.html

WUM (Web Usage Miner) is a tool that discovers navigation patterns in dynamic web sites where pages can be dynamically generated and multiple search criteria are allowed. This tool was created at Humboldt University in Berlin, Germany, by a team directed by Myra Spiliopoulou.

The main purpose of the tool is to provide insights into web site usability and possible directions for its improvement. To achieve this goal, WUM first finds typical patterns of visitors' navigation of the site and presents them in the form of a graph. These patterns are then compared to the site usage expected by the designer. Based on the results of such comparison, concrete suggestions for site improvement can be made.

The processing goes in several steps that correspond to different blocks in the architectural diagram of the WUM (Figure 2.3):

- Raw data available in the web server logs is pre-processed by WUM-prep: various heuristics are applied in order to clean the data from noise (e.g., graphics downloads), exclude robots, group sessions differentiating between multiple users using the same host. Then a "service-based" concept hierarchy of URLs is created in order to prepare the data for search pattern discovery.
- Aggregation service (WUM_agService) stores the prepared dataset in a predefined data structure, the *Aggregated Log*, merging sequences with the same prefix into the same tree structure. The subsequent data mining is applied to the set of Aggregated logs.

- Mining core (WUM-gseqm) applies data mining techniques and some heuristics to discover patterns conforming to instructions expressed in a mining language MINT[21].
- Results produced by the mining core are presented to the analyst in graphical form using SWING or the Java GUI of WUM-visualizer. The system can also execute batch processing without GUI when necessary.

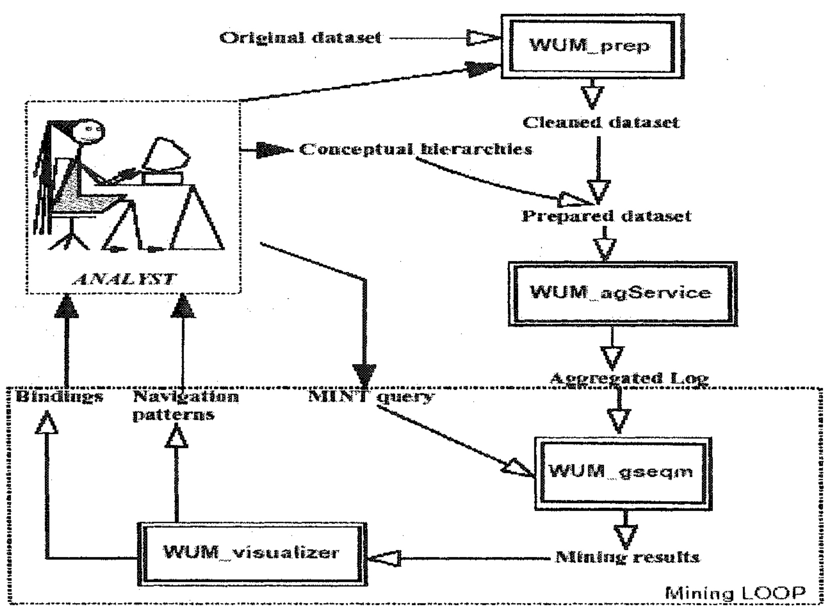

Figure 2.3. WUM architecture (from [53])

Just like other data mining tools, WUM is geared towards improving web-sites by finding generalized navigation or other usage patterns. Discovered patterns are then presented to a specialist who can make changes in the web site design in order to improve usability and attractiveness of the site. WUM is flexible because it does not have a predefined report format as most commercial products do. It allows to formulate customized queries

[21] MINT syntax is briefly explained at http://wum.wiwi.hu-berlin.de/wumMint.html.

in an easy to learn query language MINT which makes this tool a good choice as the data mining component for a knowledge acquisition system for dynamic user profiling. However, considerable modifications will be necessary to adapt this tool to the task of individualized profiling instead of discovery of navigation patterns for groups of users.

2.3 Conclusions

The brief summary of the tools reviewed so far is presented in tabular format in Table 2.1 (p.50). The table also includes some information about tools that are not available to general public and could not be described in detail.

Based on the analysis of tools and techniques presented in this chapter the following points are noted:

1. Internet questionnaires can be used to collect information in easy-to-process electronic form. A clever and well-designed small questionnaire will be acceptable to users and can supply pertinent information for building a user model.
2. Focus groups can be used to get better knowledge about consumers. During focus groups sophisticated KA tools such as WebGrid can be applied.
3. A variation of purchase intercept technique can be applied in case of Internet shopping. Instead of talking to the consumer his behavior can be "observed" and recorded by a software agent in the form of a web log or transaction/purchase

history. A record of user's actions can serve as input for building a user model using data mining techniques.

4. Other traditional techniques such as interviews with consumers can also be used to provide a knowledge engineer with necessary input but they are usually expensive and require human intermediaries, such as trained interviewers, to collect the data. Therefore, these methods are not a good choice for electronic commerce where they can be successfully replaced by more efficient, less time-consuming and less expensive methods based on data mining and machine learning.
5. Many successful KA tools such as PC PACK or EXPECT combine multiple techniques for dealing with different kinds of data. This modular approach is particularly important in the case of user modeling for electronic commerce where information can range from text to database entries to statistical data. Recent work in the area of data and web mining gives additional dimension to this modular approach.

The existing knowledge acquisition tools can be subdivided into two broad categories – classical and modern tools. There is no strict boundary between the two classes since there exist modern versions of classical tools that combine some properties of former and later.

Classical tools such as TEIRESIAS, MORE, OPAL and many others are characterized by the following traits:

1. They were designed for use by professional knowledge engineers and specially trained domain experts;
2. Their main purpose is to elicit a complex structure of an existing body of knowledge;
3. Their interface is purely functional, often command line;
4. They provide little flexibility and are system- or domain- centered.

The **new generation of classical tools** (PROTÉGÉ, PC PACK, etc.) have the following characteristics:

1. They are still geared towards knowledge engineers and domain experts with advanced knowledge of computers;
2. Their goal is again to elicit the existing domain knowledge;

At the same time now

3. They have user friendly intuitive GUI;
4. They are versatile and can be applied to different domains and used with large variety of systems written in different languages.

Modern KA tools (e.g., EXPECT, WebGrid) are aimed at different audience and incorporate changes to interface stemming from that.

1. They are designed for end user with minimal or no special training and basic level of computer literacy;
2. They provide intuitive help/guidance through the entire KA process;
3. They possess attractive and easy-to-use GUI.

Another class of modern KA tools has appeared in recent years – **knowledge capture tools** that have limited interaction with user and are designed for web log mining and large consumer database querying. These tools:

1. require interpreting of results by a professional;
2. are designed for web developers and the management of web-based businesses;
3. are able to capture the dynamic knowledge, such as changes in the user behavior;
4. usually possess attractive GUI with considerable amount of graphics.

In Figure 2.4, we present the KA tools and the historical trend in their development.

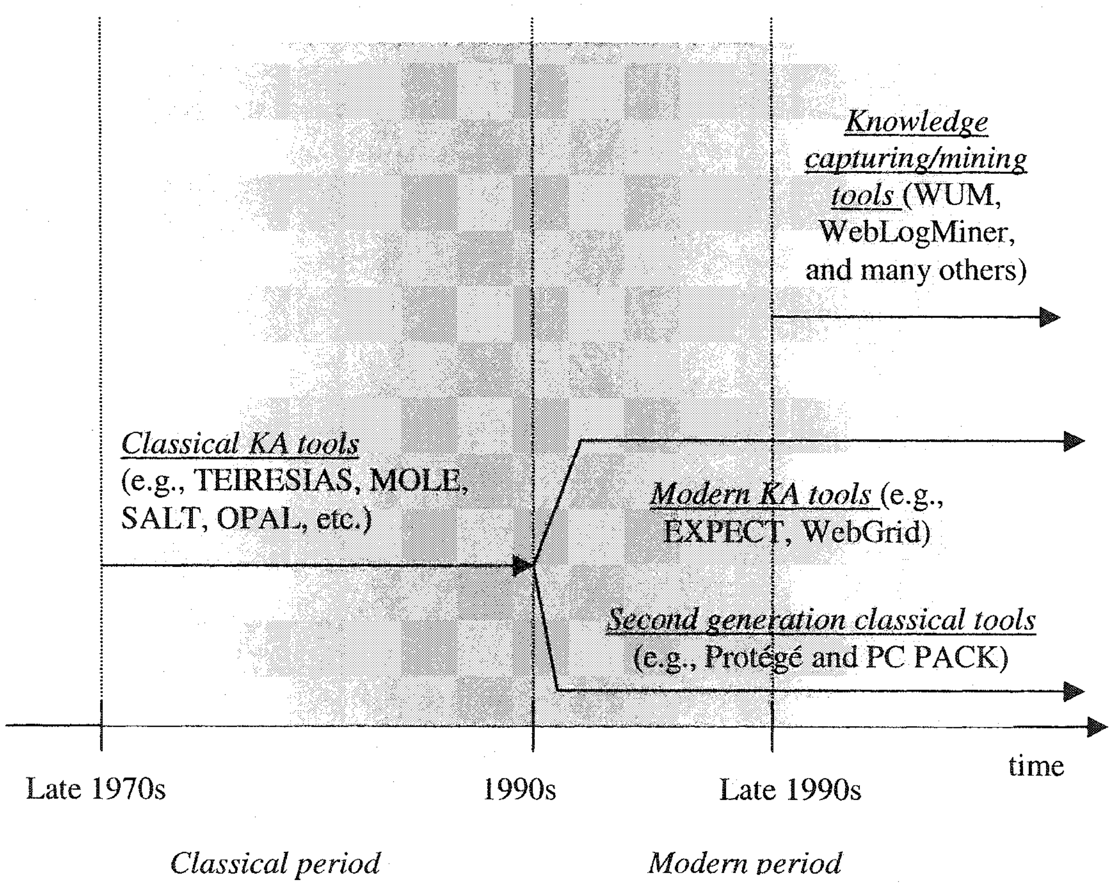

Figure 2.4. Types of KA tools in historical perspective

The analysis of existing tools, methods and techniques showed us that there are two most important general considerations related to the knowledge acquisition in electronic commerce:

a) Most Internet shoppers do not want to invest much time or effort into the creation of their profile. It is perceived that it is the seller who takes more advantage from maintaining user profiles than the customer. Therefore knowledge acquisition should rely as little as possible on direct end-user input and utilization of the user model should be pleasant and natural for the customer. This implies that complex tools requiring any extra effort from the end-user will have no success. Tools that minimize the customer's direct involvement such as web log mining software are a better choice.

b) Creation and maintenance of the user profile in e-commerce would require several steps from acquiring and encoding general domain related information to fine-tuning the individual profile. Therefore, there is no unique approach that would solve all the problems related to user modeling. Tools such as EXPECT or PC PACK, which show a combination of various approaches and techniques, would work better.

These considerations will be taken into account in designing the Knowledge Acquisition Tool for user modeling in electronic commerce presented in Chapter 4.

Table 2.1. Some Knowledge Acquisition tools

Tool	Author	Year	Link	Main technique	Note
PC PACK	Epistemics Ltd and Prof. Nigel Shadbolt (Southampton U.)	2000	http://www.epistemics.co.uk/products/pcpack/	Repertory grid, cart-sort, KADS model	commercial
EXPECT	U. of Southern California. ISI. Yolanda Gil, and others.	2000 (prev. version – 1998)	http://www.isi.edu/expect/expect-homepage.html	Interdependency Model, KA scripts	Mainly for KB modification and problem-solving knowledge acquisition
PROTÉGÉ	Stanford Medical Informatics. Stanford University.	Last version - 2000, first version – 1989	http://www-smi.stanford.edu/projects/protege/protege-2000/index.html	Skeletal plan refinement technique.. Last version uses Open OKBC Model.	free (needs registration)
WebGrid	Brian R. Gaines and Mildred L. G. Shaw. U. of Calgary	1996	http://ksi.cpsc.ucalgary.ca/articles/WN96/WN96WG/WN96WG.html	repertory grid	
START	Ai Lab at MIT Boris Katz and others.	1997	http://www.ai.mit.edu/people/boris/webaccess/	Annotation	Only querying part available
KAT	Dynamic Research Corporation	1999	http://orl03.drc.com/kat	no details available	Commercial, no downloads available
KCT	IEEE Computer society	1997	http://www.computer.org/tkde/tk1997/k0209abs.htm	best-match strategies – vector space and probabilistic ranking principle	
WUM	Humboldt U., Germany. School of business and economics. Institute of Information Systems	Last version – 2000	http://wum.wiwi.hu-berlin.de/index.html	Sequence miner. Log preparation, mining and visualization.	free
WebLog Miner	U. of Illinois at Urbana-Campaign	1998	http://www-faculty.cs.uiuc.edu/~hanj/pdf/weblog98.pdf	OLAP and OLAM	Research tool
MUSCRAT	Department of Computer Science, U. of Aberdeen	1996-1997	http://www.csd.abdn.ac.uk/~swhite/tr9605/TR9605.html		Centered on machine learning

Chapter 3

AGENT-SUPPORTED DYNAMIC PERSONALIZATION: TWO CORE CONCEPTS

This thesis is based on two core concepts, namely, software-agent's support in knowledge acquisition and dynamic personalization. In this chapter, these two core concepts are briefly discussed with reference to the context of the thesis (e-commerce domain). Personalization efforts require a clear understanding of the user needs and the goals of the optimization involved in personalization. For the purposes of the former, generally a model of the user is employed. The published literature on user models is very rich (for example, there have been already nine conferences dedicated to user modeling [http://www.um.org/], an international Journal on User Modeling and User- Adapted Interaction (UMUAI) are being published since 1991, etc.). Some aspects of User Modeling relevant to this thesis will be discussed in more detail in Chapter 4.

Most user models that are currently employed in electronic commerce have very little flexibility. In part this is due to the fact that they reflect trends in more or less large consumer groups that are relatively stable, rather than preferences of individual customers that may change rapidly. In reality, however, "the Me's of the world are unique, ever-evolving, mostly unpredictable" [57]. There is a need to adapt user models to these ever changing and unique "Me's".

Since our goal is to cater to individual needs of repeat customers the user model should be made as flexible as possible. This approach, however, can lead to creating unmanageable workload for knowledge engineers who maintain the system and they will have to keep up with all the necessary changes in dozens and even hundreds of user-profiles depending on the web-site's popularity. The proposed knowledge acquisition tool architecture is designed to address this problem by delegating a certain portion of the load to "software agents" (also called 'agents').

3.1 Dynamic User Profiling

Dynamic personalization or user profiling can be understood as "matching customers and content in real time" [58]. But this definition is very general and as further analysis shows ,it gives rise to various interpretations. The study of several systems demonstrates that the term d*ynamic personalization* is used in several ways. Below, we describe three cases that are not mutually exclusive.

Case 1: In this case, dynamic personalization means that the content and/or parts of the user interface are dynamically generated according to a (static) user profile [e.g., http://www.interactivesites.com/pdfs/DARTmail_dynamicpers.pdf]. Such systems are mostly commercial products for CRM (customer relationship management) activities, such as email campaigns, targeted advertisement, etc. Most commercial products claiming to provide dynamic personalization belong to this type.

