# Dynamic user modeling in a Web store shell

**Liliana Ardissono and Pietro Torasso** [1]

**Abstract.**

We describe a framework for the dynamic revision of user models in an adaptive Web store shell, which tailors the suggestion of goods, as well as the interaction style, to the characteristics and interests of the individual user. The behavior of the user is unobtrusively monitored, by segmenting it on the basis of the focus spaces explored in the browsing activity, and the actions performed by the user are summarized into abstract facts. These facts are used for revising the user model via the interpretation process performed by a Bayesian Network which relates user features to user behavior.

## 1 INTRODUCTION

Several Web catalogs and recommender customize the suggestion of items (e.g., [9]); however, while users may have different preferences and needs about products, they also differ in their interaction requirements. The exploitation of user models specifying interests, domain expertise, and other similar features is the basis for selecting the type of information to be delivered and the presentation style to be adopted; e.g., [4]. Moreover, unobtrusively observing the user's behavior is essential to revise the user models and react to the changes in her/his attitudes during the interaction; e.g., [7, 15].

This paper describes the analysis of the user behavior and the management of dynamic user models applied in SETA [2] [2], a prototype toolkit for the creation of adaptive Web stores. We will focus on the management of the part of the user model supporting the selection of the information to be presented, its linguistic style and the amount of information to be included in the catalog pages. Our framework relies on a feature-based representation of the user's requirements and on a structured Web store interface, which enables the execution of various actions to get specific information about products (see section 2). The user model is revised by monitoring the actions performed by user during the navigation of the catalog and periodically analyzing them (see section 3): the collected evidence is interpreted by exploiting a Bayesian Network relating user features, such as interests and level of expertise, to her/his behavior.

## 2 OVERVIEW OF SETA

**Representation of products and users**

In SETA, the product classes are organized in a Product Taxonomy, whose roots represent general classes and have subclasses describing more complex products [2]. This taxonomy represents the skeleton of the hypertext forming the Web catalog: e.g., in Figure 2, the top bar of the page lists the main product classes (phones, etc.) and the leftmost portion of the page shows that the user is inspecting the items of the "fax-phones with answering machine" class.

Figure 1 shows an example user model, taken from our prototype instantiated on the telecommunication domain. We focus on the "user features" part, which describes requirements on the catalog layout and characterizes the user's interests and familiarity with products. The user features are represented as parameters $P$ with the form:

$P$: $<low, x>, <medium, y>, <high, z>$

where $x, y, z$ represent a probability distribution for the (*low, medium, high*) linguistic values of $P$. In particular:

a) The general features include the user's *receptivity*, which describes her/his disposition toward the acquisition of large amounts of data: some users get confused if exposed to a lot of information and the interaction with these people can be improved by constraining the content of the Web pages. In contrast, very receptive users can appreciate pages containing detailed information about products; however, the user's receptivity may downgrade during the visit of the catalog, mainly because of tiredness factors; thus, even a very receptive user, after a long interaction, can benefit from synthetic pages.

b) The interests describe the user's attitudes toward *technical, aesthetic* and *functional* information about the main product classes defined at the top level of the Product Taxonomy. These types of information are the basis for classifying the product features: e.g., data like the size and color of items are classified as aesthetic features, while the facilities offered by the products (e.g., phones offer agendas to store phone numbers) are related to functional aspects. For instance, Paul's user model, in Figure 1, predicts a high value for his interest in functional information about phones (with probability 0.7); this means that the system believes that Paul prefers descriptions of

Personal data:
Name: Paul Smith;
Education Level: high_school; ...
User features:
*General features:*
Receptivity: low: 0.5; medium: 0.25; high: 0.25;
...
*Interests:*
Aesthetic Interest about phones: low: 0.3; medium: 0.3; high: 0.4;
Technical Interest about phones: low: 0.1; medium: 0.6; high: 0.3;
Functional Interest about phones: low: 0.1; medium: 0.2; high: 0.7;
Aesthetic Interest about switchboards: low: 0.7; medium: 0.2; high: 0.1;
...
*Expertise:*
Expertise about phones: low: 0.1; medium: 0.2; high: 0.7;
Expertise about switchboards: low: 0.6; medium: 0.3; high: 0.1;
...

