Tailoring the Interaction with Users in Web Stores

Liliana Ardissono and Anna Goy
Dipartimento di Informatica, Università di Torino
Corso Svizzera 185; 10149 Torino, Italy
E-mail: {liliana, goy}@di.unito.it

Abstract. We describe the user modeling and personalization techniques adopted in SETA, a prototype toolkit for the construction of adaptive Web stores which customize the interaction with users. The Web stores created using SETA suggest the items best fitting the customers’ needs and adapt the layout and the description of the store catalog to their preferences and expertise.

SETA uses stereotypical information to handle the user models and applies personalization rules to dynamically generate the hypertextual pages presenting products. The system adapts the graphical aspect, length and terminology used in the descriptions to parameters like the user’s receptivity, expertise and interests. Moreover, it maintains a model associated with each person the goods are selected for; in this way, multiple criteria can be applied for tailoring the selection of items to the preferences of their beneficiaries.

Keywords: user modeling, personalized information presentation, customization of Web stores, adaptive hypermedia, knowledge-based approaches to the personalization of the interaction, electronic shopping.

1. Introduction

The popularity of Web shopping is increasing and now heterogeneous (home and business) customers purchase goods by accessing Web catalogs without relying on intermediaries. Noticeably, this trend does not only concern traditional domains like air-flight booking, or movies and music, but also other domains, ranging from the bank services to the electronic power trade (Ygge, 1999). In all these cases, the user needs information about the alternative products/services and their advantages and disadvantages, in order to compare them and identify those best satisfying her needs and constraints.

As discussed in (Benyon, 1993), the users of software systems differ in many characteristics, such as their status, expertise, preferences and even the reason for using the systems; therefore, to enhance the usability and applicability of such systems, it is extremely important that these factors are taken into account.

In several commercial tools for the creation of Web stores, like Microsoft Merchant, IBM’s Net Commerce, ATG’s Dynamo, and Broad-
Vision, the personalization of the interaction with the customer has been enhanced to establish a long-term relation with her (in contrast to the typical one-shot interactions supported by the first electronic commerce systems). These tools offer functionalities for the identification of the items best matching the user's preferences and they tailor the suggestion of goods to the individual user. However, they adopt quite simply, if any, techniques to personalize the presentation of the selected items; therefore, they typically present the Web catalogs in the same way to all users.

We believe that all of the following issues are central to the enhancement of the business-to-customer relationship:

- Different needs and preferences in the selection of items should be satisfied. This ability requires filtering capabilities, to identify the items most suited to the specific customer; e.g., see (Popp and Lödel, 1996; Karunanithi and Alspector, 1997; Raskutti et al., 1997; Cotter and Smyth, 2000).

- The advertisements included in the catalog pages should be targeted to the customer base and selected on the basis of the interests and lifestyle of the individual user accessing the system; e.g., see (AIMedia).

- The users' preferences for the type of information provided and interaction style should be satisfied; e.g., see (Boyle and Encarnacion, 1994; Milosavljevic and Oberlander, 1998; Cheverest et al., 2000).

- The customers' software and hardware requirements, together with their connection to the Web should be taken into account to facilitate their access and navigation in the Web stores. For instance, the user should be allowed to select alternative media for the delivery of the information, depending on the efficiency of her connection to the Web (Joerding, 1998); moreover, alternative media could be exploited to deliver information to people with special needs (Fink et al., 1998).

The issue of tailoring information to the user has been deeply analyzed in the adaptive hypermedia research, where a major distinction was made between personalizing the navigation task and personalizing the descriptions (Brusilovsky, 1996). Some researchers, like (Calvi, 1997), have focused on the dynamic adaptation of the hypertextual structure to users with different backgrounds. Others, like (Paris, 1988), (Milosavljevic et al., 1996), (Hirst et al., 1997), (De Carolis, 1998),
(Wilkinson et al., 2000) and (Bental et al., 2000) have focused on the
dynamic generation of texts tailored to the user.

An analysis of electronic sales reveals other issues to be addressed
in a Web store: e.g., the user should be assisted while browsing the
catalog and selecting items to purchase; the system should keep track
of her actions; it should remember which items she has analyzed, and
other data useful to identify her real needs. Moreover, in addition to
suggesting the products most suitable for the customer, their descrip-
tions should explicitly carry relevant information for the comparison
of items, based on the user’s individual evaluation criteria. Finally,
to enhance the system’s initiative, the properties having the greatest
impact on the customer might be highlighted to convince her to buy
the items (Jameson et al., 1995).

In some systems, very general techniques, like those exploited in the
information filtering research (e.g., collaborative filtering, or TF/IDF -
term frequency/inverted document frequency) have been used to select
interesting items in environments where heterogeneous information
sources are exploited, or little information is available about the user’s
needs; for instance, see (Ackerman et al., 1997; Balabanović, 1998;
Billsus and Pazzani, 1999; Burke et al., 1997; Resnick and Varian, 1997;
Fink and Kobsa, 2000). While those techniques are suited to dealing
with large-scale applications, such as information retrieval on the Web,
other work has shown that more specific techniques can be applied to
personalize the interaction with the customer in electronic commerce
systems, thanks to the fact that the domain-specific information is at
least partially structured. For example, (Raskutti et al., 1997) describe
a recommender system for VOD applications, where the structure of
a movie database is exploited to customize the suggestion of items.
Their system analyzes the customer’s selections, identifying the prod-
uct attributes which affect her decisions; this information is then used
to customize the selection of new items to be suggested. However, the
authors don’t include in the system any knowledge about the meaning
of the attributes of movies, or the relations existing among them; thus,
their approach does not support effective strategies for personalizing
the description of items.

In contrast, (Dale et al., 1998) emphasize the selection of the infor-
mation to be presented. Their system generates labels for museum items
by summarizing the information stored in the records of the database.
Since most of this information consists of unstructured Natural Lan-
guage text, the system exploits NLP techniques to interpret the text
and generate summaries. This deep analysis of content is the basis
for the generation of personalized labels, when the user’s interests are
known; however, it requires the application of complex NLP techniques
and the exploitation of a detailed domain ontology, which might be problematic if the system had to be instantiated in other domains, where very different types of objects have to be presented.

2. Our work

In this paper, we describe the user modeling and personalization techniques adopted in SETA, a prototype toolkit for creating adaptive Web stores which tailor the interaction to the features of their customers, possibly suggesting the items fitting their preferences in the best way (Ardissono and Goy, 1999).

An on-line demo of a prototype store created using SETA is available at the URL: http://www.di.unito.it/~seta. This store presents telecommunication products, like phones and switchboards, and will be used throughout the rest of the paper as a concrete example to describe the functionalities of our system.

SETA supports several personalization strategies, concerning both the suggestion of items and the presentation style adopted during the interaction with the customer. The system evaluates how closely the items match the user preferences and suggests the best ones. As far as the presentation style is concerned, we have mainly focused on the selection of the relevant information about products and the customization of their descriptions. Moreover, we have improved the flexibility of the Web store interface by introducing an initial personalization of the layout of the catalog pages: this personalization concerns some basic parameters such as the background colors and the font face and size used in the pages.

Although the system presented in this paper offers a subset of the personalization functionalities listed in section 1, it is flexible and can be suitably extended: in fact, we have exploited knowledge representation techniques supporting a declarative description of the strategies exploited by the system, as well as of the domain-dependent knowledge. During the development of SETA, this approach has enabled us to extend it and enhance its configurability to other sales domains.

The paper is structured as follows: section 3 sketches the system architecture and section 4 describes in detail the user modeling techniques adopted in SETA. Section 5 presents the strategies used for the suggestion of items. Section 6 describes the personalization strategies used to choose the layout and content of the catalog pages. Section 7 reports some details about experiments we have made to test the system and section 8 closes the paper.
3. The SETA architecture

The customization of on-line stores is a complex activity and requires very different types of expertise, such as knowledge about users and products, and techniques to personalize the layout and content of the catalog pages; therefore, complex software architectures are needed. The exploitation of agent-based technologies helps managing this complexity in an effective way. In fact, multiagent systems can be designed, where several agents offer specialized services (possibly exploiting heterogeneous technologies) and interact with each other to produce the overall, complex service to the user; see (Sycara et al., 1996; Jennings et al., 1998).

The architecture of SETA includes a set of specialized agents, devoted to the activities carried on in the front-end of an adaptive Web store. When we designed the system, we identified the basic roles to be filled for supporting a personalized interaction with customers and we associated one agent to each role, therefore obtaining the architecture shown in Figure 1; the figure shows the complete system architecture in a situation where three users - browsers - are visiting the store in parallel. In the following, we shortly describe the agents, while a thorough description of the architecture can be found in (Ardissono et al., 1999a; Ardissono et al., 1999c).

- The Session Manager handles the communication with the browsers: each time a user connects to the store, or performs actions in the Web catalog, the Session Manager catches her actions and forwards them to the Dialog Manager.\footnote{The Session Manager is a Servlet and can identify the possible actions (e.g., following an hypertextual link, clicking on a button, etc.) by catching HTTP requests.} Moreover, the Session Manager forwards the Web store pages to the user’s browser.
– The Dialog Manager maintains the dialog context and handles the logical interaction with customers. This agent decides which page has to be shown at the next step, on the basis of the overall dialog context and of the last action performed by the user, who may follow a hypertextual link, click on a button, or other.

– The User Modeling Component (UMC) initializes and updates the user models during the interaction.

– The Product Extractor selects the product items and ranks them to suggest the best ones.

– The Personalization Agent dynamically generates the customized pages of the Web store.

– The Shopping Cart Manager keeps track of the items selected by the user during an interaction: at any time, items can be put into the cart, or removed, and the cart always displays the updated content and the total amount of money to pay. The shopping cart groups items on the basis of the (possibly indirect) user to which they are directed.

– The Products and Users DB Managers handle the Products and Users databases, respectively. The Products DB contains the information about the goods available in the store, specifying, for each item, its price, technical features, and so forth. The Users DB contains the records of the customers who have visited the store, including the goods purchased in their previous visits. Customers are explicitly required to specify whether they authorize the system to maintain their data for the subsequent visits. So, the Users DB only contains information about people who agreed to the long term storage of their personal data.

The Web store supports multi-user access; thus, in addition to a possible parallel execution of tasks during the interaction with a single user, each agent handles the session-dependent data for all the active user sessions in parallel: Figure 1 shows the active contexts of the UMC, representing them in the shapes denoted by the “UM-Session” label; the other user-dependent sessions are hidden for simplicity.

The agents retrieve the domain-dependent knowledge from declarative knowledge bases. For instance, the customer classes relevant in the sales domain are defined in a Stereotype KB, exploited by the UMC to handle the user models. Moreover, a Product Taxonomy stores the...
description of the products and specifies the relations among them. For example, similar products are associated with each other, and basic products are related with products offering the same functionalities in a single, integrated solution.

A new Web store can be created by configuring the knowledge bases containing the information dependent on the particular sales domain. The knowledge bases of the Web store are defined by using graphical interfaces which guide the designer in the definition of the data.

SETA is a multiagent architecture and the system is implemented in a Java-based environment, using Voyager (ObjectSpace) to distribute its agents and enable them to communicate with each other by means of synchronous, asynchronous and multicast messages. The Web stores are connected to the Internet by means of the Apache Web Server. In the following, we will focus on the activities of the three agents responsible for the personalization of the interaction, i.e., the UMC, the Product Extractor and the Personalization Agent.

4. Management of the knowledge about users

4.1. The User Modeling Component (UMC)

The User Modeling Component (UMC) creates and maintains the user models of the customers accessing the Web store.