Case 2: In the second case, there may be some a priori stored information about the user, but it is augmented by the dynamically varying "current context" of the user [e.g., http://www.indasea.com/dyper.htm]. The conceptual notion of the "current context" can be characterized and suitably represented for computational purposes. The time and space varying parameters of this representation are obtained (or sensed) by suitable I/O devices (e.g. [59] describes one of the projects in MIT Media Lab that is part of the research in context-aware computing). Research in the area of mobile agents, wireless communication, PDA (portable digital assistants) and collaborative working have relevance to this type. At this stage, majority of systems using this understanding of dynamic personalization are built as part of research projects in universities or research laboratories.

Case 3: In the third case, the user profile or user's interest itself is dynamic. The changes over time are captured through a chosen set of factors pertinent to the user. This approach is described in [60] and it is gaining popularity, although it is still at the research stage (e.g., IBM Dynamic Personalization research project[22]). It can be combined with data about current user contexts when applicable. For example, a system of this kind has been suggested by Koichi Goto and Yahiko Kambayashi [61] for mobile passenger support systems in the use of public transportation.

In most cases, described in the e-commerce literature, dynamic personalization serves as a means for better targeting of the marketing campaigns. It is applied for refining the profile of an "ideal" customer to whom the advertisement and other promotional

[22] http://www.trl.ibm.com/projects/mrm/dp/index_e.htm

materials should be addressed, based on the mining logs of customer's acceptance/rejection of offers. Applying dynamic personalization to maintenance of individual profiles of repeat customers in a store is a more challenging task, that has not been given the attention it deserves, despite the popularity of the concept of one-to-one marketing. Most likely this is due to the complexity of the task.

In all its usages, the word *dynamic* implies changes and we need to understand the sources and the nature of such changes. If we consider a virtual mall, there are multiple sources of changes:

1) Changes in a virtual store
   a) Changes in stock;
   b) Changes in goods and services sold;
   c) Price changes;
   d) Changes in the amount of information to be presented in a transaction or a dialog;
   e) Changes in store ownership, e.g. the company can merge with or be purchased by another store (e.g., chapters.indigo.ca; ToysRUs becoming part of amazon.com, etc.)
   f) changes in the business model, business practices, and policies.

The impact of these changes on an individual customer or customer's perception of these changes may vary and is difficult to predict (e.g., they can affect the brand loyalty in both negative and positive ways). Some modifications (e.g., in pricing or in web-site organization) can bring in considerable amounts of "noise" – for instance, customers may get confused when the familiar interface is no longer available and an atypical browsing behavior can be observed until the users get used to the new interface.

2) Changes in the customer's behavior:
   a) Changes in interest and preferences of brand names due to dissatisfaction or change of lifestyle, family situation, etc.;
   b) Short- or long-term deviations from typical patterns being influenced by different people or circumstances (e.g., trying-out recipes from a new cook book);
   c) Changes in role played in making purchase decisions, e.g., decision maker or influencer (for example, a mother buying groceries for the entire family is likely to make her own decisions about the choices unless she is shopping for her child's favorite food – in the last case the child is the influencer and purchasing decisions made under such influence may significantly vary from a "standard" pattern).

The changes to user behavior can be long term or short term; may be occasional or temporary deviations from a usual behavior (e.g., shopping for someone else or shopping for a party), others may be persistent and have impact on the user profile (e.g., special diet due to long-term illness; birth of a child). In the knowledge capturing process, it is very important to be able to differentiate between these two types and include only long-term tendencies into the persistent user model.

The changes in the user's context create a significant degree of uncertainty when it comes to dynamic personalization. The challenges are: how to distinguish between long-term changes and occasional deviations from a routine; how to handle conflicts between the effects of short-term modifications with that due to persistent information? Some of these problems will be considered in the design of the system presented in Chapter 4.

3.2 Agent Support

Creating and especially maintaining a dynamic user profile is a time consuming complex task that puts considerable load on the end-user and the knowledge engineer. To reduce the amount of effort required, software agents should be employed at different stages of the process.

Software agents have been around for about two decades and gained a considerable popularity in different areas of human-computer interaction[23]. There is no universally accepted definition of the term "agent" or " intelligent agent" ([62], multiple references can be found in [63], [64] and [65]). Software agents possess certain qualities that are common to this technology: they are autonomous, proactive, knowledge based; they assist the user and manifest social behavior[24]. E-commerce has been among the domains where agents play the most important role ranging from "search-bots" to "auction-bots".

[23] Information about various aspects of agents development and use can be found at http://agents.umbc.edu/ , useful links are provided by MIT Media Lab Software Agents Group at http://agents.media.mit.edu/resources/

[24] Other attributes often considered pertinent for agents include reactivity, temporal continuity, personality, mobility [66], [67].

Agents are popular both with sellers for whom they save time and effort, and assist in providing good quality service to customers, and with customers who can reduce time and effort necessary for dealing with information overload as well as save money by finding best deals using price-comparison agents or auction bots. Comparative shopping, customer relationship management, and price negotiations are the three best known areas in e-commerce where application of agents has been explored both by researchers (e.g., *Kasbah* created at MIT for consumer-to-consumer e-commerce, recommender system *Firefly* based on collaborative filtering, etc.) and by commercial organizations (e.g., the first shopping agent for price-comparison, BargainFinder, by Andersen Consulting; bot-based products for CRM by ArtificialLife, Inc.; and many others).

Most of the applications of the agent technology to e-commerce have been centered in assisting the sellers in the web-service maintenance and monitoring, or the buyers in the process of searching, auctioning and price negotiations [68]. User modeling in the context of agent technology has been mostly interpreted in terms of building user profiles as part of agent's knowledge base. However, in this thesis we are concerned with knowledge acquisition for personalization of the interaction between a virtual store and a customer rather than simply for transaction support in e-commerce (even though it is true that these two aspects influence each other). Two distinct approaches to KA can be identified with regard to the knowledge about the individual user[25]:

a) "Knowledge capturing" by monitoring the dialog between the user and the system and intercepting the relevant data, for data mining operations that create new

[25] These approaches correspond to two kinds of modern knowledge acquisition tools described in the previous chapter: knowledge elicitation tools and knowledge capture tools.

knowledge about the user. In this approach the user does not contribute explicitly to the KA process. The captured data is used in 'data mining' and the mined information is reviewed by knowledge engineers or other specialists before it is used. A variant of this approach would be to apply the traditional AI-based machine learning techniques with no human intervention.

b) "Knowledge elicitation" by a KA tool. This is a stand-alone phase in which the user contributes explicitly and directly to the building of a knowledge base about him/herself.

It is possible to use agent technology in both these approaches to reduce the burden on the human participants. Let us consider the following aspects of agent assistance that are labeled Assist-1, Assist-2, etc. These aspects will be illustrated by example of a wellness products store that will be our running example throughout the rest of the thesis.

The wellness store is a specialized boutique that sells health products such as organic food, vitamins and minerals, food supplements, etc. This kind of store has been growing in popularity in both brick-and-mortar and click-and-mortar versions. It has a specific customer base: people who shop there are health-conscious individuals who are willing to spend more for perceived benefits of ahealthy life-style. At the same time their concerns and interests are very different and there exists a large variety of products created to cater to these diverse needs. Selecting the right kind of product is a complex task and a system that has information about user's needs and constraints can considerably facilitate that task. In the following, we will show how agents can assist in the gathering and using of the user information in this case. The system under consideration has a product database, and a user database containing individual user profiles and a product taxonomy.

Assist-1: Agent assists in the initial profiling stage. Setting the initial profile will influence the further updates and usage of it; therefore special care should be taken at this stage to assist the user in completing the questionnaire as well as to select the most suitable stereotypes for filling up the missing values. At this stage the agent will use stored knowledge about typical users' characteristics to set missing values in the user model or to explain to the user how the parameter values might affect the system performance. The goal is to make a new user registering into the system to be comfortable and informed about the questions asked in the building of the 'initial user profile' that is individualized.

For example, a first-time user who is registering in a wellness store can provide the system with some personal information, such as health concerns, age, etc. At this stage the user input is the main source of information supplied to the agent, At the same time this source has a priority over other sources – parameter values set based on this input should have highest certainty and should overwrite any system's inferences. To ensure reliability of this information, the agent will help the user

(a) to understand the implications of the supplied information,

(b) to help the user to make correct choices, and

(c) to alleviate privacy concerns related to the use of information given to the system.

For example, if the user is answering a question about children, the agent can explain why the system needs to know if the shopper has young children (because they usually have special needs and their choices influence the choices of their parents).

At the same time we cannot ask the user to fill a really long questionnaire and also the user might not want to give all the information during the first visit to a new store for lack of time or because of privacy concerns. The information that the user supplies can go directly in the user profile, but many parameter values will remain empty because of lack of direct user input. To fill the missing slots of the user profile the agent can apply a set of rules based on stereotypes. For instance, if the customer has told the agent that she is a 30 year old woman the agent can then activate rules that encompass the stereotypical profile of such users, e.g., statistics [69] show that 60 % of heavy organic buyers are female, most of them in the 25-34 age group therefore the agent can set attribute *interest in organic* to *yes* with certainty corresponding to the probability based on these statistics. In this fashion, the agent will produce a working version of a user profile from the very beginning of the customer's interaction with the store, without asking the user for too much data.

Assist-2: The agent can process the information supplied by web log mining module and make changes to the user profile when necessary. The agent will autonomously complete the task of collecting information from these sources and decide if there is a need for modification. The agent can also assist the user in clarifying the changes before committing to them. The goal is to provide the ultimate control to the user in deciding how his/her interests are represented in the system through appropriate user models.

For example, suppose the user has told the system at profile initialization that she is vegetarian but she is now buying meat: as we have noted in Assist-1 the information from these two sources has different weight – values directly based on data supplied by the user get highest certainty, values inferred by the system have lower certainty and should not overwrite user-supplied data. The agent detects the inconsistency between initial constraints and results of observations that seem to violate these constraints and asks the user if any changes to the profile should be made. The goal of the agent is to create a profile that will maximize the satisfaction of the user therefore the agent wants to relax the existing constraints in order to give the user immediate access to necessary information. Before making these changes, the agent explains that by making the adjustment the user may get access to the list of meat products at the login time without searching for them because this information is usually not displayed to vegetarians.

Assist-3: The agent can assist in tracking the effects of the changes made to the user profile and report back to the user at a suitable point, so that the user may revoke the changes (or modify them). The goal is to enhance the user's confidence, by knowing whether the changes made in the model are yielding better performance or not. Since the ultimate goal is to converge to maximal performance of the system and maximum customer satisfaction, the agent can help in the tedious process of adjusting the profile and monitoring the results of these adjustments for the overall trend as well as for particular constraints.

For example, after a change to the profile has been made – e.g., preferred brand for yogurts has been changed from Danone to Yoplait – the products by the former producer have been moved from first browsing screen to the second. The user might not realize that the new constraint was too rigorous and might be frustrated by not finding familiar products at the usual place. The agent should then volunteer information about this change and let the user know about its effects so he/she can either choose to keep the constraints as they are and make extra effort to browse more categories or relax the constraint and get easier access to the desired information. Using such a feedback loop, the agent will achieve faster convergence of the profile to the desired performance level.

Assist-4: The agent can assist the user in trust related issues, for example, it could proactively warn about certain vendors, or products. This case is similar to Assist-3 but here the agent deals with external knowledge about the world. The agent's task is two-fold – it finds information relevant for the user (e.g., sources of in-depth knowledge, facts relevant for user's decision-making) in the "outside world" and at the same time it filters the requests coming for user information (e.g., by filtering out queries that violate user's privacy). The knowledge base about the user (KB-U) can be kept by the user agent locally. Other agents or software can ask questions to this user-agent using a standard interface such as KQML [http://www.cs.umbc.edu/kqml/] or FIPA [http://www.fipa.org/]. The user agent may answer such queries on its own, when permitted by the user, or after getting the approval of the user.

In all four cases mentioned above, the knowledge engineer's task will be limited to setting up the agents, testing and fine-tuning their behavior instead of making decisions in every single case. Building such an agent that assists the knowledge engineer, would require a very clear understanding of the knowledge acquisition process. When such agents are in operation, they are expected to reduce the amount of time and effort required from the end-user in making the profile to reflect the dynamic changes in his/her lifestyle or circumstances. Instead of manually resetting the profile and answering questions every time when significant changes occur, the agent-supported approach should reduce the human effort.

Finding a perfect balance between the necessity to provide transparency and sense of control to the user on one side, and the goal of minimizing user's time and annoyance, on the other side is a major issue when it comes to user agents. One of the possibilities to solve this problem, particularly when it is due to the mode of interaction, is to provide the user with the choice of different modes of interaction with the agent from which he/she can select one that suits him/her best. We suggest providing three modes of user-agent interaction:

- maximal user's involvement when the agent should report to the user and get his/her approval in case of any decision (e.g., changes made to user profile). It will ensure maximum transparency and control but will require a significant investment of time and effort by the user;
- intermediate user's involvement The agent initiates the dialog only when in doubt (e.g., observed behavior strongly contradicts previously set values). It is a challenge to design an algorithm to detect when an agent is in doubt.
- minimal user's involvement when the agent does not initiate any interaction with the user and the user him/her-self asks the agent in case he/she wants to have system behavior explained or changed. In this mode the actions of the system are less transparent but by the same token the user saves time and effort needed for constantly monitoring agent's actions.

The choice of the mode of interaction should be a part of the user profile. The value for this characteristic can be set in two ways:

1. Manually by the user (e.g., the user can be provided with a menu with the three choices that he/she can access any time);
2. Automatically by the processing agent. In this case the starting value would be *maximal user involvement* that corresponds to high degree of interaction because

a novice user not familiar with our system might need considerable support from the agent during the registration process. When the profile is initialized, the certainty for this value will decrease slightly; it will decrease considerably if the user chooses not to pay attention to the agent's advice (e.g., closes the dialog box). If the user, on the contrary, seems to need more help (e.g., accesses the help very often) the value can be increased. There will be two thresholds that separate three modes of interaction – T1 will serve as a boundary between high and intermediate involvement, T2 – between intermediate and minimal. For example, when the value drops below T1 (e.g., 0.5) the interaction mode is switched to *intermediate*, if the value becomes even smaller (e.g., close to 0) the mode is set to *minimal* interaction. The exact values for thresholds should be validated empirically and may differ depending on the kind of characteristics of the customer base of a particular shop/boutique (e.g.: customers in a virtual computer store might rely more on the agent than visitors of a cosmetic store; people who shop for computer games are more likely to prefer *maximal user involvement* interaction while grocery shoppers don't want to spend extra time on communicating with the agent when doing routine shopping).

In the process of knowledge acquisition for a user profile, the agent(s) can interact with the user in several ways:

- Using context-sensitive help when the user clicks on a menu choice or on a button to view pertinent information;

- using a form-based dialogue between the computer and the user when the computer (agent) asks pre-set questions and user has to choose one of the possible answers;
- using a mixed-initiative dialogue between the user and the computer. There can be two modes of this dialogue: machine-driven, when the system asks for clarification, or user-driven, when the user gives information to the system or asks the system for information without being prompted by the computer.

Context-sensitive help is a good choice for a default option that should be available any time and its major usage will be in the situation when the user chooses the *minimal involvement* mode to provide the customer with required information if needed. Form-based dialogue, especially in the form of yes/no questions, is relatively easy to implement; it can be employed in the *intermediate involvement* mode to save time and effort on both human and computer sides.