**Figure 1.** Portion of an example user model.

[1] Dipartimento di Informatica, Università di Torino, Corso Svizzera 185, I-10154 Torino, Italy, email:{liliana, torasso}@di.unito.it

[2] SETA has been 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", sponsored by Telecom Italia. We thank Roberto Furnari for his controbution to this work.


**Figure 2.** A page describing a fax-phone with answering machine.

phones and of their subclasses focused on functional features.
c) Expertise: a set of *expertise* features represent the user's familiarity with the main product classes of the catalog. As a suitable solution to the trade-off between specificity (e.g., describing the user's expertise on each possible feature of a product) and generality (e.g., defining a degree of expertise on the whole catalog), we maintain a different expertise feature for each main product class.

The introduction of these user features supports various personalization strategies: the amount of data in the pages can be tailored to the user's receptivity; the presentation of goods can be focused on features matching her/his interests; the technicality of language can be tailored to her/his expertise; see [1].

**The Web store interface**

The structure of the Web pages strongly influences the possibility to recognize the user's information needs: if the user is allowed to manifest (either directly or indirectly) the existence of a gap between her/his own needs and those satisfied by the system, then the user behavior can be effectively monitored to revise the user model.

The interface of SETA is based on two types of pages: those describing the functionalities offered by products (e.g., phones) and those presenting the items available for a product class. Figure 2 shows a page presenting the "Scriba Compact 401" fax-phone with answering machine. The system exploits personalization strategies to tailor the content of the catalog pages to the user's information needs; moreover, the pages are highly structured and contain buttons and links which the user can exploit to override the system's decisions, e.g., asking for specific information, for details different from the displayed ones, or for easier descriptions, or even hiding the displayed information. Thus, during the navigation in the catalog, the user may actively interact with the system to obtain the needed information, at the preferred detail and complexity levels. As we will see, keeping track of the links (s)he follows, and those (s)he ignores represents the basis for the revision of the user model.

# 3 DYNAMIC REVISION OF THE USER MODEL

The revision of the user model is organized in three phases: monitoring the user's actions and discovering patterns of actions to obtain a first synthesis of her/his behavior (see Section 3.1); processing the results of such analysis to obtain a contextual synthesis of her/his behavior, where the interaction history is considered, and interpreting the contextual synthesis in terms of the user features (see Section 3.2). This analysis is divided into sub-tasks for efficiency reasons: in fact, our goal is the continuous revision of the user model during the interaction with the customer, so that the system can change personalization strategies in a reactive way. In particular, the generation of a synthesis of the user's behavior enables the system to reason on a compact set of data and avoid uninterpreted log files, suited to off-line analyses; e.g., [11].

We have exploited a production system (JESS [10]) for representing declaratively the relations among the contextual information, the user's actions, and the strategies for processing them.

## 3.1 Monitoring the user's actions

Because of the limits in the length of the paper, we will focus on the analysis of the following types of action performed by the user (see Figure 2):
- **Follow the "more information" link** to get more data about the features of the displayed item (or hide such data, when displayed);
- **View the technical details** of the item ("Technical info" button);
- **Ask for help** on the description of a feature of the item ("help" buttons, each one associated to one feature);
- **Create a comparison table** to compare items with respect to a set of selected features ("You can customize the compare table . . . ").

