# A User Model to Support User Interface Navigation Adaptation Patterns in Web Applications

Juan Bautista Cabotà[1], José Ignacio Panach[1], Silvia Rueda[1], Damiano Distante[2]

[1]Escola Tècnica Superior d'Enginyeria, Departament d'Informàtica, Universitat de València
Avenida de la Universidad, s/n, 46100 Burjassot, Valencia, Spain
{Juan.Cabota,J.Ignacio.Panach,Silvia.Rueda}@uv.es
[2]Unitelma Sapienza University
Viale Regina Elena, 295, 00161 Rome, Italy
damiano.distante@unitelma.it

**Abstract.** Nowadays there are few real Web applications that support adaptive interfaces, and existing sites adapt interfaces depending on the information saved in the client browser (usually using cookies or logs). This implies that adaptations are carried out for each user independently, and that these adaptations cannot be shared among different devices used by the same user. Moreover, the system cannot extract knowledge to adapt interfaces through all the users. This paper proposes a user model for capturing attributes, behaviors and preferences of different users of a Web application, and the use of Bayesian Networks on the collected data to adapt the Web application user interface. Even though the proposed user model is generic enough to support different adaptive mechanisms, this paper focuses on mechanisms related to navigations. We have packaged the problems and solutions of such mechanisms in interaction patterns. A study on the top seven Web applications on the Internet has shown that only a few of these sites support some of our patterns, and that their implementation is not based on a user model that saves the data of all users.

**Keywords:** User Model, Interaction Patterns, Adaptive Interfaces.

## 1 Introduction

Adaptive systems are those systems able to adapt to the goals, characteristics and interests of their users, according to the knowledge represented in a user model [1]. Focusing on the user interface of a Web application, there are many aspects that can be adapted such as menus and navigation among others. In order to support these adaptations, many authors have defined a set of guidelines to consider when systems are being built. Examples of such guidelines, among others, are the ones defined by Park and Han [2] for menus, the ones defined by Brusilovsky for navigation links [3] and the ones defined by Knutov et al. [4] which are more generic . Even though some authors (such as Brusilovsky [3]) have defined a set of mechanisms to support their guidelines, how the guidelines are operationalized in a particular system depends on the specific problem to solve and on the analysts' skills. Moreover, all adaptive elements described in the guidelines depend on user' preferences, but no details are provided concerning how to save, analyze, interpret and adapt the Web application according to those preferences. All these tasks, which are not covered by existing guidelines, can be achieved through a user model. A user model represents the information about users that is essential to support the adaptation functionality [5].

Our contribution in this work is the definition of a user model to represent user' preferences in such a way that this information can drive the adaptation of the Web application. The model is based on interaction patterns as a solution to describe how to support adaptive elements using Bayesian Networks [6] as a way to save information on the use of pages and widgets by the user. The use of patterns derive from the work of Alexander [7] and has been widely used by the software engineering community. According to Tidwell [8], patterns improve the "habitability" of something and they make things easier to understand, more useful and usable.

From all the existing guidelines for building adaptive interfaces, we focus our work on the adaptive of navigation, since most of the existing adaptive systems work with this type of adaptation. Adaptive navigation is the mechanism that supports personalized access to information, adapting the links or altering their appearance. Studies such as the ones presented in [3] and [9] support the fact that users using interfaces with adaptive navigations were faster, needed fewer clicks and made fewer errors.

As an illustrative example to show the applicability of the patterns, we have chosen Amazon, as it is one of the biggest e-commerce Web applications that already supports several adaptive interfaces. Amazon only supports a few of the navigation adaptation patterns proposed in this paper, but it is useful to illustrate how all patterns could work on Amazon in the case of implementation. Note that the fact that Amazon already supports some of our adaptive interfaces does not mean that they rely on a user model to represent users' preferences and share them among all users as our proposal aims to do. How Amazon stores users' preferences and how it adapts navigations accordingly is undisclosed, which prevents the same solution from being used in other Web systems. Moreover, supported adaptations depend exclusively on the interaction of the current user, which is saved through cookies and logs. Information extracted from the interaction of other users can not be used to adapt the interfaces of the current user. The example of Amazon can help to better understand the problems that our patterns aim to solve, even though our solution is not the same as Amazon's.