The mixed initiative dialogue is the most general approach that can be applied in all three modes mentioned above. As stated by [70] "the addition of mixed-initiative techniques introduces much greater flexibility into AI systems, in terms of degree of user involvement in the automated reasoning process." This technique has got considerable attention in different areas of computer science from requirement engineering (e.g., [71]) to interface design (e.g., [72], [73]) to knowledge base development (e.g. [74]). It is often used in conjunction with agent technology (e.g. [75], works by Fleming and Cohen). There exists a large literature analyzing different aspects and applications of mixed-initiative dialogue (comprehensive overview can be found in [76]). We believe that this technique is the most suitable when it comes to providing flexibility and finding a tradeoff between the amount of effort the user is ready to put into profile creation and maintenance and his/her desire to stay in control of profile updates and their implications.

3.3 Conclusion

Dynamic personalization involves two major activities: updating the user profile based on observations and feedback, and personalizing web site content and presentation based on the most up-to-date user model. Both these activities are labor-intensive and require fast and reliable processing in which human participants will acquire machine's assistance. Therefore, the use of agent technology is a good choice for relieving considerable amount of work, if not all, from humans (customers and knowledge engineers) and for providing proactive assistance to users of both kinds. The user will have a choice of three modes of interaction with the agent ranging from *maximal* to *intermediate* to *minimal involvement* depending on user's explicit choice as well as on automatic increasing/decreasing parameter values set by the agent based on user behavior. Use of mixed-initiative dialogue will provide necessary flexibility in the process of user-agent interaction. The communication between user and agent(s) can also be implemented as context-sensitive help and as frame-based dialogue. The architecture presented in Chapter 4 will take these considerations into account.

Chapter 4

THE PROPOSED KNOWLEDGE ACQUISITION TOOL FOR DYNAMIC USER PROFILING IN E-COMMERCE

4.1 The User Model

4.1.1 What are User Needs Today?

User Modeling (sometimes also called User Profiling)[26] is an important tool that can be used to predict user's behavior, his/her actions, reactions, preferences, needs, etc. User Models are extensively used for customization of interactions with end-users, including agent-based systems. In these adaptive systems, the user model represents the system's assimilation of the interaction and contains information about the user and the current context that can increase the system's ability to exhibit *pragmatically correct* behavior and, more generally, to engage in effective communication [77], [78].

There exist several classifications of user models (see [79] for a detailed overview). One of the most popular classifications was suggested by Elaine Rich [80]. She proposed to use three criteria for classifying user models:

- Canonical versus individual. This parameter differentiates between a single "generalized" model of a typical user and a collection of models for individual users. User models based on stereotypes – sets of characteristics typical for a

[26] These two terms used interchangeably by most authors.

more or less large groups of users – can be placed in one or another point of the continuum between the two extremes depending on the model granularity;

- Explicit versus implicit. The first kind of user models are created with considerable involvement of end-users who create their own profiles (e.g., myYahoo!), the second approach entrusts the task of profile creation and update to the system that can use various sources of information such as records of user actions to infer knowledge about user preferences.
- Short-term versus long-term. The last kind of user models deals with persistent information that is expected to remain stable over time while short-term models try to capture dynamic properties in their evolution. Intermediate kind of user models are also possible for situations where some information about the user is stable (e.g., profession, color preferences) while other characteristics are dynamic (e.g., family situation; brand preferences).

The popularity of the personalization is a logical result of the mass customization [2]. [3] insist that "customers <...> do not want more choices. They want exactly what they want - when, where, and how they want it - and technology now makes it possible for companies to give it to them." The result of mass customization and one-to-one marketing is a "learning relationship - an ongoing connection that becomes smarter when two <producer and customer> interact with each other, collaborating to meet customer's needs over time" [3, p.53]. If such a relationship is built the company "will retain more customers, especially the most valuable ones: frequent purchasers." [3, p. 60].

This idea is fully supported by the recent trend towards customer retention in the on-line retail business. As shown in [8] and [10] we are currently witnessing a new shift in personalization towards catering more to the needs of repeat customers as well instead of putting all the effort into attracting new customers or make one-time shoppers spend more. Encouraging customers to return to the e-store creates new challenges for user modeling.

Traditionally, it has been considered that in electronic commerce it was more important to build canonical long-term explicit models because the user profiling was done in the interest of businesses (in most cases – large, with thousands of customers) rather than buyers. Long-term canonical explicit models were preferred by the on-line retailers because

- They are easier to create based on data that is traditionally collected by these companies for their marketing purposes;
- They are easy to maintain since they are built under the assumption that a typical customer does not change much;
- They require fewer resources – when it comes to thousands of users creating thousands of profiles, storing, retrieving and maintaining them becomes prohibitively slow and expensive.

In recent years the situation in e-commerce started to change – more and more small businesses are going on-line because it has become easier to create a virtual store: there are many options for web-hosting; some of them quite inexpensive; many specialized companies offer full implementation of a store, and there are even complete on-line stores for free download that require relatively small customization (e.g., phpshop); people have become more familiar with Internet technology and both retailers and customers are ready to use it more and more extensively. In 1999, more than a quarter of small businesses were selling their products on the Internet [81]. According to IDC[27], annual growth in the number of small businesses that conduct e-commerce in 1998-2000 has been 34.6 % [82]. This trend means that now hundreds of small businesses that have relatively narrow customer base are gone on-line and a new approach to user modeling in e-commerce that would take into account the particularities of these small businesses is needed.

At the same time the user has also changed: the Internet technology has become a commodity and the attitude of the consumer has changed accordingly – people no longer want to spend time on building their own profiles nor do they want to skim through lots of irrelevant information, that has been proposed to them just because the hypothetical typical user might have been interested in it. Customers now expect the computer to be more intelligent and to do more to help them cope with information overload and lack of time.

These trends require a matching approach to user modeling for e-commerce that will be more oriented towards individual users, will built mostly on "looking over user's

[27] International Data Conrporation. http://www.idc.com/

shoulder" (Elaine Rich) and will be dynamic. This model will be positioned in the three-dimensional space proposed by Elaine Rich as shown in Figure 4.1:

- Along the vertical axis, it will be "individual" with some elements of the "canonical model" (in case individual information is missing, details from the canonical model will be used);
- Along the horizontal axis, it will combine long-term and short-term models, depending on the attribute and profile dynamics. The measure of time is relative. Heuristic algorithms will be used to determine when to apply the rules of the long-term or short-term models;
- Along the third axis, it will be mostly implicit (based on observations), some elements of explicit model will be present for the results of initial user registration and the dialog between agent and user.

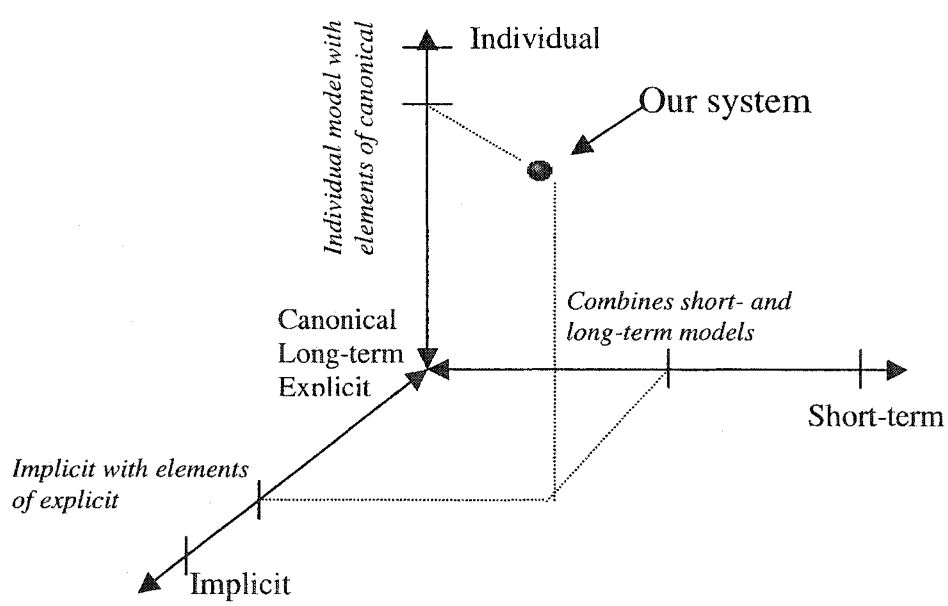

Figure 4.1. Positioning of the PUMe in the 3D classification space of [80]

We recall that the main problem faced in the knowledge acquisition for user profiling is finding a balance between cost (time and effort a user invests into creating the user profile) and benefit (perceived or actual) to the user. Since in the long run it is the seller who benefits the most from making his web site more attractive to users we believe that the cost also should be absorbed by the seller as much as possible. On the other hand the user should be given a full sense of control over the interaction with the web-site including a way of turning the personalization off completely, when it is undesired. This philosophy reflected in the proposed architecture.

4.1.2 Content and Structure of the User Model

This part of the work on user model builds upon research performed earlier by other members of the CITR project group, especially on the PIE (Preference Indication by Example) user model proposed by Rony Abi-Aad in [79].

The concept of Preference Indication by Example is based on idea suggested by Elaine Rich in [80]. She proposed to represent user's "personality characteristics" as a list of facets where each facet has a value and a rating associated with it. Similarly, each PIE stores information about attribute value, its relevance to the user and the certainty of this information. The PIEs form a natural 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. This hierarchy is useful for building and refining the user model, it can help propagate information up-and-down the DAG and thus vary the relative notions of generic and specific. The information (values, certainties) can be inherited from general categories to their sub-categories or can propagate up in the hierarchy. The information stored in specific categories has higher weight and can override the data inherited from general categories in the case of conflict between the too. The use of a hierarchical organization of the taxonomy is also useful because user's preferences can vary in precision. For example, the user can be interested in a specific product (e.g., in plain Cheerios) or in an entire category of products (e.g., cereals). The taxonomy will allow both to be reflected in the user profile and in case of both queries the user will be able to get personalized service (e.g., he/she will be presented with a list of organic cereals if that's his/her preferred kind of products).

The PIE model as presented in [79] contains three main types of information about the user:

- The categories and subcategories of products the user is interested in. This knowledge can help filter information, or personalize browsing on a general level. It is referred to as PIE (Preference Indication by Example).
- The features of those products or categories. This knowledge can help compare items and predict user's interests on a more specific level.
- 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 centered additional information is also domain dependent (e.g. expert in cars,

professional in ski); therefore it is associated to products. It is called "user additional information".

This model has been used as the basis for a slightly modified version employed in this thesis. The suggested modified user model includes:

- A hierarchy of the products as a DAG proposed in [79] that will allow to navigate the store faster and to easily apply knowledge about a certain category to a new item that can be added to it (e.g., if we know that the user prefers bran cereal we can automatically infer that a new brand of such cereal that just appeared on the market may be of interest to the user);
- At each leaf a detailed information about a product is stored, including all features that are potentially relevant for the store client;
- The user profile is stored as a frame that includes a set of attributes together with their values and a degree of certainty we attribute to the assigned value ranging from 0 to 1 inclusively where higher numbers represent increased certainty. Instances of such frames are attached to the top node of the taxonomy corresponding to the store. We decided to store the information about each user's preferences at the highest possible nodes to avoid high cost in terms of space and time due to storing duplicate data (e.g., if the customer consistently buys juices rich in vitamin C we need to store this information only once for all juices rather than for each particular brand and flavor) and 0-valued features.

The product descriptions are represented as sets of attribute-value pairs; for example: 'brand : Danone', 'calories : 62, fat : 0.5%, flavor : plain, organic : no'. In the user model corresponding features can have same values (e.g., "Preferred brand: Danone"), or quantitative values including comparisons (e.g., 'fat: <1%') as well as qualitative and Boolean values ('calories: low'; 'flavor: not peach'; 'organic: no'). The list of features included in the user profile is shop-specific and depends on the type of products sold.

Frames at leaf nodes corresponding to specific products contain detailed information about the product. At the upper levels of the hierarchy, the values of the features become more abstract and generalized (e.g., if at the leaf level we have 'price : 1.49', at the next level we will have 'price : low'). Eventually, at the root of the taxonomy that corresponds to the entire store we arrive at the generalized list of user's preferences - the complete user model for the customer (Figure 4.2 p.77).

The user model adopted in this thesis does not use "user additional information" as was suggested in [79] for two reasons: first, storing irrelevant additional information is expensive and not justified, while information important for a particular store/boutique (e.g., knowledge that a customer of a grocery store is vegetarian) is an integral part of the user profile rather than "additional" information; second, since we are using mostly implicit user profiling only data pertinent to the particular store will be collected and therefore there will be no additional information.

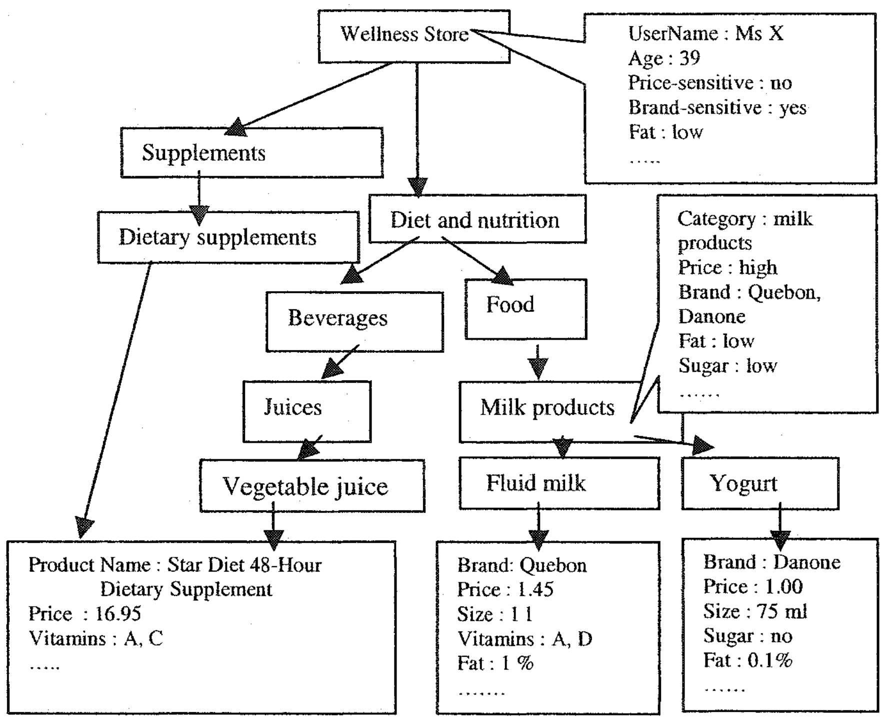

Figure 4.2. A fragment of a user model with frames representing product features and user preferences

In order to create a fully functioning user model we need to solve several problems, such as:

- How to organize a product taxonomy?
- What information about products to represent?
- What information about the user to store?
- How to get this information?

This thesis focuses on the last problem. However, we need to have answers to other questions as well in order to build a system. In the following, some answers to these questions will be suggested.

How to organize a product taxonomy?