Two cases can be distinguished. If the customer visits the Web store for the first time, no information about her is available and her preferences and characteristics have to be recognized during the interaction. On the other hand, if the customer has already registered her data during a previous visit, a precise description is generally available in the Users DB and can be used to initialize the user model and start the interaction with detailed information about her. Customers can visit the store without registering their data in a permanent way: in that case, the system creates and maintains the user models only during the interaction; therefore, the users will be considered as first-time customers at their subsequent visits. On the other hand, if they register, the system stores their data and purchases in the Users DB.

One peculiarity of Web shopping is that, to correctly personalize the interactions, the system must deal both with direct and indirect users. In fact, a customer may choose items for herself, as well as for other people (indirect users), who may have different preferences and requirements for products. For instance, imagine a young student who is looking for a present for his grandmother: the system should personalize the interaction style to the direct user, tailoring the selection of the
information about goods and the quality of their descriptions to the young student. On the other hand, effective help in the selection of the gift would only come if the system customized the suggestion of such goods to the preferences and needs of the intended beneficiary, i.e., his grandmother. Of course, if the student is looking for goods for himself, his own model should be used both to personalize the interaction style and the suggestion of goods.

The system gets from the user the information needed to adopt the appropriate personalization strategies in an interactive way: before starting the navigation of the catalog, a dynamic page is displayed to let the user specify which products she would like to inspect and who is the beneficiary of the goods. This selection is performed on the high-level product categories available in the store: these categories correspond to the roots of the Product Taxonomy and, in our prototype, they are phones, answering machines, faxes, switchboards and multiline phones. The user’s selection does not represent any commitment to purchase specific goods and its only purpose is to let the user express her needs in a compact way (for instance, she might be looking for a phone, but she might end up purchasing a fax-phone with answering machine). On the basis of this information, the system can guide the user in the portion of the catalog with the relevant goods.

Figure 2 shows the page for the initial selection of the products which the user wants to inspect: the top of the page provides an introductory prompt and explains how the user can select the product categories to be inspected during the navigation of the catalog (a “More info” button is available to see a demonstration of the way products can be selected). The central area of the page shows the main product categories and the possible destination uses for the desired products: home use, office use, or unspecified use. In the lower part of the page, the user also finds a button (gift box) for specifying whether she is looking for goods directed to somebody else; if she clicks on that button, a very similar page is displayed, where she can choose the desired product(s) and specify who is the beneficiary. The beneficiary is identified by nickname only to protect the identity of the indirect user.

The user can select the products by dragging the related icons and dropping them into the appropriate destination; multiple selections can be performed; e.g., the user might select a phone for her home, a fax and

---

2 The information about the destination use is interesting because it might determine different preferences for product properties. For instance, a home use might impose strict constraints on the cost of products; instead, an office use might influence other properties, such as their technology level. The unspecified use is especially relevant if the user is selecting gifts for other people.

3 The nicknames are used to tag the navigation contexts concerning the selection of products for the other beneficiaries.
a phone for her office, and so forth. The selection of multiple products starts parallel navigation contexts in the Web store, where the user can inspect the various product categories and choose items to purchase. The system separately keeps track of each navigation flow, in order to monitor each navigation as a separate behavior. As we will describe in section 6, the catalog pages contain a section where the active interaction paths are summarized to the user and links are offered to switch among them. Thus, the user is allowed to carry on multiple navigations in an interleaved way.

4.2. Management of the User Models

The UMC creates a direct user model to tailor the product presentation and the selection of goods to the needs of the customer visiting the Web store. Moreover, it creates a user model for each third party for whom any goods are selected: each beneficiary model is used to suggest the goods suited to the person they are intended for, if different from the direct user. The presentation style is always tailored to the direct user. The models of the indirect users contain a subset of the information stored for the direct user and omit the user features influencing the
presentation style. Since the rest of the information is the same in both types of models, we will describe the user models for the direct users.

The user model of a customer who registered her data in the Users DB is initialized by loading her record from the Users DB. On the other hand, the model of a first-time customer is created and initialized with the preferences of a very generic user of the system. Then, the model is revised as soon as the first information about her is collected.

Although the user models can be refined by monitoring the customer's behavior and applying user modeling acquisition rules, this technique is effective only after a while, when enough information about her behavior is available. We exploit stereotypical information about customers (Rich, 1989; Kobsa, 1990; Kay, 1994) to speed up the revision process: the UMC compares the initial user data with a set of stereotypes, describing the characteristics of the main customer classes identified for the sales domain. The comparison is made by exploiting a set of personal data (e.g., age, job and education level), previously requested from first-time users in a registration form. The results of the comparison are exploited to predict the user's preferences for the interaction style and the product properties; we will describe the use of stereotypical knowledge in sections 4.6 and 4.7.

4.3. STRUCTURE OF THE USER MODELS

The characteristics relevant to the personalization of the interaction may vary from one sales domain to the other and can be identified by performing a market analysis. Thus, we designed the UMC in such a way that the user models can be configured for each Web store instance. The user models are composed of a fixed part, containing a set of domain-independent attributes, and a configurable part, containing the domain-dependent attributes defined for the specific Web store.

The fixed part of the user models includes information about the user characteristics relevant to the management of the store interface; for instance, the user's receptivity (i.e., her capability of absorbing large amounts of information) and her requirements concerning the layout of the Web pages; e.g., we consider a "sight" feature, which describes the user's capability to read small texts.

The fixed part also includes high-level information, such as the user's technical and aesthetic interest, and her expertise degree: this is an evaluation of the user's familiarity with the products of the sales domain.

---

4 Filling in the registration form does not mean that the information is stored in the Users DB: each datum is just maintained in the system's working memory. The user is asked whether she wants to permanently store her data only when she closes the interaction.
These concepts provide an abstract description of the user, on which domain-independent personalization rules are based. This approach simplifies the configuration of new Web stores and supports the use of general personalization rules that customize the presentation style. However, it limits the representation of the user’s expertise and interests in domains where very different products are sold: for instance, as far as telecommunication products are concerned, a user might have good expertise on phones, but she might not be familiar with other goods, such as switchboards.  

The configurable part of the user model includes the description of the user’s preferences for product properties. Figure 3 sketches an example of a user model instantiated for Paul Smith; the user model contains five main types of information:

- Identification data: in our prototype, the user’s first and family name.
- Personal data of the user; e.g., age and job.
- Domain-independent information, such as user characteristics (e.g., receptivity) and interests, used to customize the interaction style. From now on, for brevity, we will group these types of information in a single set, called “User features”.
- Domain-dependent preferences for product properties, used by the system to decide which items are most suited to the user.
- Information about the classification of the user in the stereotypical customer classes. The use of this information will be explained later on, in section 6.4.2.

While each identification and personal datum is a simple <feature, value> pair, the user features and preferences are represented as complex parameters with the following slots; see (Torasso and Console, 1989):

- The “Values” slot contains a list of <linguistic value, likelihood> pairs, which can be interpreted as a probability distribution for the linguistic values of the parameter. The user features are characterized by the “low”, “medium” and “high” linguistic values. The user

---

5 To overcome such limitations, but maintain a simple management of the user model, we have recently modified this representation to explicitly describe the user’s expertise and interests for the high-level product categories available in the store. In this new approach, the number of concepts included in the user model corresponds to the number of high-level product categories in the Product Taxonomy; i.e., phones, faxes, answering machines, switchboards and multi-line phones.
Identification:

First Name: Paul;
Family Name: Smith;

Personal data:

Age: 55-64;
Gender: male;
Job: employee;
Education Level: high school;

User features:

Characteristics:

Receptivity:
Values: low: 0.5; medium: 0.25; high: 0.25;
Sight:
Values: low: 0.3; medium: 0.4; high: 0.3;

Knowledge:

Expertise:
Values: low: 0.5; medium: 0.3; high: 0.2;

Interests:

Technical Interest:
Values: low: 0.1; medium: 0.5; high: 0.3;
Aesthetic Interest:
Values: low: 0.3; medium: 0.3; high: 0.4;

Preferences:

Quality:
Importance: 0.8;
Values: low: 0; medium: 0.4; high: 0.6;
Ease of Use:
Importance: 1;
Values: low: 0; medium: 0.25; high: 0.75;
Cost:
Importance: 0.7;
Values: low: 0.5; medium: 0.3; high: 0.2;
Design:
Importance: 1;
Values: low: 0; medium: 0.8; high: 0.2;

User classification:

Life style:
Values: yuppie: 0.2; average: 0.6; modest: 0.2;
Domain expertise:
Values: ...

Figure 3. An example user model for the telecommunication domain.

preferences are defined by the store designer and can, in principle, have a different set of linguistic values. For uniformity, in our prototype, we have decided to use the same set of linguistic values.
The number associated with a linguistic value $lv$ of a user feature $F$ denotes the system’s estimate of the likelihood that $F$ is characterized by the value $lv$; for instance, as shown in Figure 3, the likelihood that Paul's domain expertise is low, medium or high is, respectively, equal to 0.5, 0.3, or 0.2.\footnote{These numbers sum to 1, in order to be interpreted as a probability distribution for the linguistic values of the related parameter.}

The number associated with a linguistic value $lv$ of a preference $P$ denotes the system’s estimate of the likelihood that the user prefers products characterized by the value $lv$ for the property $P$. For instance, the likelihood that Paul prefers very easy to use products is 0.75: see the “high” value of the related preference.

- The preferences are also characterized by an “Importance” slot, describing the system’s estimate of the relevance of the preference to the user for the evaluation of goods. The importance ranges in [0,1], where “0” means that the preference is totally irrelevant, while “1” means that it is extremely important. For instance, in Paul’s user model, the quality of products and their ease of use are, respectively, very important and extremely important properties.

4.4. STEREOTYPICAL KNOWLEDGE

The UMC retrieves the domain-dependent knowledge about users from a Stereotype Knowledge Base (KB), which contains a hierarchy of stereotypes clustering the properties of homogeneous customer (user) groups. Following the approach adopted in several socio-demographic studies, such as (Eurisko) for the Italian population, we assume that a correlation can be identified among a set of user data, typically including psychographic and socio-demographic data, and a set of user preferences. Therefore, we structure the stereotypes in two main parts: a profile, composed of a set of classification data characterizing the individuals belonging to the represented user group, and a prediction part, describing the user features and the preferences typical of such individuals. The classification data are used to evaluate how closely the individual customer visiting the Web store matches a stereotypical description; the predictions are used as the basic data to fill in the prediction part of the user models.

In principle, the user population may be partitioned according to different viewpoints: for instance, customer groups characterized by similar preferences for properties like products quality and cost could be identified. Moreover, the customers' life style might be modeled: in
this case, the population might be partitioned by taking into account preferences for properties like products design, brand, and so forth. As a last example, another classification of users might be considered, to model the extent to which the users are familiar with the product categories available in the store.

The Stereotype KB may include multiple segmentations of the population, which we call stereotype families. A family is a partition of the population and includes a set of stereotypes, each one representing a class of users with similar characteristics. Each family has its own set of classification and prediction data, occurring, respectively, in the profile and the prediction part of the stereotypes belonging to the family. Different families may have overlapping profiles; in fact, some psychographic and socio-demographic data may be relevant to more than one segmentation of the population. For our prototype, we have defined four stereotype families concerning:

- The users’ domain expertise: we distinguish between expert customers of the telecommunication domain and novices.

- Their life style: this family segments the customer population on the basis of socio-demographic data supporting predictions of interests, tastes, and so forth.

- Their graphical interface requirements: this family describes user preferences for graphical properties of the store interface, like colors of the backgrounds and readability of the text in the Web pages.

- The destination use of the goods: we distinguish home and business use, involving different preferences for product properties. This family is very different from the others; in fact, instead of describing a set of context-independent characteristics, it partitions the customer population on the basis of the reason they purchase goods. Stereotypical descriptions which make explicit reference to the users’ underlying goals have been used in other related work; for instance, see (Vassileva, 1996).