Apart from Amazon, there are a huge number of Web applications that already support navigation adaptation only considering the interaction with the current user. We have also carried out a study of such real Web applications that have already faced with the same problems we aim to solve through patterns. This will help to show that our generic solution specified through patterns could be useful for real Web applications, enriching their adaptations through information gathered from the interaction of any user.

The remainder of the paper is structured as follows. Section 2 overviews related work on adaptive interfaces. Section 3 introduces our model for representing knowledge about the user and his/her behavior. Section 4 describes the patterns for adaptive navigation. Section 5 studies how many top Web applications support our patterns. Finally, Section 6 concludes the paper and introduces future work.

## 2 Related Work

The use of adaptive navigation interfaces aims to help users find useful information in a Web application. Several works in the area of recommender systems have pursued this objective, such as Bobadilla et al. [10], who have conducted a survey and identified three groups of systems: (1) systems based on recommendations through filtering; (2) systems that use algorithms that include social information; (3) hybrid ensemble algorithms that incorporate location information. Other works focus recommender systems on contextual opinions from user reviews, such as the work of Chen and Chen [11]. These authors propose using a contextual weighting strategy to adapt interfaces through a linear-regression algorithm.

There are other works that tackle the problem of interaction adaptation, such as the work of Hollink et al. [12]. These authors propose a framework that gives a systematic overview of alternative assumptions to optimize menus. The assumptions consider the navigation behaviour of users in the site's log file, so adaptations are specific for each user. Hammer et al. [13] have proposed an automatic decision-making system based on Bayesian Networks to support adaptive interaction. The system monitors the user over time and applies appropriate reactions according to what the system learnt from the user. The adaptation only considers the interaction of the own user. Nascimento and Schwabe [14] propose a data-driven, rule-based interface definition model that adapt interfaces taking into account the semantics of the data.

There are earlier approaches to abstractly represent how users interact with a system through a user model. Some of these approaches are only focused on content filtering, such as the one proposed by Kim et al. [15], who propose a collaborative approach to user modelling for personalized recommendations. The user model stores meaningful user patterns extracted from collaboration with similar users. Other approaches for building a user model are based on Artificial Intelligence, such as Papatheocharous et al. [16], who propose interface adaptations according to cognitive styles. The authors aim to find any possible relationships between the cognitive styles of users and their characteristics in navigation behaviour. Dim et al. [17] propose building user models from component-based software development. Their approach consists of building blocks that can be integrated as parts of the user model. The used architecture allows several blocks to be composed and the reuse of blocks in different systems. Sato et al. [18] propose a prediction model adapted for purchase prediction. The approach is based on the use of logs that are exclusive for each user.

After analyzing all the related works, we can conclude that there are very few proposals that allow adapting interfaces which consider the interaction of all users; most works focus their adaptations on current user's logs or cookies. Moreover, works that propose adaptation techniques based on knowledge extracted from several users do not present a sufficiently abstract solution which is suitable for any system. Next we present our proposal for a user model that aims to cover both gaps: (*i*) the model allows the storing of information from any user who interacts with the system; and (*ii*) the approach is based on the notion of interaction patterns, thus opening up the possibility of reusing the solution.

## 3 User Model

In the domain of Human Computer Interaction (HCI), a user model is a representation of the knowledge of users and their preferences [19] that gives an overview of their characteristics and helps with systems adaptation. In [20], a study of different user modeling techniques used in HCI concludes that user models are usually based on logic, fuzzy logic, neural networks, or probabilistic models. Out of all of these, probabilistic models seem to be the most suitable, since they can reflect, quantify and infer uncertainty about the users' preferences [21]. Within the group of probabilistic models, Bayesian Networks [6] have been widely used to represent user models. They consist of directed acyclic graphs with nodes and arches. Nodes represent observable quantities, unknown parameters or hypotheses, while the arcs between the nodes represent probabilistic dependencies among them.

