Tailoring the Interaction With Users in Electronic Shops[⋆]

Liliana Ardissono and Anna Goy

Dipartimento di Informatica, Università di Torino
C.so 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 shell supporting the construction of adaptive Web stores which customize the interactions with users, suggesting the items best fitting their needs, and adapting 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 the user's receptivity, expertise and interests. Moreover, it maintains a profile associated to each person the goods are selected for, to provide multiple criteria for the selection of items, tailored to the beneficiaries' preferences.

1 Introduction

With the expansion of Internet, tools have been built to help vendors to set up Web stores, building the store databases and managing the order processing and payment transactions. These tools typically do not focus on issues like the personalization of the interaction with the customers. However, Web stores are characterized by two main features:

- Since they are accessed by heterogeneous users, they should satisfy different preferences in the suggestion of goods; this ability requires filtering capabilities, to identify the items suited to the specific customer. So, these stores are close to the applications in the information filtering area, where several recommender systems have been developed to exploit user models in the personalized selection of items (e.g., Popp and Lödel, 1996, Karunanithi and Alspector, 1997, Raskutti et al., 1997).
- Since they are hypermedia systems, they should meet the users' interaction needs. Benyon (1993) explains that users differ in many parameters, like status, expertise and preferences, which should be taken into account to improve the usability of systems. For instance, to increase the flexibility of the interactions, the product descriptions should be tailored to the users' expertise and interests (e.g., Milosavljevic and Oberlander, 1998), and the users' preferences on interaction media should be accounted for (e.g., Joerding, 1998, Fink et al., 1997).

[⋆] This work is developed in the project "Servizi Telematici Adattativi" (http://www.di.unito.it/˜ seta), carried on 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 fruitful discussions. Many thanks to G. Petrone and C. Barbero who have developed the SETA system together with us. **This paper is going to appear on the proceedings of the 7th International Conference on User Modeling (UM-97), Banff, MA, 1999**

The issue of tailoring information to the user has been deeply analyzed in the flexible hypermedia research, where a major distinction was made between personalizing the navigation task and the description of the information items to be presented (Brusilovsky, 1996). Some researchers, like Calvi (1997), have focused on the dynamic adaptation of the hypertextual structure to users with different backgrounds. Others, like Milosavljevic et al. (1996) and Hirst et al. (1997), have focused on the dynamic generation of text tailored to the system's user. However, an analysis of the electronic sales reveals other issues to be faced in an on-line 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, to remember which items she has analyzed, and other data useful to identify her real needs. On top of this, the product description should be planned to highlight the information most important to the user, so that she can easily compare products and decide which one to buy; finally, the properties having the greatest impact on her might be highlighted to convince her to buy the items (e.g., Jameson et al., 1995).

Various techniques 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 (e.g., Ackerman et al., 1997, ACM, 1997). We believe that, while those techniques are suited to deal with large-scale applications, such as information retrieval on the Web, virtual stores can benefit from the presence of motivated users[1] and a constrained information space. Products can be carefully defined and classified, so that the search task is performed in the presence of significant information about the hyperspace nodes. For these reasons, we believe that knowledge-intensive approaches, where detailed user profiles are built and items and descriptions are selected on the basis of a deep evaluation of the user's needs, are promising in the development of Web stores.

In this paper, we describe the user modeling and personalization techniques adopted in SETA, a tool for building adaptive Web stores which tailor the interactions to their customers' features, possibly suggesting the items which best fit their preferences. SETA is a virtual store shell and can be instantiated on a new sales domain by configuring the knowledge bases and the databases containing the domain-dependent information; our current prototype works on the telecommunication domain (selling phones, faxes and similar products). In the development of this Web store, we have focused on issues related to the adaptive provision of information, leaving apart the adaptability of the interface: we have considered the customization of the presentation style, the selection of the product information to provide, and the selection of items to recommend. In the following sections, we describe the portion of SETA handling the personalization of the interaction; Ardissono et al. (1999b) provides a description of the overall system architecture.