Usually, product classifications used by on-line stores reflect either physical organization of their brick-and-mortar counterparts or store management preferences. Such ontolgies are store-specific and often difficult to navigate. In this work, we aim at creating a more general system that would be applicable in different stores and would be consistent and transparent. Therefore, the product ontology has to have the following characteristics:

- it should be standardized as much as possible (to facilitate navigation);
- it should be universal rather than country specific (to facilitate international trade);
- it should be general and cover different kinds of products;
- including new products and categories should be easy;
- creating it should be fast and cheap.

One way to satisfy all these requirements is to take existing product taxonomies already adopted in the industry and apply them with minimum modifications. The current UN effort aiming at creation of a universal standard product classification UNSPSC [83] is, in our opinion, the best starting point. This classification covers most of the domains and will soon become a standard. Since this classification lacks granularity (it stops at the "department" level) we need some other source for more detailed information. North-

American industrial classification NAICS [84] provides necessary information to fill this gap.

What information about products to represent?

Different people look for different things even while shopping in the same store, therefore it is not possible to pre-select information without creating a certain bias or removing the information some customers might be interested in. We believe, that the manufacturer's descriptions such as product labels are the best source of information about products. The content of these labels is more or less standard across all products of the same type which provides for easy comparisons, consumers are used to reading the labels and would expect to find this very information in the product descriptions in on-line stores; the labels reflect the results of long-term research and observation and are more likely to contain data that is relevant to the consumer than any ad hoc format.

What information about the user to store?

The decision about the choice of attributes for the user model to be used in this work was made based on two major sources:

- marketing research in the domain chosen for the prototype – wellness shopping. This information can be found in publications by Hartman Inc.[28], specializing in wellness consumer research, as well as in government publications (e.g. Statistics Canada [85], Government of Alberta [69] and others);

[28] http://www.hartman-group.com/

- analysis of the product characteristics to extract a list of repeating features that could be abstracted at the category and store level (e.g., product labels for most foods contain information about nutritional values such as number of calories, quantity of fat, etc.).

In the next sections we will describe the proposed architecture of an agent-based tool for knowledge acquisition and its proof-of-concept prototype developed and implemented.

4.2 Knowledge Acquisition for User Model

It has been shown in Chapter 2 that none of the existing KA systems seems directly applicable to capturing information about the user in the context of e-commerce. While many of the existing KA tools possess a user-friendly interface and allow creation of sophisticated knowledge bases, ontologies, or conceptual models, they all require at least some training and they are relatively slow. In case of user modeling for e-commerce we need to find another approach that would make the KA tool intuitive to use even for a naïve user with no previous knowledge in computers, marketing, or psychology. In order to gain this advantage, we can sacrifice granularity because the user model does not need to make very subtle distinctions in attribute values, as is done, for instance, by WebGrid and other KSSn tools. Also, decreasing granularity would decrease the risk of error. Knowledge capture for user profiling on the web can benefit from a combination of techniques that will take some load off the user and put it on the system that will extract information from observation (activity logs, browsing behavior, shopping history, etc.) as described earlier.

Knowledge acquisition in the context of electronic commerce imposes certain constraints on methods that can be employed for information capture. It is essential to minimize the need for direct user input and to put more weight on information extraction from results of observation of user actions. Thus, the knowledge capture for user profiling on the web can benefit from a combination of techniques that will take some, but not all, load off the user and put it on the system which will extract information from observation. For this purpose, activity logs, browsing behavior, shopping history, etc. can be used.

There exist three major approaches to data collection for user profiling [86]:

1. ***Explicit profiling*** occurs when users enter data themselves by filling forms and answering questionnaires. This approach is good because the user has control over the information he/she supplies to the system that builds trust. This approach puts less load on the system. On the other hand, time and effort spent by users on entering their data into the system should be minimized. An average customer wants convenience and speed at all stages of his/her interaction with the system and not just the promise of future enhancements. Therefore another approach becomes necessary for fine-tuning and maintenance of user model.
2. ***Implicit profiling*** can fill that gap between the amount of information desirable for the system and the amount of user involvement. This method consists of tracking user behavior and drawing conclusions using various machine learning techniques. The major downside of this approach is unreliability of the

algorithmically obtained inferences (there will be a lot of noise that is difficult to interpret).

3. Using ***legacy data*** (such as transaction and browsing logs, purchase history) for complementing and updating the user profile seems to be a better choice than implicit profiling. This approach capitalizes on user's personal shopping history (previous purchases, subscriptions, information requests, etc.) kept in memory on the seller's server. This data is more reliable than browsing logs and can be a rich source of information about a repeat customer.

We use explicit profiling as the first step in KA for user modeling in e-commerce. Based on this information an initial user profile is built during the customer's first visit to the store. If the customer returns, this initial profile is updated based on the shopping history. Therefore there are two types of interaction:

1 – user-system interaction for knowledge acquisition purposes. This kind of interaction (Type-1 interaction) corresponds to explicit profiling and consists of a dialogue between system (agents) and the user for eliciting knowledge about user's characteristics and needs;

2 – transaction based interaction (Type-2 interaction) covers implicit profiling and use of legacy data where there is no direct dialogue between the system and the user. All the information is acquired from observation and analysis of current and/or past user's actions.

To make the creation of the user model easy and pleasant for a user, we propose to provide him/her with an intelligent assistant that would help in the task. Such an agent will be proactive and relatively autonomous. In the proposed system this role is played by a community of agents who get the data from the user via user interface, capture the information by observing user actions, process this data, and perform data verification if needed with minimal load on the user's time and effort.

PUMe incorporates all four agent tasks (described in Section 3.2. as Assist 1 to Assist 4.) The Agent A1 takes care of initialization of the profile (Assist-1) and dynamically updating it based on transaction data (Assist-2). The Agent A2 takes care of interaction with the user when explanation or validation of results is necessary (Assist 2 and 3). The user profile is stored locally and agent A1 ensures the safe information exchange between the user and other agents and the system software, by filtering incoming queries (Assist-4). The agents are organized into a KA subsystem that also contains other modules as shown in Figure 4.3 (p. 83).

The KA sub-system's goal consists of transforming raw input data received from dialogue with the user, through observation or event tracking, into a standardized format that can be used by the rest of the e-commerce system.

The KA sub-system tasks include:

- receiving data from user
- processing of data

- validating and fine-tuning of information about user
- storage of information about user

Therefore two major units of the KA sub-systems are Agent A1 and Agent A2.

The KA sub-system gets the input from two sources: directly from the user via a user interface (Type-1 interaction) and from the system (i.e. the Machine Learning module) based on the transaction history (Type-2 interaction). There are two kinds of data coming directly from the user:

(a) the results of filling forms with personal data (it can range, depending on the specifics of the e-store, from limited amount of personal data typically required for any registration and credit card transactions, to different sorts of additional information such as color preferences, dietary restrictions, etc.);

(b) The elicited knowledge from the user through the dialogue between user and agents aimed at clarifying or validating the system's reasoning. Type-2 interaction occurs inside the system when different modules and agents exchange information in order to dynamically update the profile.

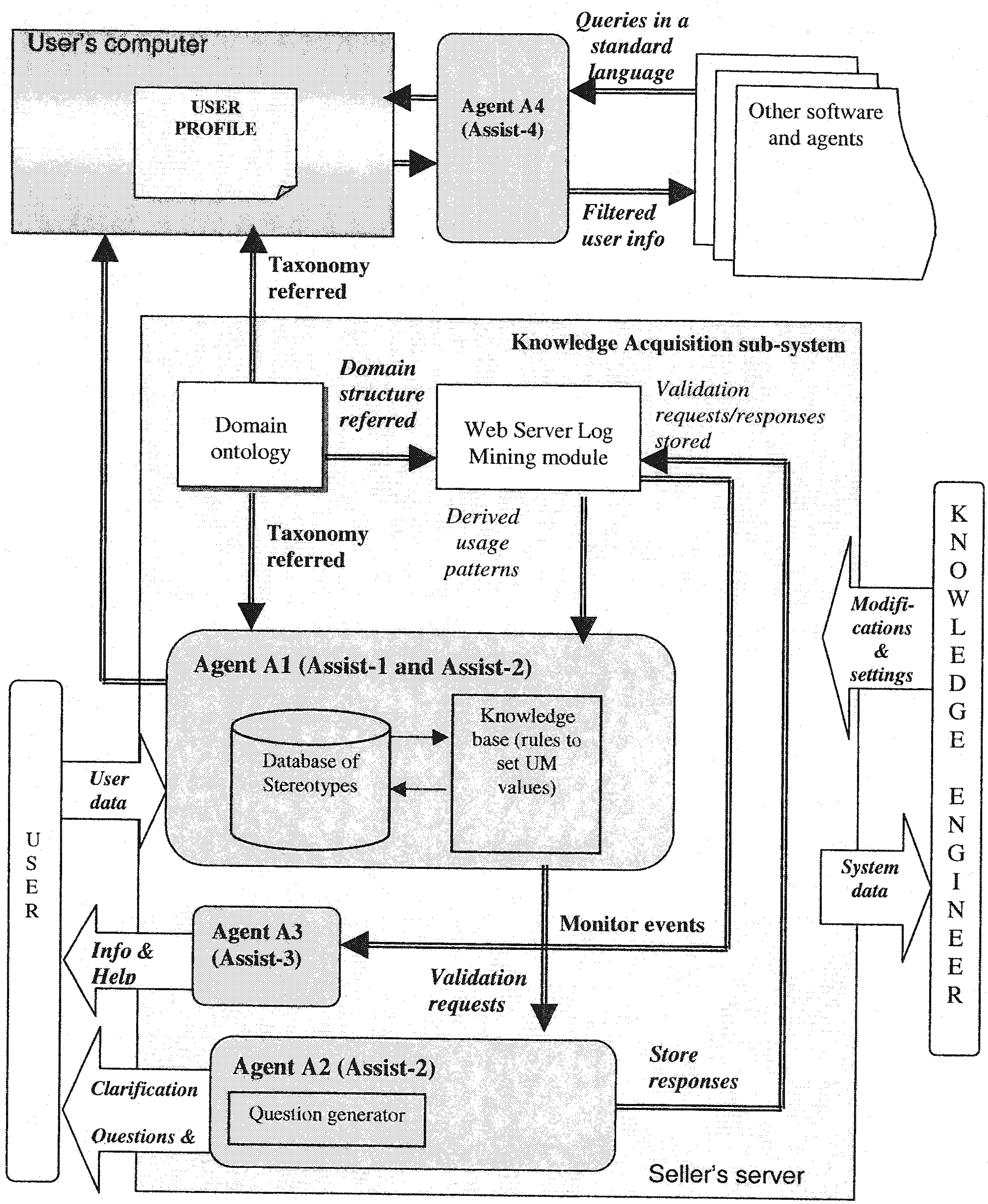

Figure 4.3. Proposed architecture of the KA sub-system

At the first stage of the process of building the user model the Agent A1 gets raw user data from the user interface, transforms it into a format defined by node frame slots, using the agent's knowledge base that includes such information as market definition, product domain ontology, consumer typology, etc. The agent compares the user information to the facts stored in the system and produces values for the user profile in a format required by the user model. This process results in setting initial values for frame slots (features). Thus even with a limited amount of information available the user model is not empty when the customer returns after the initial registration. For example, at the root of the tree (Figure 4.2) the system has stored information about name, age, and some other characteristics of the user. Immediate results of availability of such information in the user profile would be customized presentation or advertisement (e.g., a teenager will not be exposed to content or visual presentation designed for seniors). Some values can be inferred for internal nodes (e.g., if the user Ms. X mentioned "weight control" as main reason for shopping in the wellness store the system can expect that the user will prefer fat-free varieties of all products).

Suppose later the system collects information about the user by observation (browsing logs, purchase history) and updates previously set fillers of the frame slots.

Original value : <fat : low>

Value updated by observation: <product : yogurt
fat : 10 % .....>
implies: <fat : high>

Contra-
diction

Monitoring module will detect the contradiction and create a dialogue with the user to confirm if the profile needs to be updated or the violations against the profile are intentional and should be left alone.

The Agent A1 also has a special set of rules that oversee the modifications made to the profile in order to detect problems. This includes detection of cases where current user actions contradict the information stored in the profile as well as alerting the user, if the modified value gets close the threshold. When a problem has been detected Agent A1 either calls the Agent A2 which initiates a clarification dialogue with the user, or starts a trial period during which it will monitor the trend to decide whether the observed deviation in user behavior was occasional and should be ignored or persistent and should be incorporated into the user model.

Agent A3 monitors what effect the changes made to the user profile have on the system behavior (for example, by analyzing system logs) and supplies the user with appropriate information and advice at a suitable moment.

There are four typical use cases associated with this approach to user modeling.

Use Case 1. First-time registration and profile initialization.

This use case involves Type-2 interaction and corresponds to Assist-1 described above. The typical scenario for this use case develops as follows:

At registration the user keys in relevant information (personal data, appropriate additional facts, related to the current domain). At this stage the knowledge acquisition tool uses the information that is typically supplied by the user to a system for transaction authorization (thus the user does not have to put extra effort into creating the profile and is not required to give the system additional personal data). Some of the stereotype-based values can be set even before any interaction with the user (for example, the system may know that there is a 60% chance that a shopper would give preference to organic products, and the user normally shops in specialized small shops rather than in supermarkets).

This user data goes to the Agent A1. The agent uses a set of rules that compare user values to the information stored in the database of stereotypes, consults the domain hierarchical organization (ontology) and arrives at values to be assigned to the slots of frames stored at PIE nodes. A certainty factor is calculated for each value. Certainty factors of leaf values of the PIE DAG influence the degree of belief in the facts stored in internal nodes higher in the hierarchy (for example, if the user states his/her interest in a vitamin enriched vegetable juice used as dietary supplement the certainty factor is increased for juices as well as for dietary supplements; see Figure 4.2). At this point the user model available in the system is stereotype-based. It reflects typical characteristics of people of a certain age, sex, family situation, etc. (the values were assigned based on population statistics and marketing research data).

Results of this stage allow the system to start the next session equipped with certain degree of knowledge to be able to attempt the customization of the presentation, advertisement, or search.

The task flow for Use Case 1 can be described as follows:

- User Interface Agent :
    - Get the data

- Pass it to the Agent A1

- Agent A1:
  - Compare user data to the stereotypes
  - Assign default values to attributes at the user profile nodes

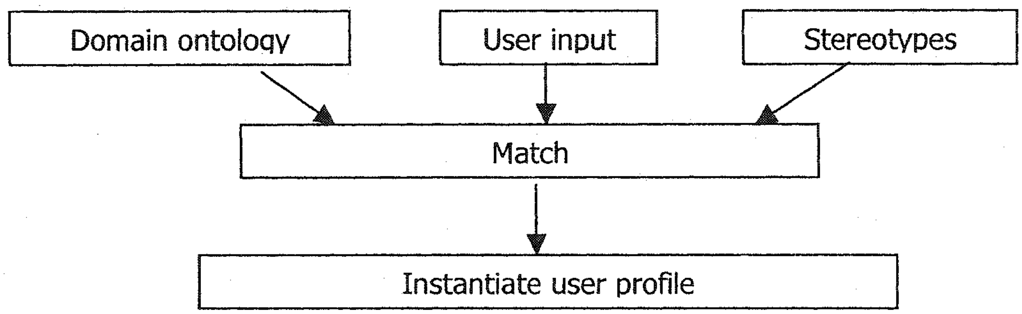

Figure 4.4. Framework for Use Case 1

To summarize, in Use Case 1 Agent A1 brings to bear function Assist-1 to support the knowledge acquisition for profile initialization.