As shown in Figure 3, the classification data used in our example domain are the users’ age, gender, job and education level. The predictions concern the user features (receptivity, sight, domain expertise, technical interest, aesthetic interest, etc.) and preferences for product properties such as quality, ease of use, portability, technicality, cost, novelty and design.

\footnote{The cost of items does not correspond to their real price: in fact, it concerns the evaluation of that feature; e.g., being economical, or expensive.}
NOVICE USER

Profile:

Age:
Values: less than 24: 0.1; 25-34: 0.05; 35-44: 0.05; 45-54: 0.15; 55-64: 0.3; more than 65: 0.35.

Job:
Values: employee: 0.3; retired: 0.3; student: 0.1; teacher: 0.05; ...

Prediction part:

User features:

Domain Expertise:
Values: low: 0.8; medium: 0.15; high: 0.05;

Technical Interest:
Values: low: 0.5; medium: 0.4; high: 0.1;

Preferences:

Ease of Use:
Importance: 1;
Values: low: 0; medium: 0.3; high: 0.7;

...

Figure 4. A portion of the “Novice” stereotype for the telecommunication domain.

4.5. STRUCTURE OF THE STEREOTYPES

In the following, we will describe the structure of the stereotypes referring to Figure 4, which shows a portion of the “Novice” stereotype, belonging to the “Expertise” family: “Novice” describes the characteristics of non-expert customers of the telecommunication domain.

4.5.1. Profile of a stereotype

The profile of a stereotype S defines the information needed to evaluate how closely a user matches the description of the stereotypical user belonging to S. The profile contains a set of classification data and specifies, for each one, the compatibility of its linguistic values with the description of the user class represented by the stereotype. Given a datum D, the compatibility of a linguistic value lv is a number in [0,1] and corresponds to the percentage of individuals belonging to C for which D takes the lv value. For instance, as far as the “Novice” class is concerned, 10% of novice users are under 24; 5% of them are between 25 and 34, and so forth.

4.5.2. Prediction part

The prediction part describes the features and preferences of a typical user belonging to the customer class described by the stereotype.
Given a prediction $P$, the number associated with a linguistic value $lv$ represents the likelihood of $lv$ for the prediction $P$, given that the user belongs to the class described by the stereotype.

4.6. **Classification of Users**

The classification process determines how closely the user matches a stereotypical description; the result of this process is a matching degree, represented by a number in $[0,1]$, where “1” represents a perfect match between the user and the stereotype, while “0” represents a null match, i.e., the user is clearly different from the typical customer described by the stereotype.

The UMC evaluates how closely the user matches a stereotype $S$ by matching the user’s classification data on the profile of $S$. Each classification datum is separately considered to obtain an individual score. Then, the scores are combined to calculate the final matching degree between the user and $S$.

4.6.1. **Evaluation of the individual scores**

Given a classification datum $D$ and the linguistic value $lv_0$ associated with $D$ in the user model, the individual score of $D$ is set to the compatibility of $lv_0$ in the profile of $S$.

For instance, let’s consider two pieces of data, $A$ and $B$, which might represent the age and job of users. If in the user model they have, respectively, the linguistic values $a_i$ and $b_j$ (e.g., the user is between 25 and 34 and she is a teacher), and the compatibility of such values are $c_{a_i}$ and $c_{b_j}$, then $score_A = c_{a_i}$ and $score_B = c_{b_j}$.

Notice that the linguistic values of the classification data are set in the user model on the basis of the user’s answers to a registration form. In order to respect the user’s privacy, the system does not force her to answer such questions; thus, when evaluating the individual scores, some data may be missing and must be ignored. Clearly, the omission of any classification data makes the evaluation of the matching degree of the stereotype less precise.

4.6.2. **Evaluation of the overall matching degree**

For each family of the Stereotype KB, the UMC evaluates the user’s degree of matching with respect to all the stereotypes of the family and it normalizes the resulting values to make them sum up to “1”.

The user’s degree of matching with a stereotype $S$ is evaluated by combining the individual scores of its classification data. These scores are merged by applying the following formula (Lesmo et al., 1985):

$$Match(score_A, score_B) =$$

\[ (i) \]
<table>
<thead>
<tr>
<th>Age</th>
<th>&lt;24</th>
<th>25-34</th>
<th>35-44</th>
<th>45-54</th>
<th>55-64</th>
<th>&gt;65</th>
</tr>
</thead>
<tbody>
<tr>
<td>scoreemployee</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>scoreage</td>
<td>0.1</td>
<td>0.05</td>
<td>0.05</td>
<td>0.15</td>
<td>0.3</td>
<td>0.35</td>
</tr>
<tr>
<td>Match(scoreemployee, scoreage)</td>
<td>0.0811</td>
<td>0.0448</td>
<td>0.0448</td>
<td>0.1111</td>
<td>0.1764</td>
<td>0.1927</td>
</tr>
<tr>
<td>Normalized match</td>
<td>0.1246</td>
<td>0.0688</td>
<td>0.0688</td>
<td>0.1707</td>
<td>0.2711</td>
<td>0.2960</td>
</tr>
</tbody>
</table>

Figure 5. Degrees of matching for the “Novice” stereotype.

\[ \text{score}_A \times \text{score}_B / (\text{score}_A + \text{score}_B - \text{score}_A \times \text{score}_B) \]

For instance, if \( \text{score}_A = 0.1 \), \( \text{score}_B = 0.3 \), the degree of matching between the user and \( S \), as far as \( A \) and \( B \) are concerned, is 0.0811. Figure 5 summarizes the results of \( \text{Match} \) in the evaluation of the “Novice” stereotype for a set of users, characterized by the same job (e.g., all of them are employees), but having different ages. The individual scores are taken from the definition of “Novice” in Figure 4, but we have repeated them in the table for the reader’s convenience: the first line reports the 0.3 score associated with the “employee” value of the job, while the second line reports the scores associated with the values of the age. The third line shows the matching degrees obtained by applying the \( \text{Match} \) function to the individual scores of job and age; the last line shows the final matching degrees, obtained by normalizing the values computed by \( \text{Match} \) so that they sum up to 1. Notice that, although we have described the evaluation of the matching degree in the case of two classification data, the general case is analogous, because the formula is associative.

Function (i) is a T-norm and evaluates the fuzzy AND of the classification data, by combining their scores in a multiplicative way. Thus, the individual scores can contribute to the overall match in a conjunctive way. Notice however that, although the overall match for a stereotype \( S \) must take into account the whole set of classification data, the matching degree must be null if at least one datum is totally incompatible with the profile of \( S \). This requirement may seem strict, but it derives from the meaning of the compatibility values: in fact, given a datum (e.g., the job) and one of its possible linguistic values (e.g., “employee”), the related compatibility value represents the percentage of people belonging to the stereotype who fit that value; in the above example, 30% of novice users are employees. If the percentage is null, none of the users belonging to the stereotype is characterized by such
<table>
<thead>
<tr>
<th>Stereotype</th>
<th>Importance</th>
<th>low</th>
<th>medium</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td>Novice</td>
<td>1</td>
<td>0</td>
<td>0.3</td>
<td>0.7</td>
</tr>
<tr>
<td>Intermediate</td>
<td>0.8</td>
<td>0.1</td>
<td>0.6</td>
<td>0.3</td>
</tr>
<tr>
<td>Expert</td>
<td>0.5</td>
<td>0.3</td>
<td>0.4</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Figure 6. Three stereotypical predictions of the preference for ease of use.

value; thus, the overall match of the stereotype must be null, regardless how closely the user matches the rest of its profile.\(^8\)

4.7. Prediction of user features and preferences

The UMC fills in the prediction part of the user model (importance of the user preferences, likelihood of the linguistic values of the user features and preferences) by merging the predictions of all the stereotypes of the Stereotype KB. The stereotypes belonging to the same family make predictions of the same set of data; those belonging to different families make predictions concerning disjunct sets of data. We have adopted this approach to simplify the predictions: in fact, the system can fill in the various parts of the user model in an independent way.

The predictions of the stereotypes belonging to a single family are merged, assuming their additive independence, by evaluating the weighted sum of the values suggested by each stereotype, where the weights are the user's degree of matching with the stereotype itself. The idea of predicting information about a user on the basis of how similar the user is to classes of users is not new; for instance, see the exploitation of user communities in (Orwant, 1995). The linear combination of the stereotypical predictions applied in our system has been exploited in other user modeling systems, such as those described in (Linden et al., 1997) and (Coberry et al., 1999).

For instance, consider the “Domain Expertise” family and the preference for products ease of use. Figure 6 reports the information about the prediction of each stereotype on the user preference. If the classification process has produced the following matching degrees:

- **Novice**: 0.7;
- **Intermediate**: 0.2;
- **Expert**: 0.1;

then the predicted the importance of the preference is:

\(^8\) Null compatibility values make a datum play the role of strict negative triggers for the stereotype.
Importance: 0.7 * 1 + 0.2 * 0.8 + 0.1 * 0.5 = 0.91.

Moreover, the prediction on the “low” value is:

low: 0.7 * 0 + 0.2 * 0.1 + 0.1 * 0.3 = 0.05.

The predictions are evaluated for each linguistic value of the datum in the user model; then, the resulting numbers are normalized, so that they sum up to 1 and can be interpreted as an approximated probability distribution for the datum.

4.8. Comments

Some comments are needed on the evaluation of the stereotypes, which we base on a fuzzy AND, instead of using other techniques, such as a probabilistic approach (Neapolitan, 1994), or multiattribute decision analysis (Keeney and Raiffa, 1976).

4.8.1. Comparison with a probabilistic approach

Our frame-based representation of customer classes means that the likelihood that the user belongs to a class is evaluated as a match between the user’s data and the profile of the stereotype. This match does not correspond to a probabilistic evaluation of the stereotype; in fact, the profile specifies, for each classification datum, the compatibility of its linguistic values with the stereotype and this compatibility represents the percentage of people belonging to the stereotype who fit the specific value for the datum (e.g., the percentage of novice users who work as employees), not the conditional probability of the stereotype, given the specific value for the datum.

Although the probability-based approaches have clear advantages in the representation and management of uncertain information, we have decided to adopt a frame-based approach because it fits well the requirements of the store designer, who has the responsibility of defining the Stereotype KB.

The Stereotype KB of a Web store can be defined using a graphical configuration tool, which enables the designer to introduce the domain-dependent knowledge without writing any Java code. By using the tool, the designer can:

- Define the personal data and user preferences (together with their possible values) which describe the users of the individual Web store. The defined information is processed by the tool, which generates a domain-dependent user model template. This template is then used within the UMC for the creation of the user models; see (Ardissiono et al., 1999c).
- Define the Stereotype KB, by specifying its stereotypes and stereotype families.

In particular, our frame-based approach supports an easy elicitation of the domain-dependent information from the experts. During the development of the telecommunication Web store, we could only exploit generic information about customers. However, the experience in other related projects indicates that the domain experts feel comfortable in describing customer classes by specifying the type of information required to define our stereotypes: i.e., the relevant data and, for each datum, an approximated percentage of people fitting its linguistic values (e.g., more or less 30% of novice users are employees). In particular, deep knowledge about customers has been provided by an Italian bank for the development of a prototype system recommending funds; during the configuration of that system, the domain-dependent knowledge has been transferred to this frame-based representation quite easily. Moreover, less recently, a similar representation was successfully used in a health-care domain by physicians who had to configure the knowledge base for a diagnostic expert system (Molino et al., 1992).

Different from our approach, a probabilistic one would require the definition of numerical dependencies among user data and stereotypes, specifying the distribution of the probabilities over all the alternative classes of a stereotype family (e.g., given that the user is an employee, which is the probability that she is a novice user, or that she is an expert one). Moreover, since the user data might not be independent, more than one datum could influence the stereotypes of a family. In general, the store designer would be required to identify the dependencies among data and to define the matrix of the conditional probabilities which relate the user data to the stereotypes. This specification is complex and not particularly intuitive because it requires a deep knowledge of probability theory, which cannot be taken for granted, or imposed on the store designers.