**Segmenting the history of the interaction**

One critical issue concerns the identification of the action sequences relevant to the analysis of the user behavior. The systems exploiting detailed user models typically adopt acquisition rules to reason about individual events, which reflect the user's attitudes toward specific knowledge items (e.g., denoting that (s)he knows/does not know a concept; e.g., [4]). In contrast, the user models used in SETA address general user features. Thus, only an analysis of sequences of related actions can provide an evidence about changes in such features: e.g., the user's knowledge and technical interest about phones can only be determined given an overview of her/his behavior, as far as phones are concerned. Similarly, the recognition of changes in the user's receptivity requires a general view on the interaction history.[3] However, in order to support a reactive adaptivity of the system to the user behavior, the analysis must be performed at several points during the interaction and the user actions cannot be analyzed, as separate information sequences, at the end of session. Thus, a central issue is the identification of the sequences of actions on which the analysis has to be focused to update the user model. More specifically: we want to separately analyze the actions concerning different product classes, because such actions provide information regarding different user features, e.g., consider her/his interests. Moreover, when analyzing the most recent user behavior, we want to be able

[3] [7] describes a framework for the recognition of changes in the user's cognitive load in a NL dialog system, based on the analysis of the user's individual utterances and on the identification of symptoms such as pauses, repetitions, and so forth. Unfortunately, although some work is being carried on to track the user's mouse activity and other related events [5], the identification of analogous symptoms in a hypertextual environment is far from obvious.

to take into account the whole sequence of related actions, e.g., to recognize actions performed more than once.

We exploit the structure of the Web catalog, as defined by the Product Taxonomy, to perform such contextual analysis of the user's behavior: in particular, we associate to each product class a *focus space* [6] including information about the product class (e.g., "fax-phones with answering machine") and its items (e.g., "Scriba Compact 401"). When the user browses the catalog, (s)he can shift her/his focus of attention, viewing various product classes and, possibly, going back to already visited ones. At each stage of the interaction, the *Current Focus of attention (CF)* points to the active focus space, associated to the product class inspected by the user.

The user can perform several actions in a focus space, without shifting the CF: in fact, while (s)he focuses on a given product class, (s)he can open the pages presenting the available items; moreover, in each page, (s)he can click on buttons to get specific information about the items. We introduce the notion of *local history (LH)* to denote a continuous sequence of actions, without changes in the user's focus of attention: an LH is composed by the user's request to inspect a product class and all the subsequent actions performed by her/him before moving to another product class. A local history identifies a portion of the interaction history which can be exploited to get meaningful information about the user: in fact, it is conceivable that (s)he has uniform interests and domain expertise as far as individual items are concerned, while such features may change from product class to product class. Thus, we perform the contextual analysis of the user behavior by separately considering each LH, as soon as the user exits a focus space. In this way, the user model can be revised periodically during the interaction and, at the same time, contextually related information is available at each step to reason about the user's actions.

**Representation of actions**

We have defined different facts for representing the possible types of action (e.g., clicks on links to view a product, clicks on help buttons, etc.). Each fact describes an action type, its arguments, i.e., the objects on which the action is performed, and other information used to reason about the action. For instance, a click on the help button associated to a feature ($feature_x$) of an item ($item_x$) belonging to a product class ($product_x$) is represented as follows:

(help $feature_x$ $item_x$ $product_x$ $tiredness$ $timeTag$)

where "help" denotes the action type; $tiredness$ is the tiredness factor before the occurrence of the action (described later on) and $timeTag$ is a temporal tag.

A particular occurrence of the action is asserted when the user clicks on the help button for a specific feature of an item; for instance:

(help transmission_speed Megafax fax 0.1 34).

The specification of the arguments of an action is essential to distinguish occurrences of the same action on different objects; e.g., if, later on, the user clicks on the help button of feature "receiving polling", the following fact is asserted:

(help receiving_polling Megafax fax 0.15 42).

The time tag is used for sorting facts on the temporal axis and for distinguishing different instances of the same action; each fact has a progressive tag.

In addition to the facts representing the possible types of user actions, we have defined control facts to declare the entering and closure of each focus space. The control facts are asserted when a user performs an action determining a focus shift: e.g., the following facts are asserted when the user follows a link to view a product class $C'$:

(exitFocus $C$ $tiredness$ $timeTag$)

(enterFocus $C'$ $tiredness$ $timeTag$)

to denote the closure of the previous CF, associated to $C$ ("exitFocus") and the opening of the new CF ("enterFocus").