Use Case 2. Second time visitor profile update.

This case describes the process of dynamic adaptation of the user profile based on activity logs. It involves Type-1 interaction and corresponds to Assist-2 described in Section 3.2.

The typical scenario for this use case develops as follows:

When a user who is already registered returns, his/her actions (time spent viewing suggestions based on initial profile, browsing sequences, shopping cart decisions, etc.) are recorded in a user activity log. These data are later analyzed by Agent A1 in order to extract relevant information about user's preferences. Results of the analysis are compared to values originally stored in the user model. In case of minor changes new, individual values have preference over those based on stereotypes. If the difference is very significant (e.g., while filling the questionnaire the user informed the system that he is vegetarian which resulted in giving the slot "food-preferences" a value of "not meat" with certainty 1; but while shopping the user showed particular interest in deli products and even bought some ham) the Agent A1 can not make a decision and calls Agent A2 to clarify the problem. Then the system might ask the user about his/her preference in this particular case in form of simple yes/no questions (see Use Case 3). User answers to the questions overwrite previously set values. Now the user model built upon the values of slots in the frames is more individualized - wherever possible stereotypes are replaced by individual user preferences and now the customer profile is differentiated from a typical profile for the same demographic segment of the population.

Summing up, knowledge acquisition subsystem components will perform the following actions:

- Machine Learning module
    - Analyses logs
    - Discovers patterns in user's choices
    - Sends results to the Agent A1
- Agent A1

- Pulls out the values stored in the system
  - Replaces values or updates the certainties of attribute values wherever appropriate
  - Informs the user about changes and their consequences at a suitable moment
- Agent A2
  - Monitors the actions of the Agent A1; if necessary - follows scenario for use case 3
- Agent A3
  - Monitors the impact of changes made by Agent A2; if necessary – informs the user of the results

This use case encompasses Assist-2 and Assist-3 functions performed by the entire community of interacting agents including A1, A2, and A3.

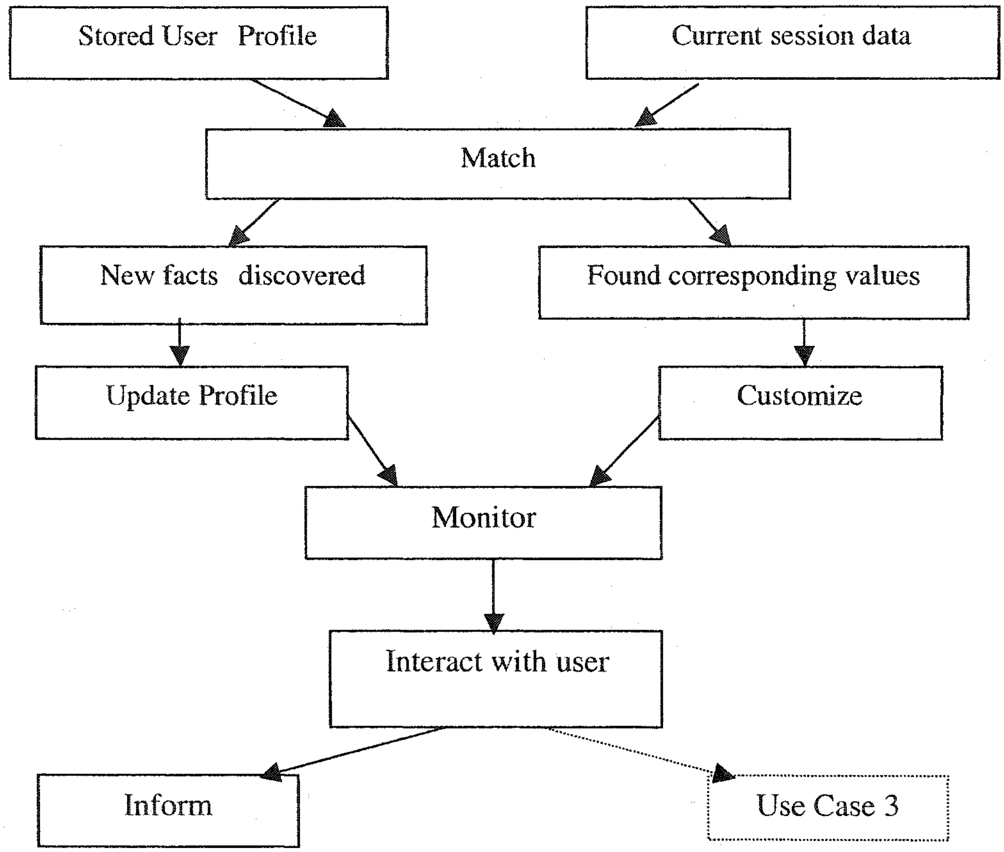

Figure 4.5. Framework for Use Case 2

Use Case 3. Repeat customer profile maintenance.

This use case is a variation of the previous one that occurs when the user model has been set based on stereotypes as well as on user activity logs and the system has to prove or disprove its conclusions based on additional data coming the from user activity logs. This use case corresponds to functions Assist-2 and 3 and makes use of both Type-1 and Type-2 interactions.

One of the possible scenarios of this use case would be the following:

While a customer makes repeated purchases in the store the Agent A1 keeps analyzing his/her activity logs in order to change the values of certainty factors (e.g., if a user is buying some 'high-fiber' products at every visit then the certainty factor associated with high fiber will increase accordingly; on the contrary, if the user at first used to purchase herbal supplements but some time later started ignoring all such products suggested by the system and did not buy any more such supplements the system will decrease the certainty factor for herbal supplements). The Agent A1 also monitors the logs for typical patterns in user's shopping behavior (e.g., typical contents of the shopping basked, brand loyalty, etc.). Significant changes in the user's behavior are registered and validated by Agent A2 (e.g., the system would ask the user about the change in his/her preferences if after having avoided any product containing cocoa the customer starts buying chocolate bars at every visit). Minor changes can be validated by the system itself. In this case the Agent A1 sets a trial period during which the validation agent monitors the particular trend in the user behavior in order to distinguish between occasional deviation from typical pattern and consistent change of behavior.

Confirmed patterns are also reflected in the user model. After several visits to the store the profile of the user becomes highly personalized and reflects mostly the user's individual preferences. Constant monitoring of user activity allows the system to perform dynamic updates of the user model in order to reflect changes in the user's preferences and to detect his/her typical shopping patterns.

The summary of agent's actions for this use case is as follows:

- Web Server Log Mining module and Agent A1 work as in Use Case 2
- Agent A2
  - Monitors the work of the Agent A1
  - Discovers a conflict in the values
  - Analyzes the conflict - was the old value a default? Does any of the conflicting values have a low certainty? Is there additional information in support/against one of the conflicting values?
  - Decides on the degree of urgency of the resolution of the conflict and sends appropriate information requests if needed
    - By initiating a Type-1 interaction with user
    - By collecting more data using Type-2 interaction

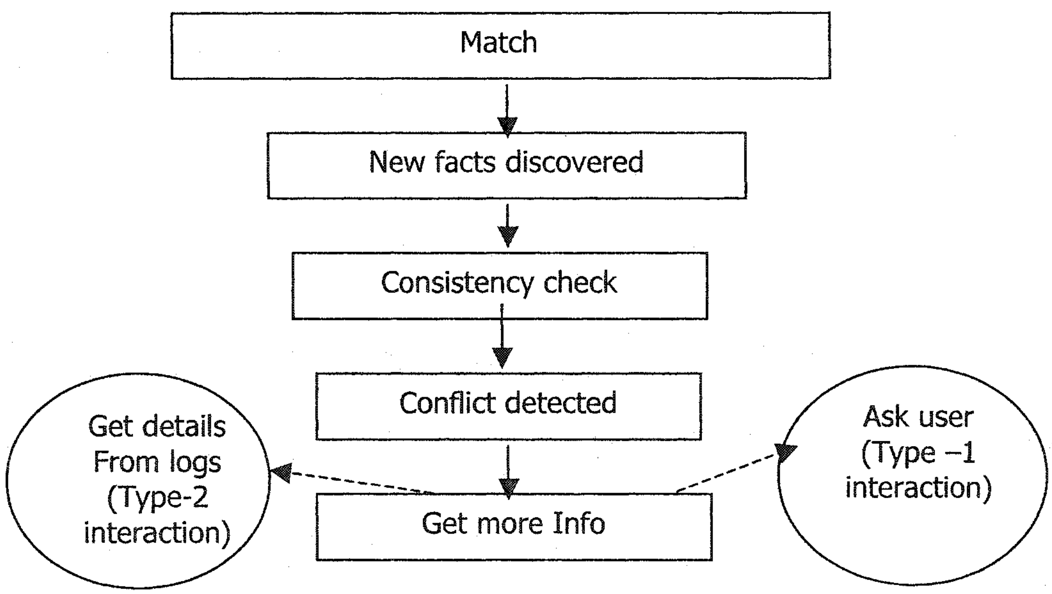

Figure 4.6. Framework for Use Case 3

Use Case 4. Information exchange with the outside world.

This use case is served by Agent A4 and involves Type-2 interaction but it can be also initialized by the user (Type-1 interaction).

There are two main scenarios for this use case depending on the direction of communication.

In the first scenario the user or his agents request the information that is not available in the system. In this case Agent A4 initializes a search for other sources of information. These sources include different forms of documents as well as other systems and agents. For example, if the user is looking for information on possible side-effects of particular weight-loss product the Agent A4 can extract necessary data from the product monograph, or request this information from the vendor's or producer's agents, or find people who used this product and are willing to share their experience. In the last case, A4 can supply the user with available contact information.

In the second scenario, the query comes from outside and the agent A4 plays the role of a gate-keeper by filtering the incoming requests and limiting the amount of information to be supplied in response to a query based on constraints set by user (e.g., a list of friendly agents) and on world knowledge (e.g., information can be given to a reliable long-term partners but not to an unknown company).

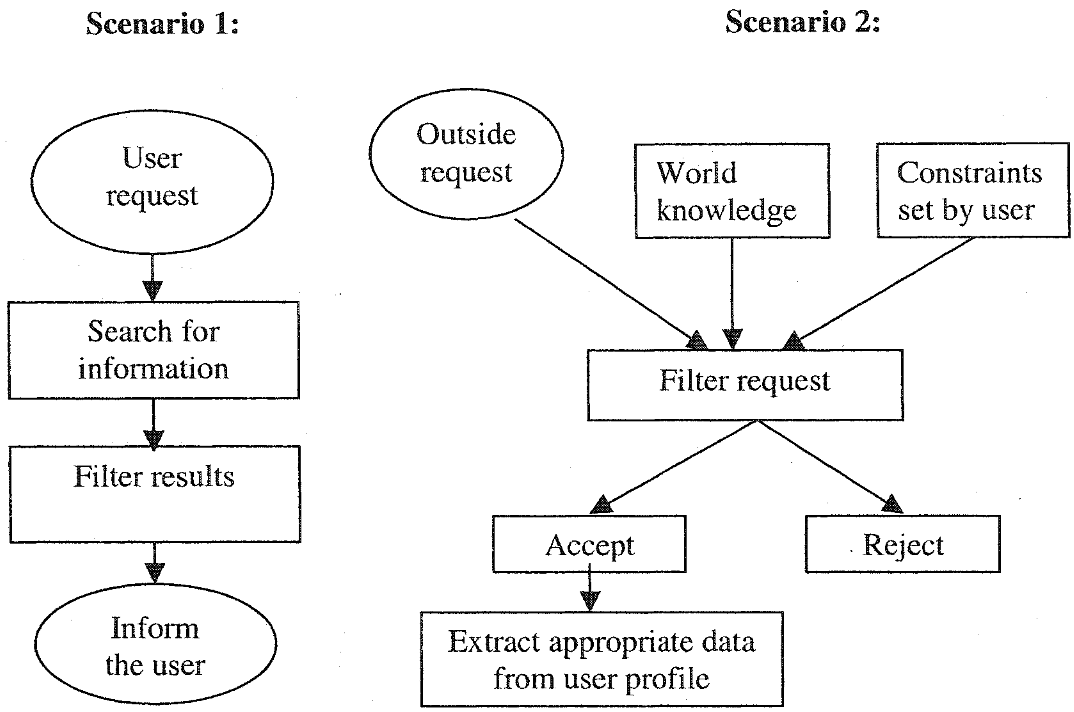

Figure 4.7. Framework for Use Case 4

4.3 Implemented Prototype of the Knowledge Acquisition Tool for E-Commerce

A proof-of-concept prototype of the system using the proposed architecture of KA tool has been implemented in order to test the architecture on a small scale and to illustrate the first three use cases described in Section 4.2[29]. It is a web-based application that allows the user to register with the system, to browse a small list of products as well as to perform a search in the product database and to make purchases. In this section three main scenarios are presented. A more detailed description of the major components of the prototype and of supported functionalities is given in the Appendix A.

[29] Use Case 4, Information exchange with outside world, has not been implemented because it involves a large community of agents that communicate with each other.

Use Case 1: First time registration and profile initialization

The user who comes to the store for the first time can either browse the products using Browse or Search options provided on the Home page and register only at the moment of the check-out or register first and browse/search after. In both cases the system will ask the user for some information before checking out. The initial information for the profile is collected through a fill-in form (Figure 4.8) that is then processed using rules based on statistical data. The agent gives brief explanations for some of the questions when the mouse pointer is on it.

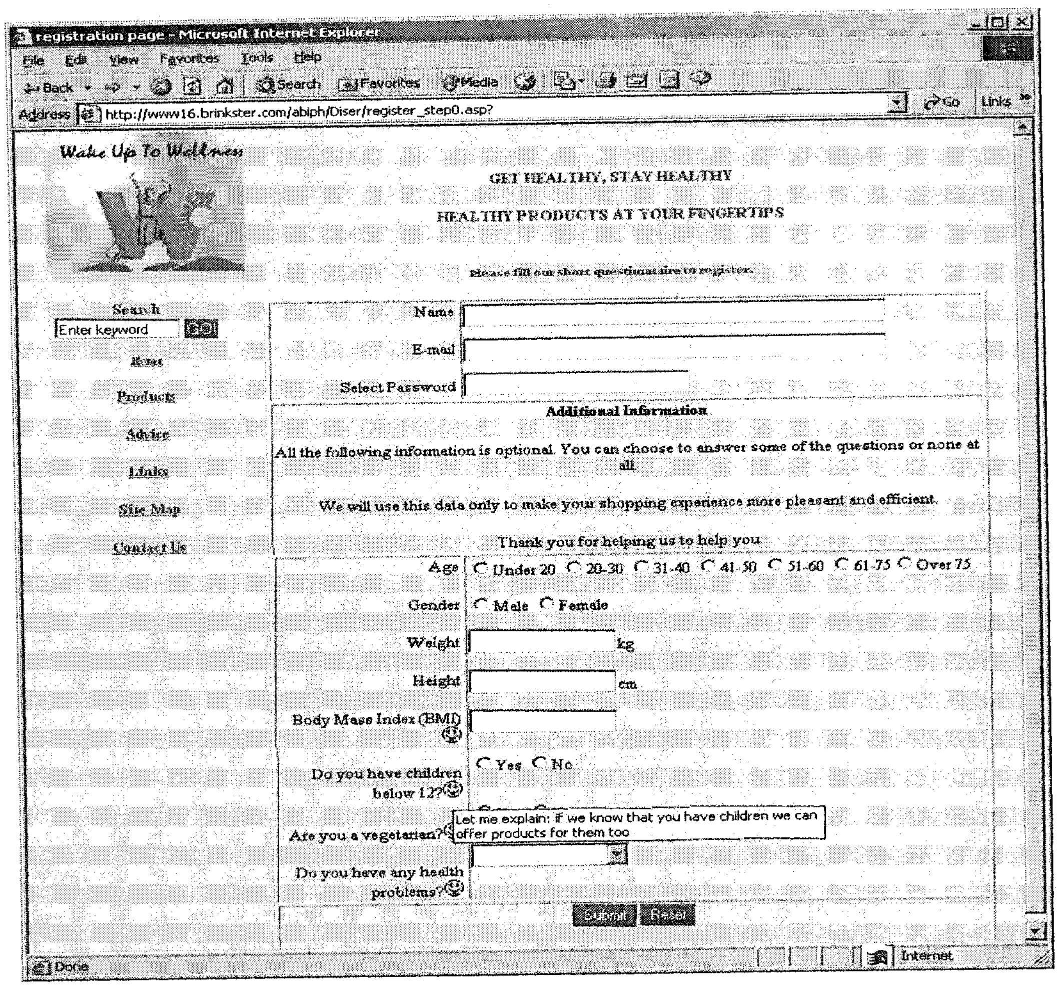

Figure 4.8. Registration Page

Use Case 2: Second time visitor profile update.