2 Management of the User Models

The user models are handled by a User Modeling Component (UMC). During a session, the direct user may choose items for herself and for other people. The UMC builds a direct user model to tailor the product presentation and the selection of goods to her needs; moreover, it creates a user model for each third person for whom she selects goods: the beneficiaries' models are used to suggest the goods suited to the persons they are selected for, if different from the direct user.

[1] They are looking for items to buy and, if they want to be helped, we can assume that they are willing to cooperate with the system and provide information about themselves.

The user models are initialized by retrieving the users' records from a Users DB. New customers are asked to fill in a form where the classification information is asked; then, the UMC initializes their models by exploiting stereotypical information.[2] The UMC retrieves the domain-dependent knowledge about users from a Stereotype KB, which contains a hierarchical taxonomy of stereotypes clustering the properties of homogeneous customer groups. The user features to be modeled in a specific sales domain are defined at the store configuration time: when a new virtual store is created, the Stereotype KB can be defined using a dedicated configuration tool, which also generates a template, used to create the user models during the interactions. The Stereotype KB may contain the description of multiple market segmentations, called "stereotype families". A family describes the features characteristic of the customer classes (stereotypes) belonging to it. The stereotype families represent orthogonal segmentations of the population and describe customers under different viewpoints; for example, for our prototype we have defined three families concerning the users' domain expertise, life style and destination use of the items to purchase (we distinguish home and business use, involving different preferences towards product properties). The stereotype families exploit (partially overlapping) classificatory data, but make predictions on different user preferences and personality traits. Each user is classified independently in each family and the stereotypical predictions are merged to initialize her user model.

The representation formalism of the user models includes a descriptive and a predictive part. The *descriptive part* stores the data explicitly asked to the user to classify her in the stereotypical user classes, and the information about the user's needs. The *predictive part* contains the information about the user's features inferred by the system, by exploiting stereotypical information.

In our prototype, the *classification data* are the user's personal data (age, gender, job, education level); however, when configuring the Stereotype KB of a new store, different user features might be selected for that sales domain (e.g., preferred colors and sports). In general, user *needs* may regard very different product features. However, as a first step to enable their recognition, we have associated needs only to the the functionalities offered by products (e.g., the transmission of voice is a possible user need and is the main functionality offered by phones). As a simplifying assumption, we currently suppose that the user is aware of her own high-level needs and can identify the generic product categories satisfying them (e.g., she knows that she needs a phone, although she may need help to select the specific item to purchase); so, needs are information that the system explicitly elicits from the user at the beginning of the interaction.

The *predictive part* of a user model concerns personality traits and preferences. The *personality traits* are used to tailor the descriptions to the user and include the receptivity (i.e. amount of information she is able to acquire), domain expertise, technical interest and interest for aesthetic information. The *preferences* represent the user's attitudes towards product properties (e.g., ease of use) and are used to suggest the items suited to her.[3] Following the formalism introduced in Torasso and Console (1989), personality traits and preferences are represented as parameters structured as follows (the personality traits have no importance slot):

Parameter Name; Importance;

Values: <Value, Probability> pairs;

[2] Recent developments of our system include the use of dynamic user modeling techniques, not described here, to update the user models on the basis of the user's behavior and product selections.

[3] We consider preferences as more general than the functionalities needed by the user. For instance, the user may have a generic preference for easy-to-use products.

The *Importance* slot takes values in the range [0..1] and represents the importance of the property to the user. Each <*Value, Probability*> pair contains a linguistic value that the parameter can take, and the probability that the user prefers that value for the product property. The probabilities associated to the linguistic values of a parameter are normalized and sum up to 1. For example, a user's preference towards the products "ease of use" might be:

Ease of use; Importance: 1;

Values: <low, 0.0>, <medium, 0.3>, <high, 0.7>.

This means that (the system believes that) the user considers the ease of use extremely important; moreover, she prefers very easy to use products with probability 0.7 (*high:* 0.7), average complexity products with probability 0.3, and she does not prefer complex products (*low:* 0.0).