**4.8.2. Comparison with multiattribute decision analysis**

In the evaluation of the degree of matching between the user’s data and the stereotypes, we decided to exploit a fuzzy match instead of multiattribute decision analysis since the former approach enables the system to rule out the mismatching stereotypes in a more effective way than the latter. In particular, the multiattribute decision theory is based on an additive evaluation of items and calculates their overall score as a weighted sum of the individual contributions carried by the attributes of the items.

As noticed in other work, this approach is well-suited to modeling the way humans rate alternative options; e.g., see (Linden et al., 1997;
Carberry et al., 1999). However, when deciding how closely a user matches a stereotype, the contribution of each classification datum must have a stronger impact than that supported by this theory: as already discussed, although the match must take into account all the classification data, the presence of totally incompatible data represents extremely strong evidence against the hypothesis that the user belongs to the user class described by the stereotype. Thus, a datum receiving a null individual score must have a drastic impact on the overall matching degree, bringing it to the minimum level, so that the stereotype can be excluded from the pool of stereotypes used to make predictions in the user model. Although, in multiattribute decision analysis, negative values can be provided for the data, the selection of values supporting this discrimination power is rather difficult. In contrast, the use of multiplicative formulae makes the task easier, because they produce scores ranging in a constrained interval ([0,1]) and guarantee that, if any contribution to the evaluation process is null, the whole matching degree is null.

Other functions have been used to compute the fuzzy AND of a set of conditions; e.g., the minimum evidence of the ANDe'd conditions, or their multiplication. As noticed in (Lesmo et al., 1985), the minimum function is too pessimistic: in fact, it only considers the contribution of the worst-matching datum. Instead, function (i) supports more precise evaluations: given two stereotypes having the same minimum score for a datum, the function takes the rest of the classification information into account to produce the overall matching degree. As long as the multiplication is concerned, it is rather similar to function (i); however, it takes lower values and is more sensitive to the cardinality of the set of scores to be combined in the evaluation of an overall matching degree. When the alternative hypotheses are ranked on the basis of different amounts of data, since the individual scores are numbers in [0,1], the multiplication tends to reduce the evaluation of the hypotheses characterized by the greater number of scores, even if such scores are high. Instead, function (i) maintains a higher evaluation of the hypotheses whose scores are good.

In our case, the matching degrees of the stereotypes of a family are evaluated on the basis of the same set of data; thus, the multiplication could be used instead of formula (i). However, we have presented this formula because, as we will see, the same evaluation technique is used in other parts of the system, to compare items which can potentially be characterized by different sets of data. In those cases, the use of the multiplication could mislead the system and produce biased evaluations of the alternatives.
4.8.3. Stereotypical information
Another comment concerns the use of stereotypes itself: although performing a detailed analysis of the sales domain to determine the characteristics of customer groups is a difficult task, this information is available to many organizations, which typically use it to design new products or services, and to select appropriate advertisements. This information is very important, as it can be exploited to customize the behavior of a Web store immediately after it is set up.

Automatic techniques should be applied to learn the stereotypes in the cases where the background knowledge about the customers is not known, or it is not precise enough: for instance, the store designer knows which user data and preferences should be described, but is uncertain about the compatibilities of their linguistic values. This issue has been explored by some researchers in the development of domain-specific adaptive systems and of user modeling servers; e.g., see (Orwant, 1995; Pohl and Nick, 1999; Paliouras et al., 1999; Paliouras et al., 2000). However, we believe that this is a rather complex issue and that, at the moment, no definite solution exists to it. For instance, we believe that a major issue for user modeling servers is the identification of which information about users (apart from demographic data and privacy preferences) and user behavior can be shared across different domains and successfully exploited in heterogeneous systems.

Up to now, we have not worked at the automatic acquisition of stereotypes; therefore, we assume that the store designer manually revises the Stereotype KB, if needed. Notice however that the identification of stereotype families simplifies the task of reusing the knowledge about users; in fact, although some families, like the destination use of products, are rather specific to the telecommunication domain, other families, like the users' lifestyle, describe user features and preferences relevant to several domains and can be used in other Web stores, targeted to the same customer base.

4.8.4. Dynamic user modeling
Although the exploitation of stereotypical knowledge enables the system to obtain fairly specialized user models as soon as the user has filled in the initial registration form, the individual user may differ from the stereotypical predictions for several reasons. In order to address this

---

9 Noticeable work has been done in the acquisition of information about shared behavior in task-based applications; e.g., see (Linton et al., 1999) and (Bauer, 1999). However, this work is developed in constrained domains, with a precise definition of the possible actions. In contrast, the analysis of log files produced by very general tools, such as Excite (Spink et al., 1999), only identifies very general user trends in user behavior. At the current stage, such trends can provide information about general usability requirements, but cannot support effective personalization strategies.
problem, we have extended the UMC to update the user model during the whole interaction with the customer: the user’s actions on the Web store catalog are captured by the Session Manager and monitored by the UMC, which analyzes her behavior to dynamically revise her features and preferences. We identified the information about the user’s interests and preferences carried by each type of action she may perform on the Web store interface: for instance, asking for technical information about an item provides positive information about the user’s technical interest and expertise, while asking for help on the description of a product feature provides information about her (missing) domain expertise. Asking for more information about an item (i.e., clicking on a “more information” link) provides evidence that the user’s receptivity is higher than what expected by the system, while adding an item to the shopping cart suggests information about the user’s preferences for product properties, and so forth.

Currently, the system does not monitor low-level events, such as the amount of scrolling and the movements of the mouse. Although, as discussed in (Goecks and Shavlik, 2000), this information is clearly useful to evaluate the user’s interest in the content of the displayed pages, we postponed its analysis for two main reasons: first, most of the pages produced by SETA can be fully displayed on the user’s browser. Second, the structure of the pages and the available buttons and links push the user to get the needed information in an active way. Thus, we have focused on monitoring the user’s actions within the navigational structure of the page, leaving the mouse activity as a secondary source of information to be investigated in our future work.

During the navigation of the catalog, the UMC maintains in its working memory all the relevant user actions and periodically analyzes this event history to obtain an up-to-date description of the user’s “level of activity”, for each action type. The collected information is fed as evidence to a Bayesian Net (Pearl, 1988) that models the dependencies among the user’s behavior and her features and preferences. The results of the evidence propagation process represent the new system’s beliefs about the user’s features and preferences and are exploited to revise the related data in the user model. Detailed information about the dynamic user modeling techniques exploited in SETA is provided in (Ardissono and Torasso, 2000).

5. Suggestion of Goods to the User

The Product Extractor handles the information about the products of the catalog. When the items of a product have to be displayed, this
agent retrieves the records matching the user's query (e.g., all the phone models) from the Products Database (DB). Then, it ranks and sorts them, depending on how closely they match the preferences stored in the model of the person who will receive the items, i.e., their beneficiary. This sorted list is used to present the best items before the other ones.

As described in section 4, the user's preferences concern product properties such as their quality and ease of use. Thus, the system's suggestions help the user to identify, among all the alternatives, the items which best satisfy her preferences. We currently leave aside the impact of individual features of items (e.g., the presence or absence of the agenda in a phone) in their evaluation. In other words, we assume that, when the user is focused on a particular product, her preferences are the only information needed to rank the items available for that product. This is not restrictive, because the presence of strict constraints can be handled by pre-selecting items and identifying those offering the mandatory features; then, the user's preferences can be considered to sort items of the resulting list on a preference basis.

5.1. Knowledge about product categories and items

The structure of the domain is explicitly described in a knowledge base called the Product Taxonomy; this is a conceptual representation of the catalog and defines the products and the relations in an inheritance net. Moreover, this taxonomy defines the product features, specifying their range of admissible values and their type. We have defined a restricted number of types, which define the meaning of a feature in a generic way; i.e., we use the existence of technical, functional and aesthetic features.

Each product category is described by a set of features, representing the offered functionalities and other more general information, such as price, size and color. In addition to the description of features, all the products are characterized by a set of properties, corresponding to the preferences stored in the user models: e.g., quality, design, ease of use, and so forth.

The Products DB contains the information about the individual features and properties of the items available in the Web store. The items are classified into the product categories defined in the Product Taxonomy (e.g., phones, faxes and answering machines) and are described by means of records containing specific information about their individual features and properties. The record of an item stores its name, code, and the value (or range of values) of its features. Moreover the record

---

10 For instance, the color of the answering machines can take several values, but the "Facile" model is only available in two colors: grey and black.
Record 1:
Name: Facile;
Code: 100025;
Features:
Price: LIT. 106000;
Color: grey, black;

Properties:
Quality: high;
Ease of use: high;
Cost: medium;
Design: high;

Record 2:
Name: BIP 9050;
Code: 100130;
Features:
Price: LIT. 89000;
Color: anthracite;

Properties:
Quality: medium;
Ease of use: high;
Cost: medium;
Design: high;

Figure 7. Records of two items of the Products DB.

contains, for each property, one linguistic value fitting the item. Figure 7 shows the description of two answering machines: “Facile” and “BIP 9050”.

5.2. Evaluation of items

The Product Extractor gets from the UMC the preferences of the beneficiary of the goods. Then, it ranks each item by evaluating how closely it matches such preferences. The overall score of an item corresponds to the degree of matching between the properties of the item and the beneficiary’s preferences: it is a number in [0,1], where “0” represents the fact that the item mismatches the user’s preferences, while “1” represents a perfect match, and thus denotes an item to be recommended to the user.

5.2.1. Computing the overall matching degree

In the evaluation of the degree of matching between an item and the user’s preferences, each property is separately considered, to check how closely it matches the related user preference; then, the scores of the properties are merged to produce the evaluation of the item.

Similar to the classification of users in stereotypes, the contributions of the properties are merged by exploiting a multiplicative formula, which enables the system to easily exclude mismatching items; although the degree of matching is evaluated on the basis of the whole set of properties of the item, if at least one of these properties receives a null score, the overall score of the item must be null, as well.
However, the evaluation of items differs from the classification of users for two aspects:

1. While the stereotypes of a family are evaluated by taking into account exactly the same set of user data, this assumption cannot be made as far as items are concerned. If no information about a property of an item is available in the Products DB, the overall score has to be computed without taking the corresponding user preference into account; a lack of information about items weakens the system’s ability to identify the best items for the user, but must be at least partially tolerated.

2. As represented in the user model, the user may have stronger or weaker preferences for product properties; thus, not all the preferences have the same impact in the evaluation of the suitability of items. While the items mismatching extremely important user preferences should not be recommended, those mismatching almost irrelevant properties could be anyway recommended. In general, an item suited to the user should match all her important preferences, possibly ignoring irrelevant mismatching properties.

The first issue is handled in the formula used to merge the individual scores of the properties and will be discussed in section 5.3. As far as the second issue is concerned, each property receives an individual score representing the acceptability of its value for the user. A score equal to 1 means that the property value is perfectly compatible with the related user’s preference (i.e., the likelihood of its value in the user model is extremely high) or, in alternative, that the property is irrelevant to her (the importance of the preference in the user model is extremely low) and should not contribute to downgrading the overall rating of the item. In contrast, a null individual score means that the property value is important to the user and incompatible with her preferences.