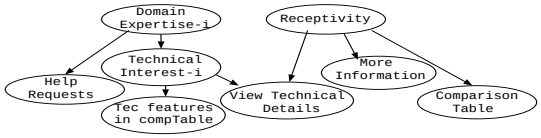


**Figure 3.** Portion of the Bayesian Network of our system.

**Summarizing the admissible actions within a CF**

As explained later on, the analysis of the user behavior is based on the evaluation of her/his "degree of activity" in the types of action (s)he may perform. Thus, for each visited page, the system must keep track of the links and buttons available to the user to ask for specific information, help, and so forth. To this purpose, for each displayed page, the system asserts a set of summarization facts: different facts are associated to the various action types and the time tag is used to recognize the local history to which the action instances belong; e.g., the number ($n$) of help buttons displayed in a page presenting an item ($item_x$) of a product ($product_x$) is represented as follows:

(helpButtons $n$ $item_x$ $product_x$ $timeTag$)

**Processing the facts describing the user's actions**

The closure of a local history $LH$ triggers the activation of a set of production rules which process the facts in the working memory related to $LH$ to evaluate the user's degree of activity in $LH$. For each action type $A$, an *occurrence rate* is evaluated as follows:

$rate_A = performedActions_A \,/\, availableActions_A$;

- $availableActions_A$ represents the number of actions of type $A$ available in $LH$ and is evaluated by production rules analyzing the summarization facts related to $A$ asserted during the last $LH$; e.g., the number of distinct help buttons in the pages displayed during $LH$ is evaluated by analyzing the "helpButton" facts;
- $performedActions_A$ represents the number of distinct actions performed by the user when focusing on the product class denoted by the CF; e.g., the number of distinct help buttons pressed by the user, as far as the pages displayed in $LH$ are concerned, is evaluated by analyzing the "help" facts asserted in working memory.

The user may visit the same pages more than once in different local histories; so, $performedActions_A$ must be derived from the whole list of actions of type $A$ performed by the user in that focus space, not only from the actions performed within the last $LH$.

The production system also maintains a "tiredness" factor (see Section 3.2), which increases as the user visits new pages, views technical details, and so forth. This factor is set to '0' at the beginning of the interaction and, each time the user performs an action, it increases according to the following formula, taking values in [0, 1]:

$tiredness' = tiredness + (1 - tiredness) * load_A$

where $load_A$ is the contribution of a action type $A$ to tiring the user.

## 3.2 Interpretation of the user behavior

In this phase, the user behavior is evaluated to see whether the current user model is consistent with the user's real requirements, or it must be updated for filling the gap between the information needs expected by the system and those displayed by her/his actions. We have represented the relations between user behavior and user features in a Bayesian Net (BN), a portion of which is shown in Figure 3. This network explicitly models the influence of a user feature (e.g., "Receptivity") on other user features and/or on a specific aspect of the

user behavior (e.g., the creation of comparison tables: "Comparison Table"). In particular, the roots and the internal nodes of the network represent the user features: they correspond to the parameters defined in the user model and take values in the set {*low, medium, high*}.[4] The leaf nodes of the network correspond to the degree of activity performed by the user, subdivided according to the action type. For each type $A$, the related variable in the BN represents a qualitative evaluation of the occurrence rate ($rate_A$), which has been discretized into three ranges, so that the variable can assume one of the linguistic values: *small, medium, large*. For instance, "View Technical Details" describes the fact that the user asks to see the technical details about items in a *small, medium* or *large* percentage of cases.

The BN explicitly represents the impact of the user features on the user behavior. Let us consider for example "View Technical Details" (VTD), influenced by the receptivity and interests on the main product classes (we only consider "technical interest-i" -TI, as shown in the figure). The respective strengths of these features on VTD is captured by the BN since this formalism requires the specification of the conditional probabilities on the values of the related variables; e.g.:

P(VTD = small | R = low, TI = low) = 0.4;

P(VTD = small | R = high, TI = high) = 0.05;

specify that the probability that a user with low receptivity and technical interest views a small number of technical details is much higher (0.4) than the the corresponding probability, for a user with high receptivity and technical interest (0.05).