After the user has registered on the web site his/her profile is remembered in the database and at the next visit he/she can log in to get personalized service. For example, when the user is known to the system a welcome message is displayed and a customized list of suggested products is shown when appropriate (Figure 4.9).

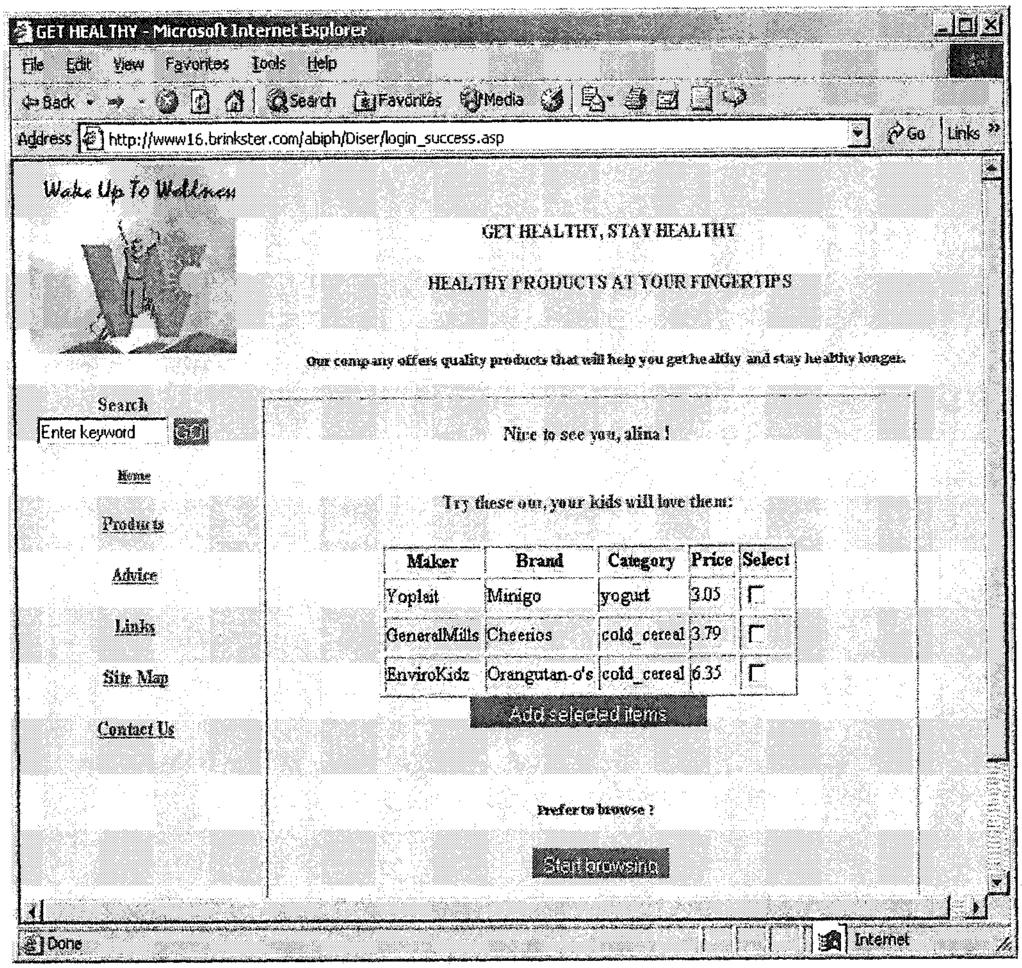

Figure 4.9. Personalized product selection at log in

After logging in the user has an option to browse, search or select one of the products from a personalized suggestions list. For example, different advice will be displayed for an unknown user (Figure 4.10) and for a registered user who has children (Figure 4.11). Displays of useful links are also customized depending on the user's profile.

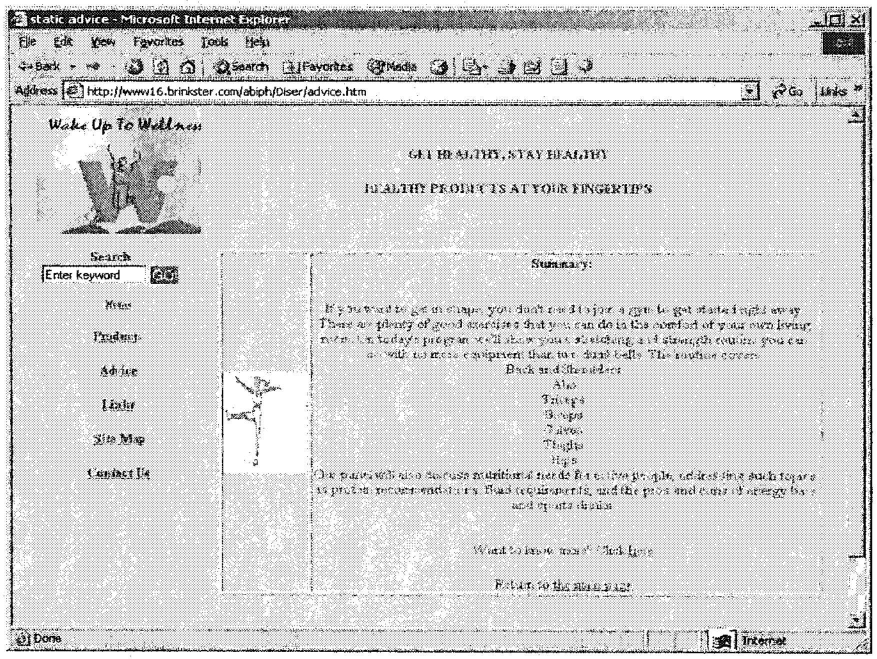

Figure 4.10. Advice for an unknown user

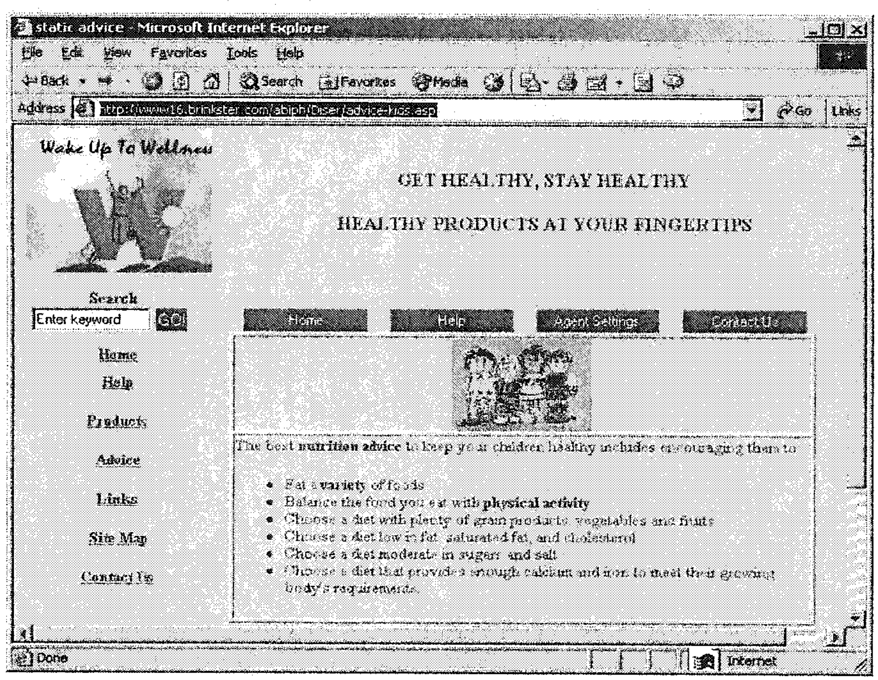

Figure 4.11. Customized advice for a parent

The prototype combines explicit profiling based on values entered by the user at the registration time and from answers to agent questions with implicit profiling. In the last case we used a ShoppingCart metaphor and a set of rules that convert data stored in the ShoppingCart data structure into values to be stored in the user profile. A ShoppingCart is an array that holds the information about user's identity (the user name) and about products chosen by this user during the current session. This array is passed from one web page to another during the entire session until the order is complete and at each web page of the site the values stored in the Shopping Cart are updated based on user's choices. At the checkout the contents of the ShoppingCart are used to update the user profile in order to reflect the data of the session (e.g. to change values of the attributes or to increment/decrement the certainty of previously set attribute values, etc.).

Use Case 3: Repeat customer profile maintenace

At the checkout the system can detect a conflict between currect purchasing activity of the user and his/her profile. If such conflict is rated as important by the system (e.g., bacause of hight certainty assigned to the value in the profile) a question can be asked to the user to validate system's inferences. For example, Figure 4.12 shows the question that is displayed if the user who is known to the system as vegetarian based on the information supplied at registration (which automatically leads to assigning certainty 1 to this fact) buys meat.

The user's answer to the question is processed by a script associated with the validation page and values and/or certainties are changed accordingly.

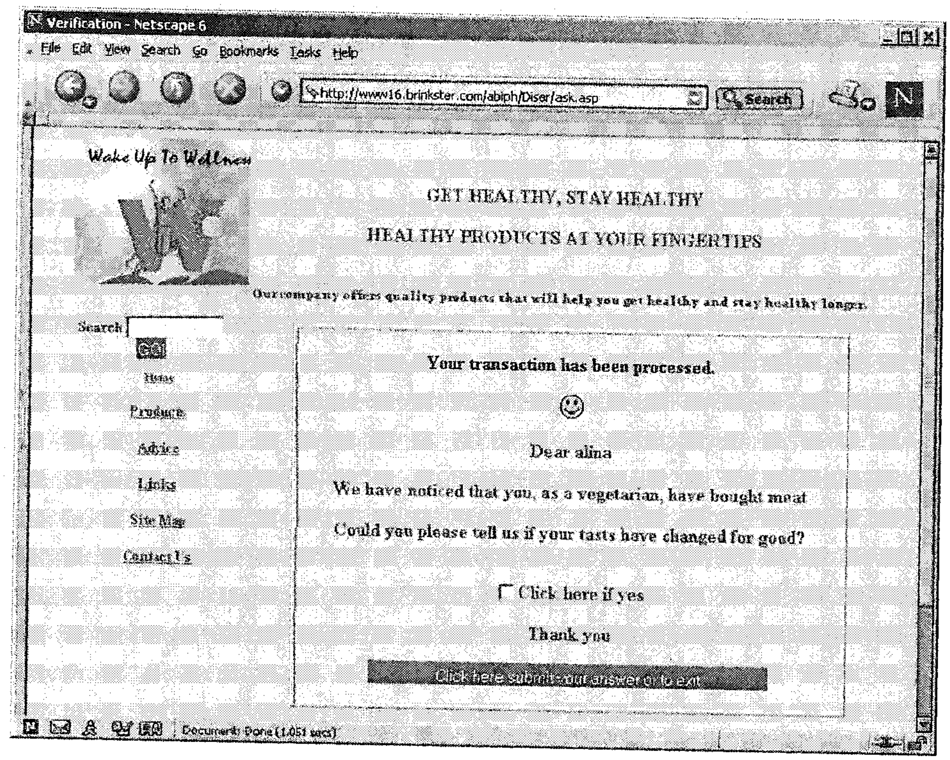

Figure 4.12. Validation question

4.4 Conclusion

In this chapter we presented the architecture of a tool for agent-supported knowledge acquisition for dynamic user profiling in e-commerce. This architecture represents a community of software agents that perform various tasks described in Chapter 3: they assist the user

- in initializing the profile based on user input and on stereotypes (Agent A1),
- in dynamically updating the user profile based on transaction data (Agent A1),
- in making informed decisions about changes and their impact on the system's performance (Agents A1, A2, and A3),

- in communication with the "outside world" including other agents and software systems (Agent A4).

The proposed architecture allows dynamic user profiling where the system shares the burden of knowledge acquisition with the user and the knowledge engineer. This aspect, we hope, will reduce time and effort spent by the customer on supplying the system with the information relevant to his/her profile to influence the system's inferences.

The four use cases described in this chapter show how the agents that make up this architecture can participate at different stages of the knowledge acquisition process and assist the user directly (via Type-1 interaction) or indirectly (via Type-2 interaction) in the process of initializing and updating the profile. Constant Type-2 (transaction based) interaction at all stages of the process can considerably speed up the process of fine-tuning the profile in order to make it more efficient and to achieve more complete user satisfaction faster. At the same time, use of implicit profiling will reduce the load put on the user. The user can find a trade-off between these two types of interaction by explicitly setting his/her preferences or by allowing the system to deduce the constraints from his/her actions.

A proof of concept prototype has been implemented to test the architecture on a small scale, to demonstrate the feasibility of the idea and to illustrate the use cases. In our opinion, the prototype indicates that the proposed architecture can be fruitfully used in electronic commerce. The prototype demonstrates the implementation of the architecture on the small scale but it also supports features for scalability.

Chapter 5

CONCLUSIONS

5.1 Contributions

In the recent years the focus of customer relationship management in the area of e-commerce started shifting from attracting new visitors to retaining the existing customers. This precipitated changes in approaches to personalization from recommender systems to more individualized models. One of the major problems encountered by this approach is the need to continuously collect considerable amount of information about customers without putting demands on their time and effort and at the same time without compromising the sense of control and trust. We believe that agent-supported knowledge acquisition for dynamic personalization is a very promising approach that can solve some of these problems.

The contributions of the research described in this thesis are summarized in this section. The major points are as follows:

Interactive knowledge acquisition tools reported in the literature have been reviewed and classified. The suggested classification divides all such tools into two broad categories – classical KA tools that are designed for use by knowledge engineers and/or experts and modern tools that either were created for naïve users without any special training or for automatic collection of information using data/web mining techniques.

After exploring the notions of dynamic personalization and agent-supported knowledge acquisition for user modeling we identified four major aspects of agent assistance in this process: assistance during the initialization of the user profile; assistance in profile updating based on web mining results; assistance in tracking the effects of changes in the user profile; and assistance in trust related issues such as communication with other agents.