The individual scores received by each property are combined in a fuzzy AND: the Product Extractor exploits the same formula described in section 4.6.2 for the evaluation of how closely the user matches the profile of a stereotype. Thus, given two properties $A$ and $B$, and their individual scores, $score_A$ and $score_B$, the overall score for the item is calculated as follows:

$$SCORE(score_A, score_B) = \frac{score_A \cdot score_B}{(score_A + score_B - score_A \cdot score_B)}$$ (ii)

In the following, we describe how the importance of the user’s preferences is taken into account in the evaluation of the individual score of a property.
5.2.2. Taking into account the importance of the user preferences

As described in section 4.3, a preference for a product property $P$ is characterized by two types of information: the importance of the preference and the likelihood of its linguistic values. The importance describes how strongly the preference influences the user's decisions. The likelihood of a linguistic value $lv$ approximates the probability that the user prefers products for which the property $P$ has the value $lv$; e.g., that she prefers high-quality products.

In order to take the importance of the user's preferences into account, the individual score of a property is computed by combining the importance of the related user preference with the likelihood associated with the linguistic value of the property. As already mentioned, the individual scores are merged by means of the (ii) fuzzy AND formula. Since the neutral value for the multiplication is “1”, we want an irrelevant property to receive a score near “1”, regardless of how closely it reflects the user's preferences: in this way, the influence of the property on the matching degree decreases proportionally according to the importance of the user's preference. Moreover, a score near “1” should be given to a property when the likelihood that the user prefers products with that property value is very high.

More specifically, consider an item and its property $A$ and suppose that the linguistic value fitting the item is $a_i$. The Product Extractor gets from the user model the importance of the preference toward $A$ ($Imp_A$) and the likelihood of the linguistic value ($p_{ai}$). The individual score of $A$ is computed in the following way:
\begin{table}[h]
\centering
\small
\begin{tabular}{|c|c|c|c|c|}
\hline
\textbf{Importance}_{P1,P2} & \textbf{Likelihood}_{P1} & \textbf{Score}_{P1} & \textbf{Likelihood}_{P2} & \textbf{Score}_{P2} \\
\hline
0.0 & 0.8 & 1.0 & 0.1 & 1.0 \\
\hline
0.2 & 0.8 & 0.96 & 0.1 & 0.82 \\
\hline
0.4 & 0.8 & 0.92 & 0.1 & 0.64 \\
\hline
0.6 & 0.8 & 0.88 & 0.1 & 0.46 \\
\hline
0.8 & 0.8 & 0.84 & 0.1 & 0.28 \\
\hline
1.0 & 0.8 & 0.8 & 0.1 & 0.1 \\
\hline
\end{tabular}
\caption{Impact of the importance of properties on the evaluation of their score.}
\end{table}

\[ \text{score}_A = \text{Imp}_{P1} \times p_{h_i} + (1 - \text{Imp}_{P1}). \]  

This Bernoulli formula raises the score of the less important data, while it does not have any impact on the extremely influential data; e.g., the score of a property corresponding to a totally irrelevant preference (where $\text{Imp}_{P1} = 0$) is “1”, for any possible value of $p_{h_i}$. Instead, if a preference is extremely important ($\text{Imp}_{P1} = 1$), then the individual score of the related property is evaluated as $\text{score}_A = p_{h_i}$.

Figure 8 shows how the importance of properties influences the evaluation of their individual scores, in the case of two properties ($P1$ and $P2$): in the first case ($P1$, represented by the solid line), the likelihood of the linguistic value of the property in the user model is 0.8. In the second case (dashed line, property $P2$) the likelihood is 0.1. As can be seen, the importance of the properties influences the resulting scores which take equal or higher values than the respective likelihoods. In both cases, if the importance is low, the individual scores are high;\(^{11}\) instead, when the importance grows, the individual scores take values close to the related likelihoods. The exact scores computed for the two examples are shown in Figure 9: in the table, the first column reports the values of the importance; the second and third ones show respectively the likelihood and score computed for property $P1$; the fourth and fifth columns show the corresponding values for $P2$.

\subsection*{5.3. Comments}

After the Product Extractor has computed the overall score for each item, it can sort the presentation list, so that the best matching items can be displayed first. As described in section 4.8, the formula (ii)\(^{11}\) Thus, the score does not sensibly downgrade the overall match of the item.
<table>
<thead>
<tr>
<th>Item</th>
<th>Quality</th>
<th>Ease of use</th>
<th>Cost</th>
<th>Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facile</td>
<td>0.68</td>
<td>0.75</td>
<td>0.51</td>
<td>0.2</td>
</tr>
<tr>
<td>BIP 9050</td>
<td>0.52</td>
<td>0.75</td>
<td>0.51</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Figure 10. Individual scores of the properties of “Facile” and “BIP 9050”.

has good discrimination power, which enables the system to rule out the items receiving a null individual score for at least one of their properties. This is important, because a null individual score is assigned to the properties of items that are extremely important to the user, but mismatch the user’s preferences.

Another advantage of (ii), with respect to other formulae for the evaluation of the fuzzy AND of a set of conditions, is its flexibility in the evaluation of items characterized by different sets of information. In fact, although the properties of an item may be only partially specified in the Products DB, the items should be ranked by exploiting all information available to the system.\textsuperscript{12} Thus, the alternative items might need to be evaluated by taking into account sets of properties with different cardinalities. In that case, the items whose properties have been fully described should be evaluated in a fair way with respect to those only partially described.

In section 4.8, we have already discussed the fact that our fuzzy match formula (denoted as (i) in that section) is more accurate than the minimum function, which would take the minimum individual score of the joint conditions as the result of the fuzzy AND. Actually, (ii) is also better than the multiplication, because it is less sensitive to the number of scores combined in the evaluation. In particular, if two items are ranked on the basis of a different number of properties, but both of them receive high individual scores for each property, the item characterized by the largest number of properties is less exposed to the risk that its overall ranking is downgraded because of the fact that many numbers in [0, 1] are combined.

As an example, consider Paul’s user model, shown in Figure 3, and the descriptions of items “Facile” and “BIP 9050”, shown in Figure 7. Figure 10 reports the individual scores received by the properties of the two items; each score is obtained by applying the formula (iii) to the importance and the likelihood of the related preference, as stored

\textsuperscript{12} An evaluation only based on the information present in all the records might lead the system to ignore data potentially essential to the evaluation task: e.g., if the ignored datum is associated with an important property mismatching the user’s preferences, the item should be ruled out.
in Paul’s model. The scores suggest that Paul should slightly prefer “Facile” to “BIP 9050”.

We now consider the results obtained in the evaluation of the overall scores of the two items by applying, respectively, the multiplication of the individual scores, and the formula (ii). We evaluate the overall scores in two cases: in the first one, we assume that all information about the item properties is available; in the second one, we assume that the items’ ease of use is missing.

- By multiplying the individual scores, we obtain the following results:

  - Full information about the items:
    “Facile”: 0.05202;
    “BIP 9050”: 0.03978;

  - Partial information about the items (missing information about the ease of use):
    “Facile”: 0.06936;
    “BIP 9050”: 0.05304;

- By applying formula (ii) to the individual scores, we obtain:

  - Full information about the items:
    “Facile”: 0.147826;
    “BIP 9050”: 0.138558;

  - Partial information about the items (ease of use missing):
    “Facile”: 0.155488;
    “BIP 9050”: 0.145267;

These results show that the formulae evaluate the items consistently with Paul’s preferences when the items are compared on the basis of the same number of properties. However, if the two items are compared in a situation where the Products DB reports full information about “Facile” and partial information about “BIP 9050” (omitting its ease of use), only the application of formula (ii) enables the system to recognize that “Facile” is the best item. In particular, the multiplication of the individual scores returns the following matching degrees, estimating “BIP 9050” as a better item than “Facile”:

“Facile”: 0.05202;
“BIP 9050”: 0.05304;

Instead, the (ii) formula returns the following matching degrees, consistent with Paul’s preferences:

“Facile”: 0.147826;
“BIP 9050”: 0.145267;

Figure 11 compares a set of rankings obtained by exploiting, respectively, the multiplication and formula (ii), in a graphical way. In the
Figure 11. Ranking items by means of the multiplication and the (ii) formula.

comparison, we have supposed for simplicity that all the individual scores of the properties take the same value (0.9) and we have compared the results of the two formulae, applied to a growing number of properties (from 0 to 10). Although the results decrease for a growing number of properties, the (ii) formula takes higher values than the multiplication and its rankings are less sensitive to the number of evaluated properties. More specifically, when items are combined on the basis of a significative number of properties, the rankings produced by (ii) decrease more slowly than those produced by the multiplication; if the items are compared on the basis of very few properties, the two formulae produce similar comparative results.

6. Personalization of the Presentation

The Personalization Agent dynamically generates the HTML code for the catalog pages, tailoring their content and layout to the characteristics of the individual customer. Since these pages are generated “on the fly”, their content is determined on the basis of the values of features and preferences stored in the user model, at each step of the interaction.

In the customization of the catalog pages, we have focused on two main issues: the selection of their content and the selection of their layout. The content concerns the amount and type of information to be included in each portion of a page. The layout concerns graphical aspects like background colors and fonts, but also other issues, like the number of items to be described in a single page.
As far as the link structure of the catalog is concerned, the Personalization Agent offers a partial adaptation: the items available for a product category are presented as a list sorted according to the user’s preferences; so, different sequences of items are presented to heterogeneous users. However, the set of links which enable the user to browse the catalog moving from one product category to another one are fixed and correspond to the relations among products defined in the Product Taxonomy. This restriction has the drawback that the logical structure underlying the catalog, as far as product categories are concerned, is the same for all users and corresponds to the store designer’s view of the relations among product categories. However, it enhances the consistency of the Web store: for instance, users visiting the catalog more than once are always presented the same hypertextual structure.

6.1. Generation of the Catalog Pages

Several types of pages have to be generated to support the navigation in the Web store. For instance, at the beginning of the interaction, the system displays a registration form where the user can specify her personal data; then, the system shows a page where the user can select the products that she would like to see in detail. When the user follows a link to see the items available for a certain product, the corresponding page must be produced; and so forth.

At each stage, the Dialog Manager selects the type of page to be produced next and requests it from the Personalization Agent, which generates the code of the page and sends it back to the Dialog Manager, for carrying on the interaction with the customer.

The Personalization Agent exploits different kinds of information to generate a page, depending on its type (e.g., a form, a page presenting a product / item, or other). In the following, we will focus on the pages describing the items available for a product category, because they are the most interesting ones from the customization viewpoint.

The Personalization Agent uses the following information to generate the Web pages:

– The set of items to be presented and the information about their features. This information can be retrieved from the Product Taxonomy and the Products DB.

– The internal structure of the catalog, used to decide which hypertextual links can be included in the page: for instance, links to products related to the one in the focus of attention.
The interaction context where the user's selections and a memory of which products and pages she has already seen are stored. This context is maintained by the Dialog Manager, which keeps track of the user's focus of attention.

- The user features, stored in the user model, which are exploited to select the graphical aspect of the page, the amount and type of information to be provided, and the technicality of the descriptions.

Personalized pages are generated, choosing alternative layouts, colors and fonts, depending on the user features. Moreover, the items are presented in different ways: the selection of the content to be included in the pages is tailored to the user's interests and receptivity; the selection of the linguistic form of the descriptions is tailored to the user's expertise. Thus, the same information content can be presented in several ways, depending on the user's characteristics.

6.2. STRUCTURE OF THE PAGES DESCRIBING ITEMS

Before describing the personalization techniques used to customize the presentation of goods, we outline the structure of such pages.
A presentation page describing the “Facile” answering machine is shown in Figure 12; the portion of the page showing the picture and the features of the item is repeated in Figure 13 for better readability.