2.1 Classification of Users

A stereotype *classification part* contains the data used to evaluate the probability that the user belongs to the represented customer class; we call this number *matching degree*, since it is calculated in a non-strictly probabilistic way (see below).[4] Each classification datum has an *Importance* slot and a list of <*Value, Probability*> pairs. The first slot describes the importance of the datum to the classification (i.e. how strongly it influences the evaluation of the user's degree of matching with the stereotype). The <*Value, Probability*> pairs represent the conditional probabilities that the user belongs to the stereotype, given the linguistic value of the datum.

To classify the user in a stereotype S, the UMC matches her data on the stereotype classification data; then, it combines the results, to obtain her matching degree with S. Let's consider two classification data, A and B (e.g., age and job), their importance, Imp_A and Imp_B, and their <*Value, Probability*> pairs: $\{<a_1, p_{a_1}>, ..., <a_m, p_{a_m}>\}$, $\{<b_1, p_{b_1}>, ..., <b_n, p_{b_n}>\}$. Suppose that the values for a user are a_i and b_j (e.g., she is a young teacher). The UMC first filters the conditional probabilities of the classification data to take into account their importance: if a datum is important, it should fully influence the task; if it is irrelevant, it should not (the neutral value is 1, see below). We used the following filtering function to modify the probability provided by a single classification datum (datum A), and obtain its matching score ($score_A$):

$$score_A = Imp_A * p_{a_i} + (1 - Imp_A). \qquad \textbf{(i)}$$

This formula raises the contribution to the classification provided by less important data, while it leaves that of important data unchanged: e.g., the score of the totally irrelevant data is 1, independent of the value of p_{a_i}; instead, if Imp_A = 1, then $score_A = p_{a_i}$.

The user's degree of matching with S is evaluated by applying the following function (Lesmo et al., 1985) to the scores computed for the classification data:

$$Match(score_A, score_B) = score_A * score_B / (score_A + score_B - score_A * score_B) \quad \textbf{(ii)}$$

For instance, if $score_A = 0.6$, $score_B = 0.9$, and A and B were the only classificatory data of S, the user's degree of matching with S would be 0.562. Function (ii) evaluates the fuzzy AND of the classification data, by combining their scores in a multiplicative way, and can be incrementally evaluated.[5]

[4] The main deviation from probability theory is due to the use of the importance of data: this factor is exploited to provide the store designer with an intuitive mechanism to tune the influence of data on the classification process, without requiring that she masters probabilistic concepts.

[5] Other functions have been used to compute the fuzzy AND: e.g., the product of the evidence of the ANDed conditions, or their minimum evidence. Function **(ii)** takes greater values than the product, which

2.2 Prediction of User Features

A stereotype *predictive part* makes predictions on the user's personality traits and preferences: each prediction has an *Importance* slot, representing an estimate of how important the preference is to the user, and a list of $<$*Value, Probability*$>$ pairs (predictions on personality traits have no importance). These are the conditional probabilities of the linguistic values for the datum, given that the user belongs to the stereotype.

The UMC initializes each personality trait and preference in the user model by merging the stereotypical predictions on its linguistic values (stereotypes in different families produce non overlapping predictions). The merge is obtained as the weighted sum of the probabilities suggested by each stereotype (the weights are the user's degree of matching with the stereotype). For instance, consider the "Domain Expertise" family and the preference towards the products "ease of use". If the user classification has produced the following matching degrees: Novice (0.7), Intermediate (0.2), Expert (0.1), and the stereotypes predict:

Novice: Importance: 1; <low, 0>, <medium, 0.3>, <high, 0.7>;

Intermediate: Importance: 0.8; <low, 0.1>, <medium, 0.6>, <high, 0.3>;

Expert: Importance: 0.5; <low, 0.3>, <medium, 0.4>, <high, 0.3>;

then the prediction on the importance of the preference, and that on its *"low"* value are:

Importance: $0.7 * 1 + 0.2 * 1 + 0.1 * 0.5 = 0.95.$; *low:* $0.7 * 0 + 0.2 * 0.1 + 0.1 * 0.3 = 0.05.$[6]

The weighted sum allows the best matching stereotypes to influence the predictions in a stronger way; moreover, if different stereotypes make similar predictions, they enforce each other.