Based on the conducted analysis an architecture for a knowledge acquisition tool for dynamic user profiling in electronic commerce has been proposed. Four major use cases related to agent-supported user profiling have been described. A small-scale prototype that has been implemented has demonstrated the feasibility of the proposed architecture.

5.2 Future Work

The research started by this thesis can develop in several interesting directions. Among them are the further development of the different forms of agent's assistance in the process of building and maintaining the user profile. In particular, Assist-4, which covers an agent's role in mediating interaction with other agents and systems is of major interest. Further exploration of impact of virtual communities and opinion based filtering of user's choice and profile dynamics seems to be a very interesting direction to follow in the future.

Future work could also involve the implementation of a prototype system on a larger scale with further validation and verification of the suggested approach to user profiling.

In the present work data and web mining aspects of the building of the user profile have been covered very briefly, but they deserve a more detailed and in depth study in the future.

Aspects of the knowledge acquisition related to usability and to different forms of the user interface (e.g., multi-modal mixed-initiative interaction between the user and the system) also seem to be an interesting direction.

REFERENCES

[1] Statistics Canada, Statistics Canada, "Electronic Commerce: Household Shopping on the Internet", *The Daily,* September 19, 2002, http://www.statcan.ca/Daily/English/020919/d020919b.htm

[2] P. Stanger et al., "The State of Online Retailing 4.0: A Shop.org study by Boston Consulting Group", *Boston Consulting Group,* Report, May 2001, http://www.bcg.com/publications/files/Summary_Shop_org.pdf

[3] B.J. Pine, *Markets of one: creating customer-unique value through mass customization*, Boston, Mass.: Harvard Business School Press, ©1993.

[4] B.J. Pine and J.H. Gilmore *The Experience Economy: Work is Theatre and Every Business a Stage,* Boston : Harvard Business School Press, ©1999.

[5] B. Violino, "Customer at the Core – Businesses are Realizing that to Stay Competitive, they have to Become More Responsive to their Customers, Using IT to Keep them Satisfied and Loyal", *Information Week*, September 27, 1999, http://www.techweb.com/se/directlink.cgi?IWK19990927S0057

[6] B. Fryer, "So Happy Together: The hottest mantra in corporate computing is customer relationship management", *CFO Magazine,* June 1999, http://www.cfo.com/article/1,5309,1425|||3,00.html.

[7] U. Manber, A. Patel, and J. Robison, "Experience with Personalization on Yahoo!", *Communications of the ACM,* August 2000, vol. 43, no. 8, pp. 35-38.

[8] IBM High-Volume Web Site Team. *Web Site Personalization,* January 2000, http://www7b.boulder.ibm.com/wsdd/library/techarticles/personalize.html/#resources

[9] G. Adomavicius and A. Tuzhilin, "Expert-Driven Validation of Rule-Based User Models in Personalization Applications", *Data Mining and Knowledge Discovery,* vol. 5, no. 1/2, pp. 33-58, 2001.

[10] E. Colkin, "Personalization Tools Dig Dipper", *InternetWeek.com*, August 23, 2001, http://www.internetweek.com/story/IWK20010823S0003

[11] Y. Gil and J. Kim, "Interactive Knowledge Acquisition Tools: A Tutoring Perspective", *Proc. 24th Annual Meeting of the Cognitive Science Society*, 2002, http://www.isi.edu/expect/papers/Interactive-KA-Tools-gil-kim-02.pdf

[12] N.J. Cooke, "Varieties of Knowledge Elicitation Techniques", *International Journal of Human-Computer Studies*, vol. 41, issue 6, December 1994, pp. 801-849.

[13] UsabilityNet, *Tools and Methods. Card Sorting,* © 2003, http://www.hostserver150.com/usabilit/tools/cardsorting.htm

[14] Information & Design, *What is Card Sorting?*, © 2003, http://www.infodesign.com.au/usabilityresources/design/cardsorting.asp

[15] J. Hom. *Card Sorting*. ©1996, http://jthom.best.vwh.net/usability/cardsort.htm

[16] IBM, *EZSort*, http://www-3.ibm.com/ibm/easy/ eou_ext.nsf/Publish/410

[17] National Institute of Standards and Technology (NIST). Information Access Division. Visualization and Usability Group, *WebCAT. Category Analysis Tool. Demo,* http://zing.ncsl.nist.gov/WebTools/WebCAT/cgi-bin/ue_setup_demo.exe

[18] G.A. Kelly, *The Psychology of Personal Construct,* New-York: Norton, 1955.

[19] C. Corbridge et al., "Laddering: Technique and Tool Use in Knowledge Acquisition", *Knowledge Acquisition,* vol. 6, 1994, pp. 315-341.

[20] P. Kottle and R.E. Turner, *Marketing Management. Analysis, Planning, Implementation, and Control*, Canadian 8th ed. Scarborough, Ont.: Prentice Hall Canada Inc.,1995.

[21] P. Underhill, *Why We Buy: The Science of Shopping*, New-York: Simon and Schuster, 1999.

[22] E. A. Feigenbaum, "Knowledge Engineering: The Applied Side Of Artificial Intelligence", *Annals Of The New York Academy Of Sciences*, 1984, pp. 91-107.

[23] R. Davis, "Interactive Transfer of Expertise: Acquisition of New Inference Rules", *Artificial Intelligence*, vol. 12, no. 2, 1979, pp.121-157.

[24] R. Davis and D.B. Lenat, *Knowledge-based systems in Artificial Intelligence*, New-York, McGrow-Hill International Book Co., v. 1982.

[25] M.A. Musen, et al., "Use of a Domain Model to Drive an Interactive Knowledge Editing Tool", *International Journal of Man-Machine Studies*, vol. 26, no. 1, January 1987, pp. 105-121.

[26] S. Marcus, "Taking Backtracking with a Grain of SALT", *International Journal of Man-Machine Studies*, vol. 26, no. 4, April 1987, pp. 383-398.

[27] L. Eshelman et al., " MOLE: A Tenacious Knowledge Acquisition Tool", *International Journal of Man-Machine Studies*, vol. 26, no. 1, January 1987, pp. 41-54.

[28] B.R. Gaines and M.L.G. Shaw, "Knowledge Acquisition Tools Based on Personal Construct Psychology", *Knowledge Engineering Review*, vol. 8, no. 1, 1993, pp. 49-85.

[29] J.H. Boose, "A Survey of Knowledge Acquisition Techniques and Tools", *Knowledge Acquisition,* 1989, 1, pp. 3-37.

[30] C.M. Kitto and J.H. Boose, "Selecting Knowledge Acquisition Tools And Strategies Based On Application Characteristics", *The Foundations Of Knowledge Acquisition. Knowledge Based Systems,* vol.4, J.Boose & B. Gaines (eds.), Academic Press, London, 1990, pp. 257-268.

[31] G. Kahn, S. Nowlan, and J. McDermott, "MORE: an Intelligent Knowledge Acquisition Tool", *Proc. 9th Joint Conference on Artificial Intelligence*, Los Angeles, California, 1985.

[32] G.R. Yost, "Acquiring Knowledge in Soar", *IEEE Intelligent Systems and their Applications*, vol. 8, no. 3, June 1993, pp. 26-34.

[33] S. Huffman, C. Miller, and J. Laird, "Learning for Instruction: a Knowledge-Level Capability within a Unified Theory of Cognition", *Pro.15th Annual Meeting of the Cognitive Science Society,* Boulder, Colorado, 1993.

[34] S.B. Huffman and J.E. Laird, "Flexibly Instructable Agents", *Journal of Artificial Intelligence Research,* vol. 3, 1995, pp. 271-324, http://citeseerr.nj.nec.com/huffman95flexibly.html

[35] McGuinness et al., "The Chimaera Ontology Environment", *Proc.17th National Conference on Artificial Intelligence (AAAI),* 2000).

[36] B.W. Porter, R. Bareiss, and R.C. Holte, "Concept Learning and Heuristic Classification in the Weak –Theory Domains", *Artificial Intelligence,* vol. 45, 1990, pp. 229-163, http://citeseer.nj.nec.com/porter90concept.html

[37] P. Clark et al., "Knowledge Entry as the Graphical Assembly of Components: The SHAKEN System", *Proc. K-CAP*, Victoria, BC, Oct. 2001, http://citeseer.nj.nec.com/493298.html

[38] A. Ginsberg, S.M. Weiss, and Politakis, "SEEK2: A Generalized Approach to Automatic Knowledge Base Refinement", *Proc. 9th International Joint Conference on Artificial Intelligence (IJCAI 1985)*, vol. 1, Los Angeles, CA, August 1985, Morgan Kaufmann, 1985, pp. 367-374.

[39] Megaputer, *Data Mining*, © 2003, http://www.megaputer.com/dm/ reasonsforgroth

[40] J. Kim and Y. Gil, "Acquiring Problem-Solving Knowledge from End Users: Putting Interdependency Models to the Test". *Proceedings of AAAI-2000*, 2000, pp. 223-229.

[41] W.E. Grosso et al., *Knowledge Modeling at the Millennium (The Design and Evolution of Protege-2000),* 1999, http://www-smi.stanford.edu/pubs/SMI_Reports/SMI-1999-0801.pdf

[42] Stanford Medical Informatics. *Protégé-2000 User Guide,* 2003, http://www-smi.stanford.edu/projects/protege/protege-2000/doc/users_guide/index.html or http://protege.stanford.edu/publications/UserGuide.pdf

[43] P. Speel et al., "Knowledge Mapping for Industrial Purposes", *Proc. 12th Workshop on Knowledge Acquisition, Modelling Management* (KAW'99), 1999, http://sern.ucalgary.ca/KSI/KAW/KAW99/papers/Speel1/

[44] G. Shreiber at al., *Knowledge Engineering and Management: The CommonKADS Methodology,* MIT Press, 2000.

[45] G. Shreiber et al., "CommonKADS: A Comprehensive Methodology for KBS Development", *IEEE Expert, Intelligent Systems and their Applications,* vol. 9, no. 6, December 1994, pp.28-37.

[46] B.R. Gaines and M.L.G. Shaw, *Knowledge Acquisition Tools Based on Personal Construct Psychology*. 1995, http://ksi.cpsc.ucalgary.ca/articles/KBS/KER/

[47] B.R. Gaines and M.L.G. Shaw, *WebGrid II: Developing Hierarchical Knowledge Structures from Flat Grids*, 1998, http://repgrid.com/reports/KBS/WG/WG.pdf

[48] B. Gaines and M.L.G. Shaw, "Developing for Web Integration in Sisyphus-IV: WebGrid-II Experience", 1996, http://www.repgrid.com/reports/KBS/Sis4/

[49] B. Swartout and Y. Gil, "Flexible Knowledge Acquisition Through Explicit Representation of Knowledge Roles", *1996 AAAI Spring Symposium on Acquisition, Learning, and Demonstration: Automating Tasks for Users*, Stanford, CA, March 1996, http://www.isi.edu/expect/link/publications.html

[50] J. Blythe et al., "An Integrated Environment for Knowledge Acquisition", *Proc. Conference on Intelligent User Interfaces* (IUI'0), January 14-17, 2001, http://www.isi.edu/expect/papers/blythe-kim-rama-gil-iui01.pdf

[51] S. Ansari et al., "Integrating E-Commerce and Data mining: Architecture and Challenges", *ICDM'01: The 2001 IEEE International Conference on Data mining,* 2001, http://www.kohavi.com

[52] O.R. Zaiane, M. Xin, and J. Han, "Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs", *Proc. Advances in Digital Libraries Conf.* (ADL'98), Santa Barbara, CA, April 1998, http://citeseer.nj.com/zaiane98discovering.html

[53] B. Berendt and M. Spiliopoulou, "Analysis Of Navigation Behaviour In Web Sites Integrating Multiple Information Systems", *The VLDB Journal,* vol. 9, no. 1, 2000, pp. 56-75.

[54] H. (K.) Dai and B. Mobasher, "Using Ontologies to Discover Domain-Level Web Usage Profiles", *Proc. Second Workshop on semantic Web mining at the 6th European conference on Principles and Practice of Knowledge Discovery in databases (PKDD'02),* Helsinki, Finland, August 2002, http://maya.cs.depaul.edu/~mobasher/pubs-subject.html

[55] B. Mobasher et al., "Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization", *Data Mining and Knowledge Discovery,* vol. 6, 2002, pp. 61-82.

[56] F. Tanudjaja and L. Mui, "Persona: a contextualized and personalized web search", 35th Hawaii International Conference on System Sciences (HICSS-35 2002), CD-ROM / Abstracts Proceedings, 7-10 January 2002, Big Island, HI, USA. IEEE Computer Society, 2002 - Track 3, p. 67.

[57] D. Rubini, "Overcoming the Paradox of Personalization: Building Adoption, Loyalty, and Trust in Digital Markets", *Design Management Journal,* vol. 12, no.2, Spring 2001, pp. 49-54, http://www.dmi.org/dmi/html/publications/journal/ pdf)01122RUB49.pdf

[58] ILOG Press Release, *Open Market And ILOG Collaborate To Bring Highest Level Of "Dynamic Personalization" To E-Merchandising Customers*, 2000, http://www.ilog.com/corporate/releases/us/000403_openmarket.cfm

[59] A. Lockerd and E. Arroyo, "Personal Data for Personal Use: Case Studies in User Modeling for Context-Aware Computing ",*AAAI Fall Symposium on Etiquette for Human Computer Interaction,* November 15-17, 2002, Sea Crest Conference Center, North Falmouth, MA, pp. 81-84, http://cac.media.mit.edu:8080/contextweb/AAAI-PersonalDataPersonalUse-LockerdArroyo.pdf

[60] T. Hannigan and C. Palendrano, "Personalization Can Be Quite Dynamic", *DM Review*, October 2002, http://www.dmreview.com/master.cfm?NavID=198&EdID=5798

[61] K. Goto and Y. Kambayashi, "Dynamic Personalization And Information Integration In Multi-Channel Data Dissemination Enviroments", *Second ACM International Workshop on Data Engineering for Wireless and Mobile Access*, 2001, Santa-Barbara, California, United States, pp. 104 – 109.

[62] J. Bradshaw, "Introduction to Software Agents", *Software Agents,* Menlo Park, Ca.: AAAI Press, 1997.

[63] I.S. Terpidsidis et al., "The potential of Electronic Commerce in re-engineering consumer-retailer relationships through Intelligent Agents" , *Advances in Information Technologies: The Business Challenge*, J.-Y. Roger, B. Stanford-Smith, and P. Kidd., Eds., IOS Press, Amsterdam, Denmark, 1997.

[64] Moukas, A., Guttman, R., and Maes, P. Agent-mediated electronic commerce: An MIT Media Laboratory perspective. In *Proceedings of the International Conference on Electronic Commerce* (Seoul, Korea, Apr. 6–9), ICEC, Seoul, 1998, pp. 9–15.

[65] N. Negroponte, "Agents: From Direct Manipulation to Delegation", J. Bradshaw, ed. *Software Agents,* Menlo Park, Ca.: AAAI Press, 1997.

[66] O. Etzioni and D.S. Weld, "Intelligent agents on the Internet: Fact, fiction, and forecast", *IEEE Expert*, Vol. 10, no. 4, 1995, pp. 44-49.