Fig. 1 shows the metamodel of the proposed user model, which includes classes to store user data and classes to represent the adaptation mechanisms, in particular those based on Bayesian Networks and probabilistic data. The *User Model* is composed of several *Mechanisms* that aim to support interface' adaptation based on the previous actions of any user. Each mechanism provides a different technique to adapt interfaces. Examples of mechanisms are the filtering or ordering of navigation links, or the customization of the combination of colors used in the interface. Each mechanism is supported through several *Patterns*. Each pattern proposes a solution for a specific problem. Examples of patterns are described in next sections of this paper in detail. The solution of the pattern can be tackled in several ways, each one is called a *Function*. For example, a pattern to guide the user can use a function to highlight the suggested navigation links through different colors or a function to highlight them based on their size. The function depends on probabilistic data stored through a set of Bayesian Networks. These Bayesian Networks store two types of probabilities*: Probability of Use* and *Probability per User*. Probability of Use stores the probability for a specific *Widget* on a *Page* to be used. Probability of Use is useful for example to identify the most used widgets on a specific page. Probability per User stores the probability that a specific user uses a Widget on a Page. This information is useful to know the most frequently used widgets per type of user according to user characteristics. Widgets used in both Probability of Use and Probability per User can involve navigation to another page. In the Bayesian Network, each Page represents a node and the use of each Widget represents an arc. Widgets can be grouped to display options through the class *Widget Grouping*. The existence of both probabilities allows us to register users' preferences while they interact with the system.

The User Model saves the data of each user through *User Data*. For each user, the model stores his/her *User Characteristics*, which saves any information on the user that may be useful to adapt interfaces, such as his/her age or profession (among any other characteristic). Adaptation rules are defined through the class *Rule*, which includes an *Operator* that defines the type of comparison to check between a *Value* and probabilities stored in the Bayesian Network. Rules can be based both on the Probabilities of Use of any user, or on the Probability per User to adapt interfaces to users with specific User Characteristics. Note that the definition of rules (Operators and

Values) to know when to apply the function of adaptation is beyond the scope of this paper and is more related with HCI and psychology. When the condition expressed through this formula is satisfied, the interface is adapted according to what is specified in the class *Display Option*. Note also that this work does not tackle how to represent interaction features through the attribute *Configuration*. There are already many notations in the literature that propose conceptual primitives to model interfaces, such as UsiXML [22] or IFML [23].

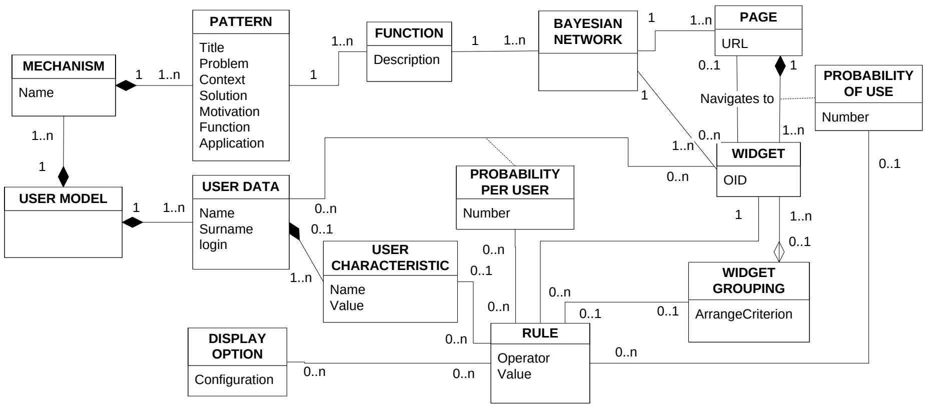


**Fig. 1.** Metamodel of the proposed user model.

As an example of a User Model, we can specify a Mechanism such as "arranging widgets" through the Pattern "link ordering". This pattern has the function to "order the widgets". The Bayesian Network can store Probabilities of Use to adapt interfaces according to the use of any user and Probabilities per User to adapt interfaces to users with User Characteristics similar to the current user, such as "Age". The Rule defines when to apply the ordering according to probabilities, the Widget Grouping defines the group of widgets to order and the Display Options defines how to show the ordered widgets. Due to reasons of space, out of all the existing Mechanisms which adapt interfaces, this paper focuses on Navigation which is also the most frequently used [12] [24]. Next, we define a set of interaction patterns for adaptive navigation..