As shown in Figure 12, the catalog page is split into several areas, displaying the contextual information, the navigation and control buttons, and so forth. The central area of the page is devoted to the description of the item and contains links to information about it. More specifically:

- The topmost bar, beside the store logo, provides the links to the main product categories available in the store.
- In the leftmost portion of the page the system displays the active interaction paths and enables the user to switch among them. Each path represents a dialog context, that specifies:
  - The target of the product: the user may consider products for home or business use: this information is displayed by means of an icon. Moreover, she can select goods for herself, or for somebody else. In this example, she is looking for a product for her own home, and for another product, for Mary’s office.
  - The initial selection: when entering the store, the user is asked to choose the main product categories she is interested in. The system keeps track of this choice providing the label “Initial selection” and a link to the category itself. In our example, the user initially selected the answering machines and the phones categories.
• The last visited page, for each dialog context: while browsing the catalog, the user can move from the page presenting the initial product category to pages describing other products or showing the available items. The system displays this information under the “Last visited” label. In Figure 12, this label is present in the second context and shows that the user moved from the initial selection, “phones”, to the page describing the items of “multifunction phones”.

The current context is highlighted and the “Last visited” label is replaced with “Now displayed”. In the example, the user is looking at a page presenting the items of answering machines.

– The central area describes the functionalities and features offered by the presented item. This area can contain one or more items, depending on the user's receptivity: if the system assumes that the user is in trouble when too much information is provided in the same page, it describes only one item per page, as in Figure 12; otherwise, it can list two or three items per page. For each displayed item, the area devoted to its description contains:

• Its name, picture and price.
• A button to put the item into the shopping cart.
• A button to display the technical details about the item.
• The description of its features. If the system shows only a subset of the features, a “more information” link is added to see the whole list.

– The lower area of the page, below the description of items, contains several buttons and links. In particular:

• The links for browsing the list of available items: “previous items” and “next items”.
• A button to display the whole list of items available for the current product category. The user can use this list to ask for the description of a specific item, without browsing all the pages of the items, via the previous/next items links.
• A button to create a customized comparison table: the user can create such tables “on the fly”, by selecting the products to be examined and the features which she would like to consider. In this way, she can avoid the examination of large, precompiled structures, which compare all the items of a product category on the basis of all their features.
• The link to go back to the page describing the product category of the displayed item. This link has the “back to” label and, in the figure, refers to the page describing the answering machines.

• The bottom bar, containing general control buttons, such as the “end of session” one and the link to the site map.

6.3. Selection of content in pages describing items

Not all the portions of a catalog page are personalized: some parts only depend on the interaction context. For instance, the left area reporting the user’s selections only depends on which products she has decided to see, and which one represents the current focus of attention. Moreover, the content of other areas, like the top bar showing the main product categories, or the bottom bar of the pages, is fixed for all the pages. From the personalization viewpoint, the most interesting area is the one describing a specific item: in fact, the system tailors the number of features to be shown, their order and the technicality level of the linguistic descriptions to the individual user.

The Personalization Agent exploits a set of rules to customize the description of the features of an item and operates in the following way:

1. It sorts the features to be described, on the basis of their relevance to the user’s interests and their intrinsic importance to the description of the item.

2. It decides how many features should be displayed, on the basis of the user’s receptivity. Those falling out of the limit are linked to the page by means of the “more information” button.

3. It selects the linguistic form for the descriptions, on the basis of the user’s expertise.

4. It produces the HTML code for the portion of the page.

In the following, we describe these tasks in detail.

6.3.1. Evaluation of the relevance of features

We assume that the relevance of a feature depends on two main parameters: the user’s interests and the objective importance of the feature to the description of the item. Both parameters are important: in fact, the item descriptions should focus on the information interesting to the user; however, particularly important features should not be omitted, even if they don’t fit the user’s interests perfectly.
We have defined four feature types, corresponding to high-level interests, as described in the “user features” part of the user models.

- Technical features represent technical information about products; for instance, the printing resolution is a technical feature of faxes.

- Functional features represent the information about the particular functionalities offered by the items; for instance, the agenda is a functional feature of phones.

- Aesthetic features represent generic information about the look of items; for instance, the color and the size.

- Generic features include miscellaneous information, such as the price of items.

Each feature $F$ is characterized in the Product Taxonomy by its type and its importance.

- The *type* is the classification of $F$ into one of the four defined categories of features.

- The *importance* denotes the objective importance of $F$ to the description and can take one of the following values:
  $Imp_F = 1$ if $F$ has a low importance.
  $Imp_F = 2$ if $F$ has a medium importance.
  $Imp_F = 3$ if $F$ has a high importance.

The features of an item are ranked by combining their importance with the user's interest in an additive formula:

$$score_F = w_1 * Imp_F + w_2 * Interest_T$$

where $Imp_F$ is the importance of $F$ and $Interest_T$ is calculated on the basis of the user's interest in the category $(T)$ of features to which $F$ belongs. More precisely:

$Interest_T = 1$ if the user's interest in $T$ is low;

$Interest_T = 2$ if the user's interest in $T$ is medium;

$Interest_T = 3$ if the user's interest in $T$ is high;

The user's interest is retrieved from the user model, as the most probable linguistic value of the related user feature. For instance, referring to Figure 3, Paul's interest in technical features is medium, while his interest in aesthetic features is high.

The weights in the formula are used to tune the influence of the two types of information in the evaluation of the overall score of the features. In our prototype, $w_1 = w_2 = 1$; however, the store designer
may configure them, depending on whether the user’s interests, or the
objective importance of features, should influence the evaluation in the
strongest way.

Given an item to be presented, all its features are ranked by applying
the formula above. The result is a sorted list of features, partitioned in
groups of equally scored elements.

6.3.2. Evaluation of the amount of information to be displayed

The user’s receptivity, available in the user model, is exploited to tune
the length of the presentations, so that more or less detailed descrip-
tions of the same item can be generated, depending on the system’s
estimate of the user’s ability to absorb information. We measure the
length of a description in terms of how many features are shown in
the page. The number of features to be presented is applied to the
previously sorted feature list.

For each linguistic value of the receptivity, we have set a threshold
describing the maximum number of features to be shown, in a gen-
eral situation. The maximum number of features to be described for a
specific item is calculated by taking into account the thresholds in the
following way:

\[
\text{number of features} = x \pm \delta
\]

where \( x \) is the threshold associated with the level of the user’s recep-
tivity (approximated to the most probable linguistic value of the related
user feature) and \( \delta \) is used to cut the feature list in a flexible way: if
the value of \( \delta \) falls inside a group of equally scored features, a decision
should be taken to identify which features have to be shown and which
ones should be excluded; \( \delta \) enables the system to extend or restrict the
number of features to be shown.

For instance, let’s assume that the user’s receptivity is high and \( x \)
takes value \( x_0 \); if the number of equally ranked features exceeding \( x_0 \)
is greater than \( \delta \), then the group containing these features is hidden;
otherwise, it is shown.

The principle underlying the personalization is that the system should
never omit any information: it should only highlight the most inter-
esting data, by sorting it and constraining the descriptions. For this
reason, the most relevant information is presented in the portion of
the page describing the item, but the rest of the list is reachable via
the “more information” link, as a form of adaptive stretchtext useful
to get detailed information on demand; see (Kobsa et al., 1994;
Brusilovsky, 1996).
6.3.3. Selection of the linguistic form

The linguistic form for the presentation of the features is selected on the basis of the user's expertise, whose value is approximated to the most probable linguistic value in the user model. Simple descriptions are produced if the user has a low domain expertise, while technical (and typically compact) ones are produced for expert users. In the version of SETA described in this paper, the descriptions are almost completely handcrafted; in order to separate the specification of the meaning of a feature from that of its values, the descriptions are structured in more than one part and include parameters as place-holders for the expressions specifying the feature values. The system extracts from the Products DB the values of the feature offered by the item. Then, it generates the complete linguistic description by filling in the place-holders with linguistic expressions corresponding to such values. For instance, the “color” has the following parametric description:

It is available in the following colors: #0,

where “#0” is the place-holder for the linguistic description of the feature values. When the feature has to be described, the available colors are extracted from the Products DB and the “#0” parameter is replaced with their linguistic description. For instance, as far as “Facile” is concerned, we obtain:

“It is available in the following colors: grey, black.”

A difficulty level is associated with each description, basically depending on its technicality. The difficulty levels are related to the linguistic values of the expertise in a one-to-one correspondence: there are simple, medium and technical descriptions. Thus, given the user's domain expertise, the selection of the appropriate alternative is straightforward. Consider, for instance, the three descriptions available for the “digital memory” of the answering machines:

“<L>It stores the received messages using a secure technology, which enables you to listen to them without rewinding the tape.”

“<M>It stores messages using a digital technology.”

“<H>Message storage on digital memory.”

The “<L>”, “<M>” and “<H>” markers specify the low, medium and high complexity levels. Since the feature takes a boolean value (either the item offers a digital memory, or it does not), the place-holder for its linguistic description is not used in this example: only features with positive values, i.e., those actually offered by the item, are described; negative descriptions are simply omitted.

The presence of alternative descriptions requires the introduction of variants for the same feature, but is the simplest way for tuning the language to the user models. For instance, in the above example, the
description of the digital memory for the low expertise level explains the feature very carefully, using simple terms. On the other hand, the description for the high expertise level is short and technical, under the assumption that an expert user already knows what a digital memory is. Anyway, the system can be configured in a single-description modality, where the same parametric description is used for all the three expertise levels. In fact, when configuring a new Web store, the designer decides whether the interface should tailor the linguistic form to the user’s expertise or not. In the second case, the same parametric description (probably, a medium-level one) can be introduced for all the three difficulty levels.

We are developing a tool which supports the configuration of the store interface. This tool automatically proposes the first description introduced by the designer for one level as a default for the other ones. For instance, suppose that the store designer provides the following medium-level description for the digital memory: “It stores messages using a digital technology”. The tool will propose the same description for all the three levels and, if the designer accepts this option, the store will work in a single-description modality, at least as far as the digital memory is concerned. However, the alternative descriptions for the three levels can be included at any time, by redefining the parametric descriptions associated with the “high” and “low” values of the domain expertise.

6.3.4. Generation of the descriptions
The previously described steps select the content of the portion of the page devoted to the description items. The final HTML code is generated from this text, by exploiting a set of layout packages that will be described in section 6.4. However, at this stage, some special markers are included in the text, to highlight the most relevant features. For instance, the group of features having the highest score can be marked to be displayed in boldface, while the minor features are marked to be shown in a smaller font than that used for the rest of the page.

6.3.5. Comments
The customized product descriptions produced by the Personalization Agent may differ in several aspects, among which:

- The set of presented features and their number;

- The order in which the features are displayed;

- The linguistic style adopted in the descriptions.
For example, compare Figures 13 and 14. The fragment shown in Figure 13 has been generated for a novice user, not very familiar with telecommunication products, with low receptivity and high aesthetic interest (e.g., Paul, see Figure 3). On the other hand, the fragment shown in Figure 14 is extracted from a page presenting the same answering machine to an expert, receptive user, very interested in technical details. The first page contains simple descriptions, where technical terms are avoided; moreover, it only shows 5 features of the item, 2 of which are aesthetic characteristics. Instead, the description in Figure 14 contains technical descriptions and includes 10 features. The two pages also differ in the selection of the displayed features: as the second description contains much more information than the first one, all the features described in Figure 13 are shown in Figure 14, as well: they are just sorted differently. However, in other examples, some features may be hidden in one page, while they are shown in the other one. The selection of the descriptions to be linked as further information is critical: in fact, the user does not necessarily follow the “more information” links to read the hidden data. However, as discussed later on in section 7, constraining the number of displayed information items is beneficial to less receptive users, who appreciate short descriptions.