[67] S. Franklin, and A. Graesser, "Is It an Agent or just a Program? A Taxonomy for Autonomous agents", In *Proceedings of the Third International Workshop on Agent*

Theories, Architectures, and Languages. New-York: Springler-Verlag, 1996, http://www.msci.memphis.edu/~franklin/AgentProg.html

[68] P. Desharnais, *Agent Assisted Price Negotiation for Electronic Commerce,* M. Comp. Sc. thesis, Dept. Computer Science, Concordia University, Montreal, 2000.

[69] R. Cunningham, *The organic Consumer Profile*, Alberta, Agriculture, Food and Rural Department, April 2001, http://www.agric.gov.ab.ca/food/organic/ organic_profile.pdf

[70] M. Fleming and R. Cohen, "User Modeling in the Design of Interactive Interface Agents", *Proc. 7th International Conference on User Modeling*, pages 67-76, http://ai.uwaterloo.ca/~mwflemin/pub.html

[71] R. Lecoeuche, C. Mellish, and D.S. Robertson, "A Framework for Requirements Elicitation through Mixed-Inititiative Dialogue", *Proc. 3 rd International Conference on Requirements Engineerring, ICRE 1998,* April 6-10, 1998, Colorado Springs, CO, USA, pp. 190-197.

[72] M. Tsai, P. Reiher, and J. Popek, "Baby Steps from GUI towards Dialogue: Mixed-Initiative Computerese", *AAAI'99 Workshop on Mixed Initiative Intelligence*, July 1999, http://lasr.cs.ucla.edu/ficus-members/tsai/AAAI.mixed.initiative.doc

[73] S.A. Wolfman et al., "Mixed Initiative Interfaces for Learning Tasks : SMARTedit Talks Back ", *International Conference on Intelligent User Interfaces (IUI'01)*, January 14-17, 2001, Santa Fe, New Mexico, 2001.

[74] G. Tecuci et al., "Mixed-Initiative Development of Knowledge Bases", *Proc. AAAI-1999 Workshop on Mixed-Initiative Intelligence*, July 18-19, Orlando, Florida, AAAI Press, Menlo Park, CA., 1999, pp. 51-58.

[75] R.G. Eggleston, "Mixed-Initiative Transactions: A Cognitive Engineering Approach to Interface Agent Modeling", *Proc. 1999 AAAI-99 Workshop on Mixed-Initiative Intelligence,* July 18-19, Orlando, Florida, AAAI Press, Menlo Park, CA., 1999, pp. 35-39.

[76] D.G. Novick and S. Sutton., "What is mixed-initiative interaction?", *Proc. AAAI Spring Silmposium on Computational Models for Mixed Initiative In tercation*, Palo Alto, CA., April 1997, pp. 114-116.

[77] M. Sarini and C. Strapparava, Building a User Model for a Museum Exploration and Information-Providing Adaptive System, *Proc. 2nd Workshop on Adaptive Hypertext and hypermedia, held in conjunction with The 9thACM Conference on Hypertext and Hypermedia, 1998, Computer Science Note, 98/12.*

[78] A. Kobsa and W. Wahlster, "Preface; The History of this Volume", *Computational Linguistics,* vol. 14, no. 3, September 1988,

[79] R. Abi-Aad, *A New User Model to Support Electronic Commerce*, M. Comp. Sc. thesis, Dept. Computer Science, Concordia University, Montreal, 2001.

[80] E. Rich, "User Modeling via Stereotypes", *Cognitive Science*, vol. 3, no. 4, 1979, pp. 329-354.

[81] V. Williams, E-commerce Small Businesses Venture Online, Office of Advocacy U.S. Small Business Administration, July 1999, http://www.sba.gov/ADVO/stats/e_comm.pdf

[82] M. Pastore Small Business Embraces Net, Shuns E-Commerce", *Internet.com, Small Business,* 2001, http://cyberatlas.internet.com/markets/smallbiz/article/0,,10098_860861,00.html

[83] United Nations, *United Nations Standard Product and Services Code,* http://www.eccma.org/unspsc/browse/

[84] U.S.Census Bureau, "NAICS, the Official Statistics", *1997 Economic Census,* 1999, http://www.census.gov/prod/ec97/97numlist/311.pdf

[85] Statistics Canada, http://www.statscan.ca/english/

[86] P. Buono, "Integrating User Data and Collaborative Filtering in a Web Recommendation System", *Proc. 3rd Workshop on Adaptive Hypertext and Hypermedia, Hypertext'01,* Arhus, Denmark, August 2001, pp. 135-146, http://wwwis.win.tue.nl/ah2001/papers/costabile.pdf.

Appendix A

MAIN COMPONENTS AND FUNCTIONALITIES OF THE PROOF-OF-CONCEPT PROTOTYPE

The prototype includes three major components:

- The Interface
- The Database
- The Agents

The database includes three tables:

i. *UserInfo table:* responsible for storing the user models.

ii. *Products table:* responsible for storing information about the products.

iii. *Taxonomy table*: responsible for storing the product hierarchy.

The design of the prototype is object oriented and there are eleven main objects:

i. *Home page*: this page presents the user with three options: to log in into the system, to register or to browse.

ii. *Registration form:* allows the first-time user to register in order to take advantage of the personalized service.

iii. *Processing agent:* performs analysis of the registration results and forms the initial profile based on data supplied in registration form and general knowledge about wellness consumers.

iv. *Logged-in*: to welcome repeat customer with appropriate suggestions.

v. *Browser:* an interface for browsing the product hierarchy.

vi. *Search:* an interface for displaying search results.

vii. *Customizing agent:* personalizes interfaces based on the user profile (personalized advice, useful links list, suggestions).

viii. *Check-out form:* to allow the user to buy selected products.

ix. *Profile-updating agent*: keeps track of user interactions with the system and at log-out updates the profile and if necessary generates validation questions.

x. *Validation question:* to ask the user for clarification in case of inconsistent behavior.

xi. *Logout greeting:* this screen appears when the user has successfully logged out.

The choice of the implementation languages was dictated by speed of prototyping and requirements of the web-hoster. The interface was coded in HTML, the database was implemented in MS Access 2000, and agents and interfaces between components – in ASP and VBScript[30].

The application provides users with the following functionalities: browsing, search, shopping cart and purchasing, receiving advice (customizable) and browsing related links (customizable).

[30] There exists also a CLIPS version of agents knowledge base but there is no standard interface between CLIPS and other programs. Using JESS would not solve the problem because while JESS is easily connected to Java code, finding a provider who would support Java and/or JavaScript is difficult and expensive.

When the user arrives at the store web site he/she is welcomed with a complete set of options. The user can log-in by supplying the required information and clicking "Log in" button, or he/she can register: button "Register" will take him/her to the corresponding screen; or he/she can choose to browse without registration. Figure A.1 shows the Home page GUI.

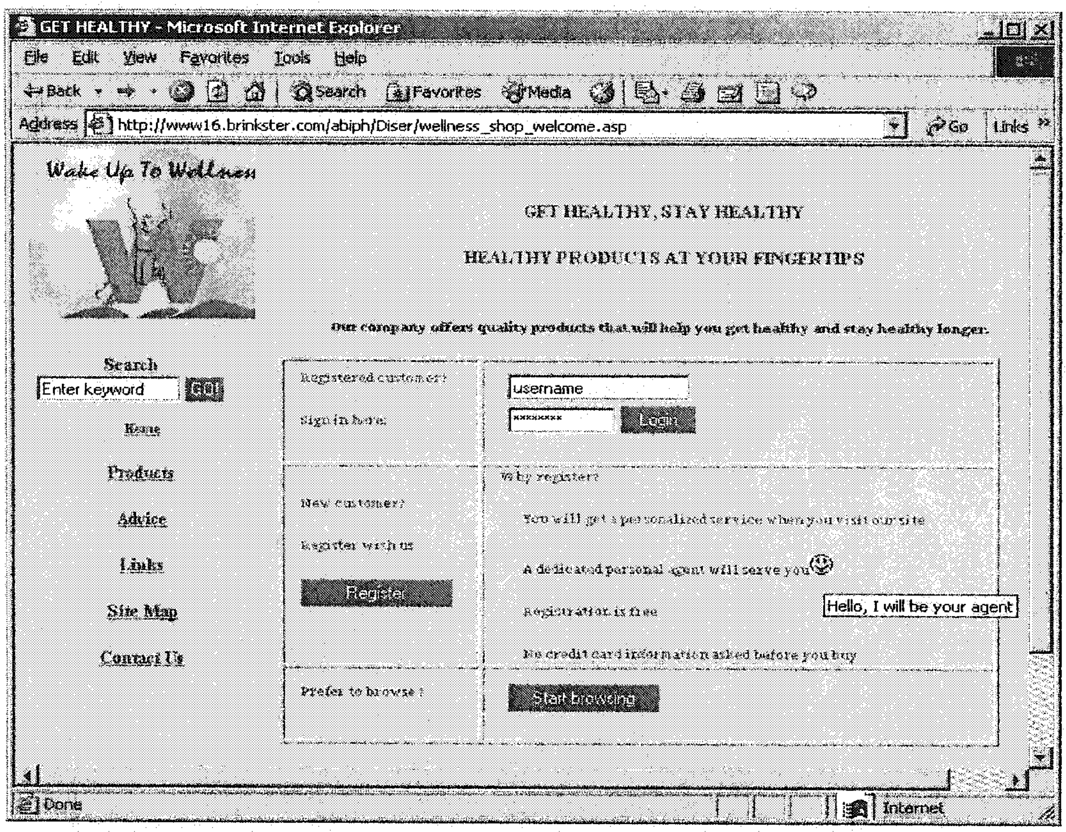

Figure A.1. Home page

From Home page the user can choose one of the several options: log-in, register, browse or search, as well as explore the links to Advice and Links pages. All users can browse or search product selection without registration or log-in. Figure A.2 shows a window to display search results. Browsing results are presented in similar format.

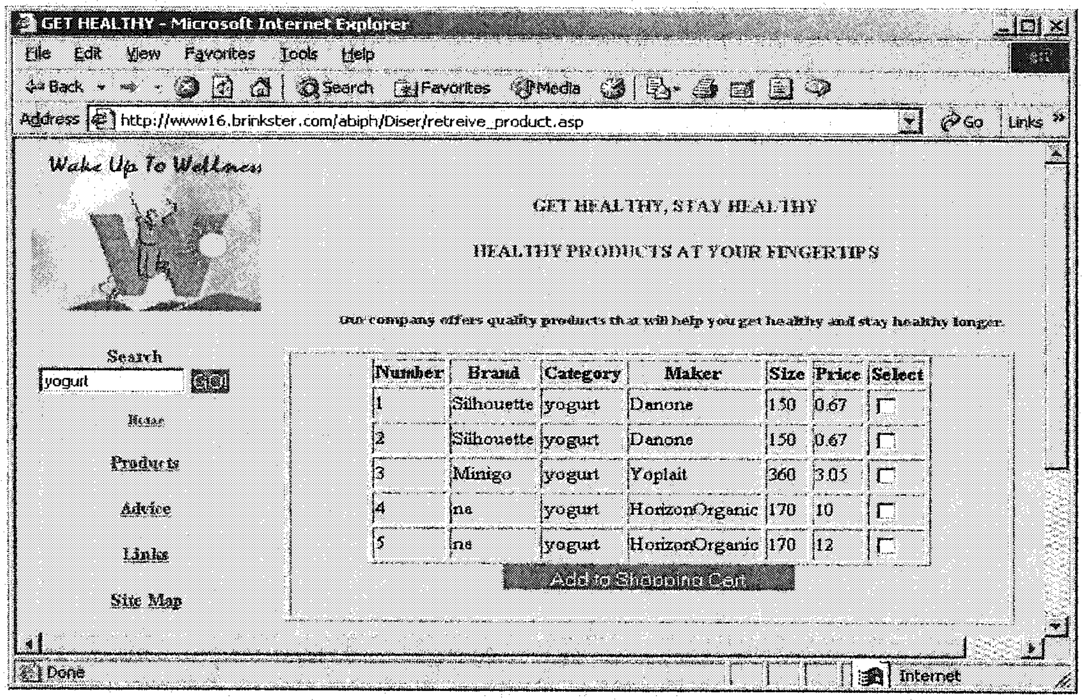

Figure A.2. Search Results Window

The information about the current session is recorded in the Shopping Cart array that contains username and a list of selected products. At check out if the user is known to the system (through registration or log-in) he/she can check-out products currently in his/her shopping cart (Figure A.3). Otherwise he/she will be asked to log-in or register. After the purchase information has been processed necessary updates are made to the user profile to reflect current product choices.

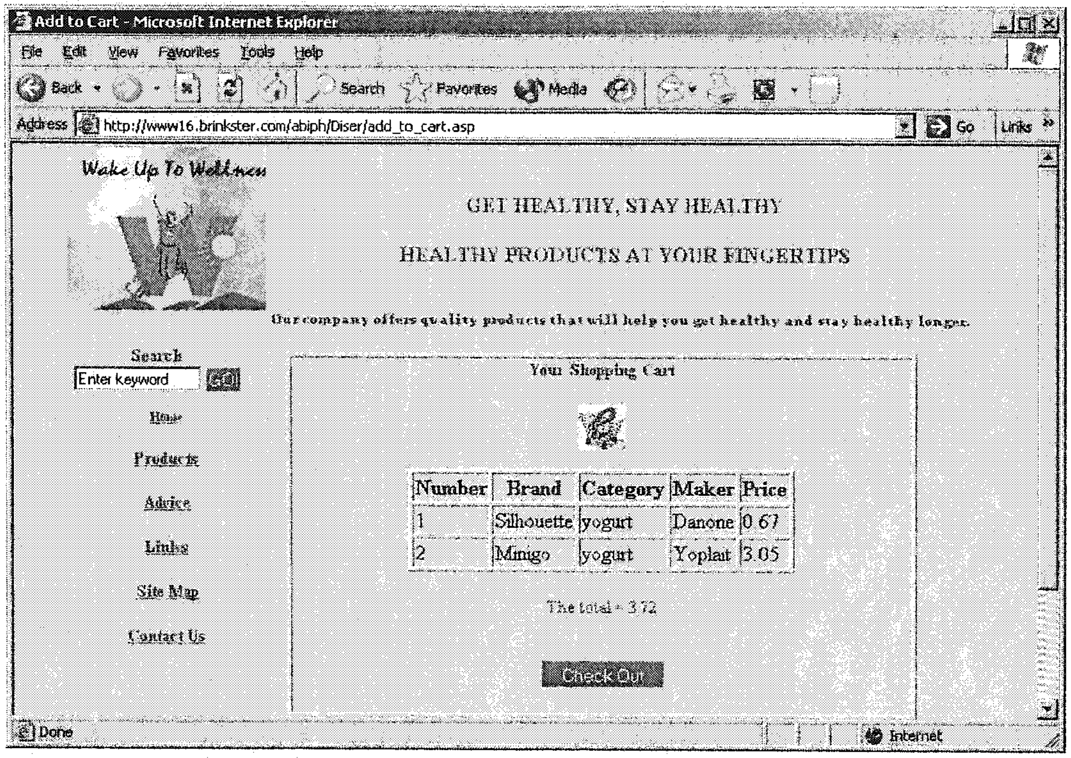

Figure A.3. Checkout Page