The only observable parameters are the ones related to actions (the leaves of the BN), while the values of the user features can only be inferred. The BN formalism has the advantages of enabling interpretation (explaining behavior in terms of user features) and prediction (on behavior, depending on the value of the user features). The mechanisms for propagating evidence in both directions are well known [12] and have been used for revising the user features on the basis of the observations of the user behavior in each local history. In the following we provide some details about the different steps involved in the interpretation of the user behavior.

**Initialization of the Bayesian Network**

While the conditional probabilities associated to the network are determined once for a given domain, the prior probabilities associated to the root nodes of the BN depend on the individual user. At the beginning of each interaction, the prior probabilities for the root nodes of the BN are initialized with the probability distribution of the user features contained in the current user model.

**Generation of the evidence for the BN**

Given a local history, the evidence for a leaf node of the BN associated to an action type $A$ is obtained in two steps:

1) First, the occurrence rate of $A$ is mapped on a probability assignment for the linguistic values of the node. The mapping assigns a probability distribution to each relevant interval of the occurrence rate;[5] let us consider, for example, the rate of action "View Technical Details", assuming that the observed value is 0.15. On the basis of the mapping described as:

0.0 - 0.1 → <small, 0.85>, <medium, 0.14>, <large, 0.01>;
0.1 - 0.2 → <small, 0.75>, <medium, 0.24>, <large, 0.01>; .....;
0.9 - 1.0 → <small, 0.01>, <medium, 0.14>, <large, 0.85>;

the assigned probability distribution is:

<small, 0.75>, <medium, 0.24>, <large, 0.01>.

The mapping allows the system to obtain an a-contextual evidence ($cur_{ev}$), resulting from an evaluation of the user behavior restricted to the local history under examination.

2) The a-contextual evidence ($cur_{ev}$) may be revised to consider the historical evidence, based on the less recent user behavior, and the possible tiredness of the user. For each action type (leaf of the BN), the historical evidence ($expected_{ev}$) represents the system's expectations on the user behavior and is summarized by the marginal probabilities computed by using the conditional probabilities of the BN and the current priors associated to its root nodes. The current priors capture the model of the user as it has evolved in the interaction so far. Given the expected and current evidence, we evaluate:

$\Delta = cur_{ev} - expected_{ev}$

where $\Delta$ represents the distance between the two probability distributions. The evidence to be fed to the BN is evaluated as:

$new_{ev} = expected_{ev} + (1 - tiredness) * \Delta$

If $tiredness$ is null, the evidence fed to the leaf nodes of the BN is exactly the a-contextual evidence (i.e., $new_{ev} = cur_{ev}$), whereas, if $tiredness > 0$, the evidence provided to the BN results from a merge of past history and observed user behavior in the local history. The larger is the tiredness, the smaller is the impact of $\Delta$ in changing the evidence.

**Interpretation of the synthesis of the user behavior**

The last step in the interpretation process concerns the revision of the user features, according to the user behavior. As soon as the new evidence $new_{ev}$ for a local history is evaluated, it is fed to the BN and in particular the probability distribution for each leaf node is given. This fact activates the interpretation mechanism of the BN: the observed evidence is propagated starting from the leaf nodes according the propagation algorithm for singly-connected networks described in [12]. We evaluate the new most probable explanation [12] for the user features and we take it as the updated version of the user model.

## 4 EVOLUTION OF THE USER MODEL

The need to update the user model according to the observed behavior determines the overall mechanism of user modeling. However, the opportunity that the style of the interaction with the user does not change too frequently or abruptly has to be considered. There are several reasons we have introduced the tiredness factor: first, the user model exploited to initialize the BN at the beginning of the interaction may contain partially wrong information, so that the user behavior can differ from the expected one. In order to quickly update the system's beliefs, it is thus important that the system is initially very sensitive to the user actions; later on, a conservative approach can be adopted (progressively ignoring the changes in the user behavior), assuming that the precision of the system's beliefs increases and the variations in the user's behavior are accidental; see also [3]. Second, when the user is tired, her/his behavior downgrades, reducing the number of actions that (s)he performs to get more information about products. While the user's receptivity is directly influenced by the tiredness, the other variables of the user model should not be strongly affected by this factor: e.g., the user's expertise does not downgrade as a consequence of the tiredness; however, being overloaded, the user becomes less active in asking for complex information, such as the one provided from the technical details.[6]

In order to test the ability of the system to update the user model on the basis of different actions performed by the user while brows-

[4] "Domain Expertise-i" and "Technical Interest-i" refer to one of the main product classes corresponding to the roots of the product taxonomy (e.g., phones). We have not reported the other features for simplicity.