3 Suggestion of Goods to the User

The information about the items sold in the store is maintained in a Products DB, where they are grouped in product classes (e.g., phones, faxes). When the items to suggest have to be displayed, a module, the Product Extractor, retrieves from the Products DB the records of the items matching the user's query (e.g., the phone models available in the store). Then, it ranks and sorts them, depending on how close they match the preferences in the beneficiary's user model. In this way, the items can be presented showing the most interesting ones first. The matching between the beneficiary's preferences and the properties of items is possible because their records contain information about features and *properties* (e.g., ease of use), and the properties correspond to the preferences specified in the user models. An item record stores, for each property, the linguistic value fitting the item; e.g., the "Super Slim" phone is described as follows:

Features:
code: SuperSlim;
price: LIT. 59000;
color: black;

Properties:
quality: high;
ease of use: medium;

is influenced by the number of conjuncts to evaluate. Moreover, **(ii)** is more precise than the minimum, which takes into account the worst-matching feature and ignores the other classification conditions.

[6] After these assignments, the values are normalized so that they sum up to 1 and can be interpreted as probabilities again.

The Product Extractor gets from the UMC the preferences of the beneficiary of the goods. Then, it ranks each item by evaluating how close its properties match such preferences. To this extent, the Product Extractor exploits the same formulae described in Section 2.1 for the classification of users: an item suited to the user should match all her important preferences, possibly ignoring irrelevant mismatching properties; moreover, if an item mismatches an extremely important user preference, it should not be recommended at all. So, the user preferences are used as classification conditions for evaluating the overall matching degree of the item and are combined, again, in a fuzzy AND. First, given the property values of the item and the importance of such properties to the beneficiary (importance of the preference in the user model), the formula **(i)** is applied to neutralize the impact of irrelevant (mismatching) properties (raising their contribution to 1). Then, the filtered scores are inputed to the formula **(ii)** to obtain the overall score of the item.

4 Personalization of the Presentation

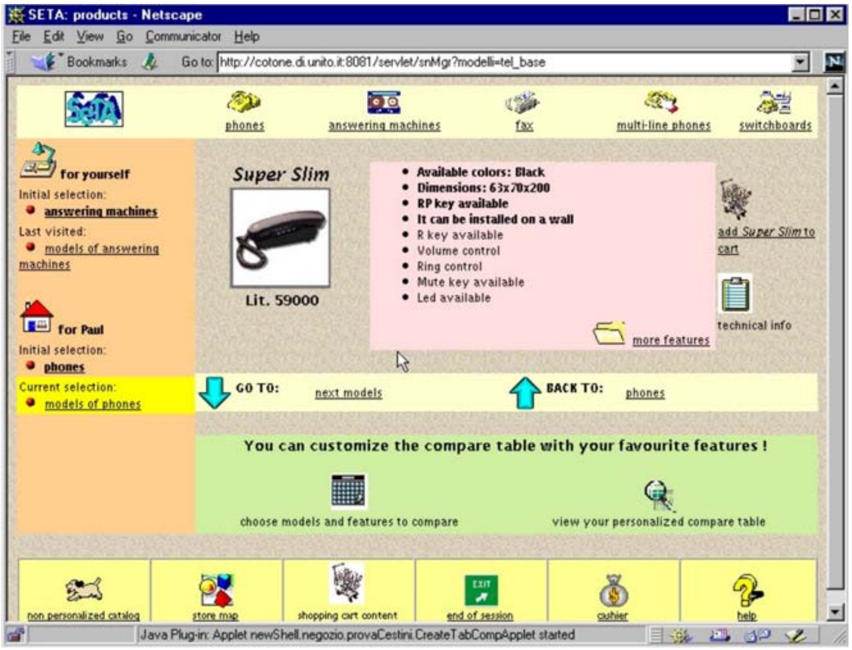
In SETA, a Personalization Agent dynamically generates the HTML code for the hypertextual pages to display during the interaction with the user. This module applies some personalization strategies to customize the pages, varying their contents and layout on the basis of the user's features (stored in the user model) and of the characteristics of the information to provide. The information sources used by the Personalization Agent include:

- The type of page to produce; e.g., a form, a page presenting a product / item, and so forth.
- The product / item to be presented, and the information about it. This information is stored in the Product Taxonomy (a knowledge base keeping the conceptual representation of the product classes), and in the Products DB, containing the specific data about items.
- The internal structure of the catalog, used to decide which hypertextual links to include (e.g., links to more specific products). This structure corresponds to the organization of products in the Product Taxonomy.
- The interaction context keeping the user's selections and a memory of which products and pages she has already seen.[7]
- The user's personality traits, used to select the information to provide and the technicality of the descriptions.
- A set of personalization rules, used to decide how to customize the pages on the basis of the contents of the user model.

The dynamic generation of pages and the fact that their contents are retrieved from declarative knowledge sources represent a step forward with respect to previous approaches, like that of Popp and Lödel (1996), where multiple static versions of the catalog exist and the system selects the one to display. In fact, in our approach, the pages may be adapted to the user's needs at the granularity level of the product features; moreover, the configurability of the system is enhanced.

The Personalization Agent generates different pages (choosing different layouts, colors, etc.) depending on the user's "backgrounds" (such as age and job); moreover, it produces different product descriptions, tailoring the selection of content to the user's interests and receptivity, and basing the selection of the linguistic form on her domain expertise.

[7] This context is maintained by the Dialog Manager, a component of the system we will not describe.



- **Available colors: Black**
- **Dimensions: 63x70x200**
- **RP key available**
- **It can be installed on a wall**
- R key available
- Volume control
- Ring control
- Mute key available
- Led available

You can customize the compare table with your favourite features !

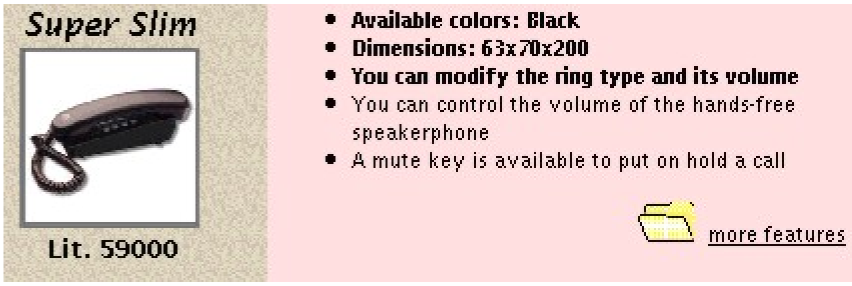
Java Plug-in: Applet newShell.negozio.provaCestini.CreateTabCompApplet started

Figure 1. A page generated by our system presenting the “Super Slim” phone to an expert user.

Figure 1 shows a presentation page describing the “Super Slim” phone. In the leftmost portion of the page the system displays the user’s selections and enables her to switch among them. The central area describes the functionalities and features offered by the product and is tailored to the user’s interests, expertise level (showing detailed and technical descriptions) and receptivity (many features of phones are described). Below this area, there is the bar containing the links to the more specific/generic products in the Product Taxonomy View (“GO TO”, “BACK TO”). The topmost bar provides the links to the main product categories available in the store, while the bar at the bottom of the page contains the general control buttons (e.g., “EXIT” button and link to the site map). Figure 2 focuses on the description of the same item, but it has been produced for a non-expert user; as it can be noticed, in this case fewer features are listed; moreover, the sentences displayed are simpler and more intuitive than those in Figure 1.

The Personalization Agent exploits a set of production rules to customize the portion of the hypertextual pages describing the features of products and items: it first selects the features to describe, by ranking their relevance on the basis of the user’s interests and their intrinsic importance to the product description. Then, it plans the overall appearance of the Web page. Since its decisions have to be taken on the basis of criteria concerning different factors (e.g., user characteristics and information about products, etc.) a separate set of rules is applied to take each decision and the final result is obtained by combining the various contributions.