The dynamic generation of the catalog pages, based on information provided by declarative knowledge sources (such as the Product Taxonomy and the Products DB), avoids the need to maintain multiple static versions of the catalog, suited to the various types of users, as was done in the first adaptive hypermedia systems: e.g., see (Popp and Lödel, 1996; Brusilovsky, 1996). In fact, we don’t need to store redund-
dant information about the content of pages; moreover, the pages are adapted to the user’s needs at the granularity level of the individual product features deciding, one by one, how important it is, and so forth. Some redundancy still comes from the use of the alternative parametric descriptions, in the three technicality levels. However, in the latest version of our system, we have improved the generation of such descriptions: now, template-based Natural Language Generation techniques (Reiter, 1995) are used to produce the sentences describing the features in a more flexible way; see (Ardissono and Goy, 2000).

The classification of features into types and the description of the user’s interests in such information types supports the exploitation of ranking techniques which are applicable across different sales domains. Moreover, this approach does not require the introduction of complex knowledge bases explicitly representing all the domain concepts. In fact, the only information needed to apply the personalization techniques to a feature is the definition of its type and importance to the descriptions. Although the four feature types which we have defined are relevant in a technical sales domain and may not be sufficient in very different applications, the system can be extended with new types, with a modest effort.

6.4. SELECTION OF THE PAGE LAYOUT

SETA provides four layout packages, suited to different types of customers; given the model of the current user, the system selects one of the packages and uses it for formatting the catalog pages.

6.4.1. The layout packages

The alternative layouts mainly differ in the background colors and the fonts they use. In particular:
– The “readable” package focuses on the readability of the information and is suited to elderly people.

– The “young” package includes bright colors, original fonts and big pictures and is suited to young people.

– The “elegant” package is characterized by pale colors and classic fonts. This package is suited to customers sensitive to pleasant presentations.

– In the “technical” package, colors are not very important, fonts are simple and pictures are small. This package is suited to customers mainly interested in the technical aspects of the products.

We describe the internal representation of the layout packages by considering the “elegant” one, whose definition is shown in Figure 15 and which has been used to generate the catalog page of Figure 12.

The Profile section contains the set of user features exploited for the selection of the package. These features correspond to a subset of the characteristics and interests defined in the user models. The only exception is the “life style” datum, that explicitly refers to the classification of the user into the stereotypes belonging to the “life style” family: see the “User classification” section of the user model, in Figure 3. For each datum of the profile, the linguistic values compatible with the package are specified.

The Layout settings section contains the definition of the layout: it describes the colors, fonts, images to be used in the HTML code of the page under construction. We don’t describe these data in detail, since the names are self-explanatory.

6.4.2. Evaluation of the layout packages
Each layout package is ranked by calculating the matching degree between its profile and the related data in the user model. For each datum, the linguistic values compatible with the package are considered and the likelihood of such values is retrieved from the user model. If the datum has one compatible value, its score is the corresponding likelihood; if there is more than one compatible value (e.g., “technical interest: low, medium”), the maximum likelihood is selected.

The overall score of the package is calculated by applying formula (ii), already used for the evaluation of items, to the individual scores of its data. In this way, if at least one datum is totally incompatible with the user model, the package receives a null evaluation.

\[13\] The likelihood of each linguistic value of the “life style” datum corresponds to the degree of matching between the user and the related stereotype, reported in the “User classification” part of the user model.
6.4.3. Selection of the layout packages
After the evaluation of the alternative layout packages, the Personalization Agent selects the best matching one and includes its layout settings (colors, fonts, etc.) in the HTML code of the store pages.

Although the content of the user model can change during the interaction, the layout remains fixed until the user exits the Web store to provide a coherent view of the catalog.

6.4.4. Other personalization aspects
Another personalization aspect that influences the layout of the pages presenting items is the number of models shown in a page. This number has to be configured by the store designer and three values can be set, corresponding to the three levels of the user’s receptivity, defined in the user models. Our current prototype displays only one model per page if the user receptivity is low (as in Figure 12), two if it is medium, and three if it is high. The number of models per page has a strong impact on the amount of information provided in a page and, obviously, on the length of the page itself: longer pages with richer information are produced for highly receptive users, while shorter and simpler ones are generated for less receptive people.

6.4.5. Comments
We decided to use the “life style” datum in the profile of the layout packages, although it belongs to the system’s domain-dependent knowledge, because the customers’ life style is a very important factor in the determination of their “tastes”. For instance, in other systems, the life style is exploited to select the look and style of the advertisements; e.g., see (AlMedia) and (Ardissono et al., 1999b). In our system, we exploit this factor to choose the layout on the basis of the information provided by a set of parameters, including objective parameters, such as the users’ sight, and subjective ones, such as the user’s tastes. Given the importance of the life style, we assume that in most sales domains the customer population will be segmented on the basis of this viewpoint. In any case, the profile of the layout packages is totally configurable; thus, if the life style is not considered in a specific domain, the store designer can easily modify the definition of the packages.

6.5. Generation of the HTML code of the catalog pages
The personalization Agent generates the final HTML code of a catalog page by putting together the HTML fragments of the various portions of the page. In the final step of the code generation, the system produces such structure by inserting the fixed parts of the code, which
are retrieved from an internal HTML tag repository. Then, it inserts the background colors and font settings provided by the selected layout package. Finally, it includes the portions of HTML code produced for the areas of the page.

6.6. Personalization in Other Electronic Commerce Systems

We have focused on a subset of the personalization techniques which could be exploited to improve the front-end of a Web store and other strategies could be added to SETA for enhancing its adaptivity; e.g., see (Kobsa et al.) and (Riecken, 2000). For instance, our system delivers textual information and images, but other media could be exploited to improve its usability for customers with different technical requirements, as done in (Joerding, 1998; Joerding, 1999), or with users characterized by special needs (Fink et al., 1998). Moreover, life-like characters could be used to actively interact with the user in special cases, such as the provision of help (André and Rist, 2000). Finally, the Web store might be animated by exploiting some basic Virtual Reality techniques, as done in (Chittaro and Ranon, 2000).\textsuperscript{14}

In SETA, we preferred to focus on the quality of the descriptions produced by the system, leaving aside other presentation forms, because a careful description of product features is in our opinion the central requirement for domains such as the sales of telecommunication products. In other domains, like the sales of cars, other functionalities, such as the possibility of allowing the user inspect the shape of a car, or see short movies showing its performance, become crucial to the improvement of the presentation and have to be taken into account. In those cases, the exploitation of monitoring activities, such as those described in (Joerding, 1998), is essential to recognize the user’s preferred media and customize the layout of the presentations accordingly.

7. Evaluation of SETA

The current interface of SETA, its interaction style and its functionalities are the result of several revisions, which we have done thanks to the comments and suggestions of about 100 users, involved in a subjective evaluation of our telecommunication prototype. We could only perform a set of laboratory tests, while an evaluation of the system with real users was not possible. In fact, the prototype is not a real Web store, where people can purchase goods: users can select items and put them

\textsuperscript{14} The use of Virtual Reality is however a critical aspect as far as the portability and efficiency of the system is concerned.
into the shopping cart, but they cannot pay for them; so, we cannot be certain that their behavior is fully motivated, especially as far the selection of goods is concerned.

In the following, we will report the results of this first test phase, describing the consequent revisions to the system. Then, we will describe the initial results of a second test phase which we are currently carrying on (again, in our laboratory), using the interface of SETA described in this paper. Finally, we will conclude with some comments about a completely different type of evaluation, concerning the applicability of the SETA system shell to other application domains, such as the one of personalized news servers.

7.1. First Test Phase

In this phase, we collected information about the behavior of quite heterogeneous users: we have involved psychologists, computer scientists and people belonging to the economic, administration, and humanities areas, asking them to try our prototype and make comments about its interaction style, graphical layout, presentation strategies, flexibility and understandability of the interface, and so forth. The test was carried out in several steps: at each one, we revised the system on the basis of the collected information and tested it again, to check the impact of the changes on new users. A summary of the results follows:

- Our users did not complain about the initial registration form, where the personal information is asked for, provided that they were informed about two main issues:
  
  - They could skip the initial form, or fill it in partially.
  - The requested information was only used for personalization purposes during the individual session and was not permanently stored, unless users authorized the system to do that, at the end of the sessions.

  The acceptability of the registration form is not surprising, given its similarity to the forms typically provided by the most well-known Web-based services.

- A critical issue is the selection of the high-level product categories which the user would like to inspect and the specification of the beneficiary of the related goods. Although this information is essential to tailor the suggestion of goods to their actual target, it introduces an overhead in the interaction. Initially, we proposed a form for the introduction of such information, but this solution required
that the user separately answered quite a long list of questions. To limit the burden on the user, we then designed the “drag & drop” selection page shown in Figure 2: this page enables the user to select multiple products for multiple beneficiaries in an straightforward way and largely reduces the number of actions that the user has to perform for notifying the system about her selections.

- Another critical aspect of the interaction with the user concerns the browsing in the hypertextual catalog, to visit the various product categories: as described in section 6, the catalog structure is determined by the Product Taxonomy, which reflects the store designer’s view of the sales domain and does not necessarily match the user’s viewpoint. Some users had problems in finding the navigation path which led them to the desired products. In order to improve the clarity of the catalog, we have structured it as a two-level hypertext, where the higher level describes the product categories and the lower level describes the items available for a specific category. At the higher level, each product page synthetically describes the functionalities offered by the presented product category and offers two types of links: the link to the items available for that category (to enter the lower level of the hypertext) and those leading to other related product categories. These links enable the user to quickly browse the catalog: user is provided with an overview of the main functionalities offered by the product categories, without checking detailed descriptions of the individual items.

- As far as the presentation of items is concerned, most users appreciated the personalization of the descriptions and the suggestion of the most interesting items before the other ones. However, some users asked for the introduction of tools which enable them to bypass the presentation order established by the system. In fact, they wanted to be able to access goods in a direct way, without following the system’s suggestions. The users also asked for the introduction of tools for the comparative evaluation of items. Thus, we extended the interface of the Web store as follows:

  - We extended the pages presenting items by adding a “list of models” button to get the complete list of items belonging to a product category. This button enables the user to directly request the presentation page of a specific item.
  - We introduced the possibility of dynamically creating comparison tables, so that users can efficiently choose the items they want to compare, specifying which features have to be used for the comparison.
- Experimental work carried out in other application domains has clearly identified the users’ need to feel that they are always able to control the interaction and switch off the system’s initiatives, if undesired (Miller, 2000). The results of our experiments have identified these needs, as well. In order to enable the user to bypass the system’s assistance, we added a “search by key” button, which enables the user to directly access the page presenting a specific item, without entering the Web catalog in the customized navigation mode. This functionality is not the final solution to the problem, but certainly represents a first step toward balancing the system and user’s control of the interaction.

7.2. Second test phase

7.2.1. Organization of the test

We are currently carrying on a second testing phase, which involves a new pool of users, selected from heterogeneous user groups, but sharing a certain familiarity with Web navigation. These people are using an enhanced version of SETA which corresponds to the one described in this paper, but has one additional functionality: the description of the features of an item is introduced by a prologue summarizing the properties of the item. For instance, the “Facile” answering machine is currently introduced by the following sentence:

“Facile is an economical and medium quality item; it has a low technological level and is very easy to use. In particular:”.

In the page, this sentence is followed by the same description of the features offered by the item as the one shown in Figure 14.

The main issues identified in this second test phase refer to experiments made with few users (currently, 20): we have selected people belonging to two very different user classes to check the reactions to the system’s personalization strategies in two extreme cases (groups A and B below). Other experiments will be carried on in the near future, to collect information about the behavior of a more relevant number of users and also to cover other user classes. Although a generalization of the results of our experiments is premature, they suggest several interesting aspects related to the layout of the interface, the quality of the descriptions and the interaction style. In the following, we will report the most interesting ones. The profiles of the two user groups are the following:

---

15 In these experiments, we have switched off the dynamic user modeling functionality to maintain static user models and verify the impact of stereotypical information in the prediction of the users’ features and preferences.
A) People around (or over) 60, typically retired, characterized by a low education level. We asked these users to browse the Web store searching for a phone for their home.