[5] We have defined the mapping on the basis of the activity of users involved in a evaluation of SETA. Different mappings could be defined by considering the behavior of very different users classes; e.g., disabled people.

[6] Currently, the receptivity is updated as the other variables, on the basis of the first argument. The possibility to update it in a different way, or to extend the BN for explicitly modeling the tiredness factor (see [7]), is future work.

| values | start-LH1 | end-LH1 | start-LH2 | end-LH2 | limit |
|---|---|---|---|---|---|
| low | 0.1 | 0.1364 | 0.1585 | 0.1608 | 0.1628 |
| medium | 0.2 | 0.4271 | 0.4371 | 0.4381 | 0.4390 |
| high | 0.7 | 0.4365 | 0.4044 | 0.4011 | 0.3982 |

**Figure 4.** Evolution of the user's technical interest about phones.

ing the Web catalog (and consequently to personalize the interaction in a suitable way), we have performed a set of experiments by varying the user's degree of activity. The experiments confirm that the probability distributions of the user model may quickly change at the beginning of the interaction, while the stability of the user model increases as the user continues to browse the catalog.

Figure 4 shows the evolution of the user's technical interest about phones in two local histories, $LH1$ and $LH2$, performed by the same user in two phases of her/his navigation within the Web catalog. In both histories, the system displays three phone items and the user views the technical details of one item (with three available actions of that type) and follows the "more information" link in the description of two items (three available actions). The available actions also include the creation of a comparison table (to compare items on the basis of at most 10 features) and 17 help buttons. The figure also reports (in the "limit" column) an approximation of the probability assignment which would be obtained if the user maintained a uniform behavior for a significative time, always performing actions with the same occurrence rates as those described above (i.e., viewing the technical details once every three items, etc.). As it can be seen, in $LH1$ the probability assignment of the feature changes significantly; instead, in the second case, the revision takes the probabilities closer to the limit values, but the revision is much less relevant. This is partially caused by the tiredness factor, whose value is low in $LH1$ (0.1673), but increases in the rest of the interaction and, at the end of $LH2$, takes the value 0.4224.

## 5 DISCUSSION

We have described a framework for handling dynamic user models in an adaptive Web store which tailors the content of the catalog pages to the user's interests, domain expertise and receptivity. The revision of the user models is based on two main activities: monitoring the user's actions and interpreting them to revise the user model; in our approach, a production system and a Bayesian Network (BN) are exploited, respectively, to carry on the two tasks.

The exploitation of a production system for gathering the evidence about the user's actions allows an explicit description of the strategies for establishing the relevance of the actions and has noticeable advantages from the viewpoint of the system developers, because it enhances the understandability of the inferences performed by the system and it supports the revision of the strategies adopted to perform such inferences. Furthermore, a use of a BN is suited to handling the uncertainty in the inferences about the user [8] and supports an explicit representation of the variables relevant to such inferences and of the dependencies among them [14, 7].

An important constraint in the system architecture concerns efficiency issues. The set of experiments we have performed so far have shown that the overhead introduced by the dynamic revision of the user model is quite small. The subdivision of the whole process into two distinct steps (user monitoring and interpretation of the user behavior) has also advantages from a computational point of view. The production system is reasonably efficient since the monitoring activity is performed by subdividing the interaction into local histories of limited duration. Moreover, the analysis of the user's actions, performed within a local history, and not action by action, makes it possible to obtain a synthesis of the user's behavior and to exploit a static Bayesian Network (BN), where the individual actions that the user could perform don't have to be explicitly represented (and therefore the size of the BN is relatively small).