1) A first set of rules rates each product feature on the basis of the user’s interests: for instance, a rule takes into account the user’s technical interest (TI) and assigns the following scores to all



Super Slim

Lit. 59000

- **Available colors: Black**
- **Dimensions: 63x70x200**
- **You can modify the ring type and its volume**
- You can control the volume of the hands-free speakerphone
- A mute key is available to put on hold a call

more features

Figure 2. Detail of a presentation page describing the "Super Slim" phone, tailored to a non-expert user.

the technical features: 1 for TI=*low*, 2 for TI=*medium*, 3 for TI=*high*. Similar rules exist for the aesthetic interest, and so forth.

2) A rule ranks the features on the basis of their objective importance to the product description: this information is part of the description of products and is stored in the Product Taxonomy (it can take the following values: 1 for *low*, 2 for *medium*, 3 for *high* importance).

The estimates produced by the rules are weighted, to tune their influence to the evaluation of the overall score of the features: in our prototype, the weights are set to 1; however, the store designer may appropriately configure them, depending on whether the user's interests, or the objective importance of features should influence the evaluation most (she can also set them to 0, to neutralize the impact of the information to the evaluation task). This aspect is especially relevant in electronic sales applications, where there are typically data about products that have to be communicated to the customer, independently of her interests. The application of the rules in 1) and 2) produces a list of product features, partitioned in groups of equally scored elements. At this point, those to show to the user are selected.

3) A rule sets the number of information items that can be displayed in the same page, on the basis of the user's receptivity. This number enables the Personalization Agent to identify the portion of the feature list to show; the rest of the list is made available by introducing a "more information" link. Given the total number of features in the ordered list (n), and the user's receptivity (*low*, *medium*, or *high*, represented by r), the rule works as follows:

<u>If</u> $Receptivity = r$ <u>then</u> number of features to show = $x \pm \delta$;

In the formula, the value of x is a function of r and is retrieved from a correspondence table defined on the basis of the results of psychological studies about cognitive load. The δ variable represents a "tolerance factor", used to cut the feature list in a flexible way: if the value of *x* falls inside a group of features having the same score, an arbitrary decision should be taken to identify the features to show and those to exclude; δ enables the system to extend or restrict the number of features to show. 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 then δ, then the group containing these features is hidden; otherwise, it is shown.

4) Finally, a rule selects the appropriate linguistic form for each feature, on the basis of the user's domain expertise. The linguistic descriptions are generated by exploiting templates, whose slots are filled by linguistic expressions corresponding to the feature values extracted from the records of items in the Products DB. A difficulty level is associated to each template (basically

depending on its technicality) and matched with the user's domain expertise level. In this way, the appropriate descriptions can be selected for the user: if the user's expertise is low, very simple descriptions, suited to a novice user, are used; if her expertise is high, the Personalization Agent includes concise and technical descriptions.

The final HTML code is generated by choosing the appropriate display settings (the font size, the color, and other special markers), on the basis of the scores associated to the features (e.g., some features in the Figures 1 and 2 are bold faced). As shown in Figure 1, the system also offers facilities to ask for more information about goods; for instance, the user can create comparative tables on the fly, by selecting the goods to examine and the features (and properties) which she would like to consider, therefore avoiding the examination of huge, precompiled structures, which compare all the items of a product category with respect to all their features.

5 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 advanced AI technologies can be applied in the electronic sales area, where two goals have to be merged: on one side the interaction needs to 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, on-line stores are real-world applications; so, the AI methods have to be embedded into robust and usable prototypes, which impose 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 needs is stored. This model is initialized by means of stereotypical information and it 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 the (direct) user's expertise, interests and receptivity. Moreover, the system maintains a model associated to each person the user is selecting goods for; so, the user's selections may be supported by suggesting the items most suited to their beneficiary, taking into account that the beneficiary's needs may be different from those of the direct user. The system can be configured on specific sales domains and the knowledge used to describe products and handle the user models can be configured by the store designer, by means of appropriate configuration tools.

For the moment, we have tested our prototype and tuned it on the basis of the suggestions collected from a restricted number of users (including psychologists and computer science experts), who have helped us to improve the interface and the interaction mode. A field trial is necessary to test the system on real users. As future work, we would like to increase the system's initiative: e.g., during an interaction, the system should interact more frequently with the user, asking her whether she would prefer items with slightly different characteristics (Linden et al., 1997); moreover, the system should promptly react to the user's behavior, exploiting her interests to promote specific products (Greer et al., 1996). We feel that reactive planning approaches, as that of de Carolis (1998), could be very effective to this task.