B) Young people (under 40), with high education level and quite dynamic and challenging jobs; e.g., working in a bank, in the university, in a company, and so forth. We asked these people to browse the catalog searching for a fax for their office.

The stereotypical predictions for these two groups can be summarized as follows:

A) The typical user of this group has a low domain expertise, technical interest and receptivity. She is interested in simple and economical products, with medium quality and low technology level. Moreover, she strongly prefers easy to use and economical products.

B) The typical user of this group is characterized by a high expertise, technical interest and receptivity. She likes highly technological and trendy products, considering the quality a very important property and the cost a relatively important one.

We tried to validate the use of stereotypical information by comparing the predicted user features and preferences with the evidence collected by monitoring the behavior of such users while they interacted with the system. During the experiments, we let the users free to browse the catalog, with the only requirement that, before closing the session, they had to select the item (phone / fax, depending on the user group) which they would have purchased in a real case. Sometimes, we asked them why they performed certain actions and whether they needed any functionalities which the system does not offer. We also asked them to comment on the type and amount of information about products provided by the system, on the quality of the linguistic descriptions and on the appropriateness of the system’s decisions, as far as the presentation order of the features of items is concerned.

7.2.2. Results of the experiments with users belonging to group A
Most users of this group were only acquainted with the simplest telecommunication products, such as the phones, and did not know the most technical features; so, their domain expertise could be approximated to a low value. Their receptivity, which we estimated by considering the number of pages they visited and the number of links they explored to get more information about products, was medium in some cases and low in most of the others. As far as their navigation behavior is concerned, we noticed the following things:
– Typically, these people only followed the path leading to the pages describing the phone items, inspecting such pages one after the other, until they found the item which they liked. Moreover, they typically did not click on buttons to get technical details, or to create comparison tables, unless for comparing prices. We interpreted this behavior as avoidance of any distracting actions. These people relied heavily on the system’s capability to tailor the descriptions to their own knowledge and interests; they also appreciated the conciseness of the descriptions of items and the fact that only the most relevant information was displayed, while more detailed information was optional and only attainable on demand, by following the “more information” link.

– The users did not read the information about the active navigation context(s) displayed in the leftmost portion of the pages; when asked why, they explained that they did not need such information at all. Clearly, this comes as a consequence of their very limited browsing activity; in fact, since they navigated the catalog along a single path (looking for one item targeted to a single beneficiary), all the information needed to inspect the various available items was contained in the central and rightmost portions of the pages.

– These people carefully read the prologue describing the properties of the items and strongly relied on such descriptions to select items and compare them to one another. When asked why, they confirmed that the description of properties is very useful, because it helps to evaluate each item without analyzing its technical details, or other complex features. Notice also that most users did not want to compare technical features offered by the items; their selections were only based on the pictures, prices and descriptions produced by the system in the pages presenting the various items.

7.2.3. Results of the experiments with users belonging to group B
The users of this group had good expertise in the telecommunication domain. In particular, as far as faxes are concerned, some of them had high technical expertise and knew the technical features of the items very well. Thus, their domain expertise could be approximated to a high value. Their receptivity was high, as well, and they displayed a marked technical interest. Their navigation behavior can be described as follows:

– They browsed the catalog in a very active way, clicking on buttons (technical details, comparison tables, etc.) and following several
paths in the hypertext. Of course, they maintained a single navigation context, because they were only asked to search for a fax. However, some of them also explored the pages describing other product categories, such as fax-phones.

- They used the information about the active contexts displayed in the leftmost portion of the page. In particular, they confirmed that, although they were carrying on a single active navigation path (so that they did not need to switch from one navigation path to the others) the contextual information was very useful as a summary of their navigation, which provided the links to go back and forth to the initially selected product.

- They tended to ignore the description of the properties of items, only focusing on the individual features. When asked whether they found the description of the properties helpful, they answered that they generally did not trust qualitative descriptions. In particular, they explained that they did not need such information from the system, because they were able to autonomously infer it from the description of the features characterizing the item. Finally, they admitted that, if they had to reason about products which they were not familiar with (e.g., switchboards), such descriptions would have helped them in their selections.

- They selected the goods by checking their features in a comparative way: given the set of features which they considered important (e.g., the price and certain technical characteristics), they analyzed the various items to see which ones offered such features and progressively excluded items, as soon as they found other devices with the same features but other superior aspects.

When the users were considering a small number of features, they compared the alternative items by reading only the related presentation pages. Instead, they created comparison tables when they wanted to check many features at the same time.

7.2.4. Other results

In the following, we describe some behavior types which were common to users belonging to both user groups.

- The “drag & drop” page for the selection of products that the customer would like to inspect and the specification of the beneficiaries of the goods is much more effective than the first form-based solution, but it is still a bit problematic. In particular, none of the users had problems with the “drag & drop” interaction modality...
per se; the users liked that type of interaction for its simplicity and they considered the “home”, “office” and “unspecified” destinations intuitive. However, some users were confused about which icons had to be dragged. Moreover, referring to Figure 2, some users tried to drag the product icons onto the destination icons showed in the upper part of the page (where the system describes the actions to be performed), while ignoring those located in the lower portion of the page. As a first solution to this problem we added to the selection page a “More info” button which starts a short animated demonstration showing the selection of a product for a destination use. Then, we will redesign the ambiguous part of the page to address the confusion introduced by the duplicated icons.

– Almost all the users exploited the shopping cart as a temporary memory where they could store the interesting items which they were deciding whether to purchase or not. They put alternative items into the cart in order to subsequently choose the preferred one. The users appreciated very much the fact that the shopping cart displays the selected items by specifying their beneficiary and destination use, because this type of presentation is very clear. However, using the shopping cart as a buffer, they would have liked to see also the pictures of the items.

– As far as the selection of items is concerned, the results were quite good: most users, after having analyzed several alternatives, selected one of the first goods in the sorted list presented by the system.\footnote{For instance, out of 14 available phones, most users of group A selected one of the first three items and only a couple of users selected the sixth one.} One interesting fact is that, typically, they did not check the complete list of items because, after a while, the presented solutions did not satisfy them any more. In order to check the ranking power of the system, after the users selected the item to purchase, we asked them to examine the items again and to consider also the goods which they had initially skipped. Basically, all of them admitted that, after a certain number of items, the overall acceptability of the other presented solutions degraded and that the last part of the presented list contained the “worst” items, according to their evaluation criteria.

– Some users complained that they had difficulties in viewing the pictures of the items and reading the descriptions of the features. Moreover, most users would have preferred that the buttons to create comparison tables, to view the whole list of items and to check
the shopping cart content were displayed at the top of the pages, in order to attract the user's attention in a more effective way. We will revise the layout packages to satisfy such requirements.

7.3. Other application domains

Another type of test is coming from some parallel initiatives carried out together with the other people involved in the SETA project: we have developed other prototypes, based on the same architecture, which provide different services. For instance, we have developed a small system working as a personalized news server on the Web (Ardissono et al., 1999b), which exploits user modeling and personalization strategies to tailor the detail level in the presentation of news to the users' domain expertise and interests.

Moreover, a further instance of the SETA architecture, presenting services in the banking area has been developed: this system tailors the suggestion of investment funds to the individual user, taking into account factors such as her disposition for “risking”, her family situation, and so forth.

Finally, a very small prototype based on the same architecture guides the user through a knowledge base containing information about cultural heritage in the area around Torino.

7.4. Comments

In the previous sections, we have already mentioned some possible improvements to the system. A further important revision concerns the definition of the layout packages: currently they are used to customize the background and fonts of the catalog pages, but they could be exploited to structure the catalog pages in a different ways. For instance, some components, such as the contextual information (leftmost portion of the page) and the prologue describing the properties of items, could be optional in the various page layouts. In our future work, we will redesign the layout packages taking into account the evidence collected in these experiments and considering usability criteria suggested in related experiments on hypermedia systems, such as those described in (Specht and Kobsa, 1999).

Moreover, the feedback collected in our experiments will be used to revise the stereotypical knowledge currently exploited in the telecommunication prototype, in order to possibly improve the information used to predict the initial features and preferences of first-time customers.
As far as the development of prototypes instantiated in other application domains is concerned, the most valuable feedback certainly concerns the improvement of the SETA architecture; although these prototypes are simpler than the telecommunication one, the experience gained in their design and development has enabled us to improve the flexibility and configurability of SETA, in order to enhance its applicability in other domains. In particular, in the adaptive news server we have introduced a personalized selection of advertisements. Instead, the experiments in the banking domain have been particularly useful to the identification of different types of dialog with the user, which cannot be based on the presence of a well-defined product taxonomy, but must be mainly based on a negotiation over product features, aimed at identifying the specific item satisfying the user's needs.

While we have not yet integrated these functionalities in SETA, we will do that in our future work.

8. Conclusions

We have described the user modeling and personalization techniques exploited in SETA, a shell supporting the construction of adaptive Web stores. The system demonstrates how these types of techniques can be applied in the electronic sales area, where two goals have to be merged: on the one hand, the interaction should be user-friendly and personalized, and this task requires the application of techniques developed in the user modeling, knowledge representation and human-computer interaction research areas. On the other hand, the Web stores are real-world applications; so, these methods have to be embedded in robust and usable prototypes, imposing in many aspects a pragmatic approach to the development of systems.

Our system maintains a detailed user model where information about the user's preferences and characteristics is stored. This model is initialized by means of stereotypical information and is updated during the interaction on the basis of the user's behavior. The user model is exploited to dynamically generate Web pages tailored to the user: the system customizes the description of products, varying their length, terminology, and graphical appearance on the basis of several features of the direct user, such as expertise, interests and receptivity. Moreover, the system maintains a model associated with each person the user is selecting goods for; so, it can support the user's selections by suggesting the items most suited to their beneficiary, who might have different preferences with respect to those of the direct user.
SETA ranks items following a content-based approach, in contrast to the clique-based one exploited in many other recommender systems, because the former approach is superior to the latter in the evaluation of new items; in fact, they can be matched on the user’s preferences as soon as their properties are specified, without requiring any explicit relevance feedback from previous customers (Karunanithi and Alspector, 1997; Cotter and Smyth, 2000). At the same time, the exploitation of stereotypical information about the preferences of relevant customer classes supports the attribution of preferences to first-time customers immediately after they connect to the Web store.

In our future work, we plan to investigate the benefits of embedding implicit relevance feedback, obtained by analyzing the properties of the selected items, to acquire information about which products are purchased by the various customer groups; e.g., see (Greening, 2000) and (Net Perceptions). As described in (Fu et al., 2000), such feedback can be obtained without asking users to rank items, by observing their behavior in an unobtrusive way and applying data mining techniques to discover hidden information from their navigation history. This information, related to the individual user, is already used in SETA to revise the user model; however, it could also be used to revise the description of the stereotypical classes and to acquire information useful for customizing advertisements and special offers.

9. Acknowledgments

This work has been developed in the project “Servizi Telematici Adattativi” (http://www.di.unito.it/~seta), carried out at the Dipartimento di Informatica of the University of Torino within the national initiative “Cantieri Multimediali”, granted by Telecom Italia.

We are grateful to L. Console, L. Lesmo, C. Simone and P. Torasso for having contributed to this work with suggestions and discussions; moreover, we want to thank Giovanna Petrone, Roberta Meo, Cristina Barbero and Marino Segnan, who have developed the SETA system together with us.

Finally, we want to thank the anonymous reviewers, who have helped us to extend this paper and improve its clarity with their detailed and fruitful comments.

References


Ardissono, L. and A. Goy: 1999, ‘Tailoring the interaction with users in electronic shops’. In: *Proc. 7th Int. Conf. on User Modeling*. Banff, Canada, pp. 35–44.