Finally, thanks to the fact that our system is based on a parallel agent architecture, the activity of the production system and the propagation of evidence in the BN can be performed in a separate thread of execution with respect to the main activity of the system; thus, the only activity which could cause a delay is the update of the user model with the new probability assignments.

As a last remark, our approach is suited to analyzing the behavior of a user who inspects the catalog in detail; instead, our interpretation of the actions is too pessimistic if the user browses the catalog without focusing on any product: in that case, the system takes her/his lack of activity as a sign of a low receptivity and interest for the information available in the catalog. An interesting extension would be the recognition of different navigation styles (see [13]) to apply appropriate interpretation strategies in the various cases.

## REFERENCES


[1] L. Ardissono and A. Goy, 'Tailoring the interaction with users in electronic shops', in *Proc. 7th Int. Conf. on User Modeling*, pp. 35–44, Banff, Canada, (1999).

[2] L. Ardissono, A. Goy, R. Meo, G. Petrone, L. Console, L. Lesmo, C. Simone, and P. Torasso, 'A configurable system for the construction of adaptive virtual stores', *World Wide Web*, **2**(3), 143–159, (1999).

[3] D. Chin, M. Inaba, H. Pareek, K. Nemoto, M. Wasson, and I. Miyamoto, 'Multi-dimensional user models for multi-media i/o in the maintenance consultant', in *Proc. 4th Int. Conf. on User Modeling*, pp. 139–144, Hyannis, MA, (1994).

[4] J. Fink, A. Kobsa, and A. Nill, 'Adaptable and adaptive information for all users, including disabled and elderly people', *New review of Hypermedia and Multimedia*, **4**, 163–188, (1998).

[5] J. Goecks and J. Shavlik, 'Learning users' interests by unobtrusively observing their normal behavior', in *Proc. 2000 Int. Conf. on Intelligent User Interfaces (IUI'00)*, pp. 129–132, New Orleans, (2000).

[6] B.J. Grosz and C.L. Sidner, 'Attention, intentions, and the structure of discourse', *Computational Linguistics*, **12**, 175–204, (1986).

[7] A. Jameson, R. Sähafer, T. Weis, n A. Berthold, and T. Weyrath, 'Making systems sensitive to the user's changing resourse limitations', *Knowledge-Based Systems*, **12**, (1999).

[8] A. Jameson, R. Shäfer, J. Simons, and T. Weis, 'Adaptive provision of evaluation-oriented information: tasks and techniques', in *Proc. 14th IJCAI*, pp. 1886–1893, Montreal, (1995).

[9] T. Joachims, D. Freitag, and T. Mitchell, 'WebWatcher: a tour guide for the World Wide Web', in *Proc. 15th IJCAI*, pp. 770–775, Nagoya, Japan, (1997).

[10] Sandia National Laboratories, 'JESS, the Java Expert System Shell', http://herzberg.ca.sandia.gov/jess/.

[11] G. Paliouras, C. Papatheodorou, V. Karkaletsis, P. Tzitziras, and C.D. Spyropolous, 'Large-scale mining of usage data on web sites', in *Working Notes of the "Adaptive User Interfaces" Spring Symposium of AAAI (Tech nical Report SS-00-01)*, pp. 92–97, Stanford, CA, (2000).

[12] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publ., San Mateo, CA, 1988.

[13] D. Petrelli, A.De Angeli, and G. Convertino, 'A user centered approach to user modelling', in *Proc. 7th Int. Conf. on User Modeling*, pp. 255–264, Banff, Canada, (1999).

[14] W. Pohl and A. Nick, 'Machine learning and knowledge representation in the LabouUr approach to user modeling', in *Proc. 7th Int. Conf. on User Modeling*, pp. 179–188, Banff, Canada, (1999).

[15] I. Schwab, W. Pohl, and I. Koychev, 'Learning to recommend from positive evidence', in *Proc. 2000 Int. Conf. on Intelligent User Interfaces*, pp. 241–247, New Orleans, (2000).