Our system is a Three Tier Application Architecture written in Java and based on the JavaSoft Java Web Server 1.1; see (Ardissono et al., 1999b, Ardissono et al., 1999a) for details.

References

Ackerman, M., Billsus, D., Gaffney, S., Hettich, S., Khoo, G., Kim, D., Klefstad, R., Lowe, C., Ludeman, A., Muramatzu, J., Omori, K., Pazzani, M., Semler, D., Starr, B., and Yap, P. (1997). Learning probabilistic user profiles. *AI Magazine* Summer:47–55.

ACM (1997). Recommender systems. *Communications of the ACM* 40(3).

Ardissono, L., Barbero, C., Goy, A., and Petrone, G. (1999a). An agent architecture for personalized web stores. To appear on *Proceedings of the Third International Conference on Autonomous Agents*.

Ardissono, L., Goy, A., Meo, R., Petrone, G., Console, L., Lesmo, L., Simone, C., and Torasso, P. (1999b). A configurable system for the construction of adaptive virtual stores. To appear on the *World Wide Web journal*.

Benyon, D. (1993). Adaptive systems: a solution to usability problems. *User Modeling and User-Adapted Interaction* 3:65–87.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction* 5(2-3):87–129.

Calvi, L. (1997). Multifunctional (hyper)books: a cognitive perspective (on the user's side). In *Proceedings of the workshop "Adaptive Systems and User Modeling on the World Wide Web*, 23–30.

de Carolis, B. D. (1998). Introducing reactivity in adaptive hypertext generation. In *Proceedings of the Thirteenth European Conference on Artificial Intelligence*.

Fink, J., Kobsa, A., and Nill, A. (1997). Adaptable and adaptive information access for all users, including disabled and the elderly. In *Proceedings of the Sixth Conference on User Modeling*, 171–173.

Greer, J., MacKenzie, M., and Koehn, G. (1996). User models for coercion, persuasion and sales. Research Report 96-1, ARIES Laboratory, Department of Computer Science, University of Saskatchewan.

Hirst, G., DiMarco, C., Hovy, E., and Parsons, K. (1997). Authoring and generating health-education documents that are tailored to the needs of the individual patient. In *Proceedings of the Sixth Conference on User Modeling*, 107–118.

Jameson, A., Shafer, R., Simons, J., and Weis, T. (1995). Adaptive provision of evaluation-oriented information: tasks and techniques. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1886–1893.

Joerding, T. (1998). Intelligent multimedia presentations in the web: Fun without annoyance. In *Proceedings of the Seventh World Wide Web Conference*.

Karunanithi, N., and Alspector, J. (1997). Feature-based and clique-based user models for movie selection: a comparative study. *User Modeling and User-Adapted Interaction* 7:279–304.

Lesmo, L., Saitta, L., and Torasso, P. (1985). Evidence combination in expert systems. *International Journal of Man-Machine Studies* 22:307–326.

Linden, G., Hanks, S., and Lesh, N. (1997). Interactive assessment of user preference models: The automated travel assistant). In *Proceedings of the Sixth Conference on User Modeling*, 67–78.

Milosavljevic, M., and Oberlander, J. (1998). Dynamic hypertext catalogues: Helping users to help themselves. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia*.

Milosavljevic, M., Tulloch, A., and Dale, R. (1996). Text generation in a dynamic hypertext environment. In *Proceedings of the Nineteenth Australasian Computer Science Conference*, 417–426.

Popp, H., and Lödel, D. (1996). Fuzzy techniques and user modeling in sales assistants. *User Modeling and User-Adapted Interaction* 5:349–370.

Raskutti, B., Beitz, A., and Ward, B. (1997). A feature-based approach to recommending selections based on past preferences. *User Modeling and User-Adapted Interaction* 7:179–218.

Torasso, P., and Console, L. (1989). *Diagnostic Problem Solving*. North Oxford Academic.