# 4 Specification of Adaptive Interaction Patterns for Navigations

## 4.1 Background

From all the existing guidelines to define interaction patterns, we focus our proposal on the work of Brusilovsky [3]. The choice is based on three main reasons: (1) the Brusilovsky's set of guidelines is wide, (2) the proposed adaptation techniques can be found in many real Web applications, and (3) they are focused on navigation features compliant with the idea of working with Bayesian Networks to represent probabilities. The guidelines defined by Brusilovksy are the following:

- *Direct Guidance*: Suggest the “next best” node (or several alternative nodes) to visit for the user, according to the user’s goals, knowledge, or/and other parameters that have been represented in the user model.
- *Link Ordering*: Prioritize all links of a particular page according to the user`s preferences and some user-valuable criteria: the closer to the top, the more relevant the link is.
- *Link Hiding*: Restrict the navigation space by hiding, removing, or disabling links to irrelevant pages.
- *Link Annotation*: Augment the links with some form of annotation, which lets the user know more about the current state of the page behind the annotated links.
- *Link Generation*: Create new links on a page. There are three types of link generation: (1) discovering new useful links between the documents and adding them permanently to the set of existing links; (2) generating links for similar items; and (3) dynamic recommending links that are useful within the current context (i.e., the current goal, knowledge, or interests, as reflected in user’s preferences).

Brusilovsky describes these guidelines in such a generic way that makes their implementation difficult. There are not details about what we can measure to personalize the adaptation or how to store the information extracted from each subject. In order to solve these problems, we have defined a set of patterns based on our proposal of a user model.

### 4.2 Template Definition

Patterns are usually described as formal ways of documenting solutions to common design problems, and they are used in different fields of expertise [8]. More specifically, a pattern is the description of a common problem and a possible solution to it, following a defined structure. It provides important benefits to software development in terms of re-usability and flexibility.

User interface analysts usually have to design user interfaces taking into account different issues such as, technical and functional requirements, graphical richness, responsiveness, accessibility and also efficiency and usability. The use of interaction patterns to deal with these problems is very common. Examples of interaction patterns are those provided by the Yahoo User Interface Design Pattern Library [25]. We propose the use of interaction patterns as a solution to build adaptive interfaces. These patterns must allow any system to learn from the interaction of each subject and to build a user model (particularly the one proposed in Fig. 1) that gathers users’ preferences. Next, this user model can be used to adapt interfaces for each subject. For the description of our patterns we will adopt a template derived from the one most widely used to specify Yahoo patterns. This template includes the following elements:

1. A *title*, an intuitive name that unequivocally identifies the pattern.
2. The description of the *problem* that we intend to solve.
3. The *context of application*, the types of widgets where the pattern could be applied.
4. The navigation support technology that we suggest as a *solution* to the problem.

5. The *motivation*, explaining the origin of the interaction technology solution.
6. The *function* used to save users' preferences and to adapt interfaces according to them. This is an instance of the class Function in Fig. 1.
7. A *concrete application* of the pattern illustrated with an example.

Note that the element *function* has been added as an extension of Yahoo User Interface Design Pattern Library patterns, since the original template does not have sufficient expressiveness to represent users' preferences through a user model. How we propose working with functions is explained in next section.

### 4.3 Adaptive Interaction Patterns for Navigation

This section defines the set of interaction patterns that we propose to tackle the adaptation mechanism Navigation (see Fig. 1). From all the existing Web applications, to illustrate the applicability of the patterns, we have chosen Amazon, since it is widely used and known [26]. Amazon already supports some adaptation mechanisms based on navigation. However, these mechanisms have not been tackled as patterns. The lack of patterns implies that the reuse of the solution in other Web applications is not possible, moreover, it is impossible to find out how Amazon saves and works with data on users' profiles and behaviors, since the adopted user model (if any) is not known. What we aim to illustrate with the example of Amazon is to show how the patterns that we propose can work in a practical way. This does not mean that real Web applications already support our proposal, but that some of them (such as Amazon) implement some guidelines to support adaptability in their own way. In some other cases, Amazon does not support the functionality of the pattern yet but it is useful to illustrate how the pattern could be applied. Taking as input Brusilovsky's guidelines [3], we have proposed a set of user interface adaptation patterns:

**Direct guidance**

*Problem*: The existence of too many links on a Web page might confuse the users, especially in the case of novice users, when they have a concrete goal and they only need a few links to achieve it. *Context of application*: Processes in which the user has to follow different links in consecutive order to reach a concrete goal (e.g., buying a product). This is suitable for contextual links, non-contextual links, tables of contents, indexes and hyperspace maps [3]. *Solution*: The most used links in each step of the process can be emphasized, or generated automatically by the system according to the user model. *Motivation*: Several studies such as [1] demonstrated that novice users have problems in dealing with different navigation choices and are better supported by direct guidance technology.

*Function*: The Bayesian Network saves data about the most used links (through their probabilities) and data about users' characteristics that use such widgets. This way, the system can display with a different Display Option the links whose Probabilities of Use fit within an Operator and a Value defined through a Rule. Moreover, we can also define Rules to highlight the most used links through a specific Display Option for users with the same User Characteristic as the current user. Probability per User saves the most used widgets for a specific user. *Concrete application*: In the process of buying a product on Amazon, there is a wizard that guides the user throughout the process. Imagine that we would like to highlight in yellow the most used link for all users in general (this functionality is not supported by Amazon currently, which always highlight the same button). Fig. 2a shows the links presented on the Web page for selecting the shipping address in the Amazon checkout process while Fig. 2b shows an example of an Object Model to represent that the most probable links must be highlighted. The object "Guidance" is a Bayesian Network that saves the different Probabilities of Use of each widget on the page. For the widget "Ship to this address" we have specified a Bayesian Rule in such a way that widgets whose Probability of Use is greater than 0.7 must be displayed in "Yellow".

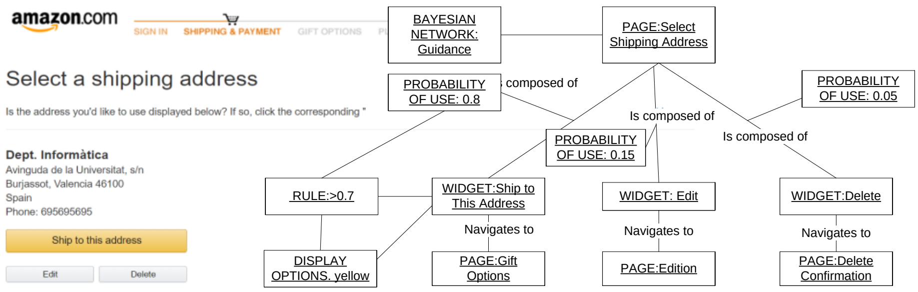


**Fig. 2.** a) Example of Direct Guidance in Amazon; b) Object Model to apply Direct Guidance.

**Link ordering**

*Problem*: In some Web applications, a page may present a huge quantity of items in an irrelevant order for the user, who would spend a considerable amount of time browsing them in order to find target items, thus entailing a lack of usability. *Context of application*: Non-contextual links such as lists of learning topics, lists of news, list of products, ideas and tags clouds, and any other lists of resources where the unstable order of links creates no problem. Several studies, such as [27], have demonstrated that unstable order creates problems for some categories of users. *Solution*: The links are ordered according to users' preferences stored in the user model, which can be done by monitoring the browsing history, or collecting user characteristics such as age, gender, nationality, etc. *Motivation*: Kaplan et al. [24] demonstrated that link ordering reduces navigation time and the number of steps that are required to locate the information that the user is looking for.

*Functions*: The Bayesian Network registers the percentage of clicks executed on each link, so that the system may order the links according to them. Another option is to implement a Rule that orders the links according to how they are Probably Used by other users with the same User Characteristics as the current user. For example, a list of links may be more suitable for users with ages in a specific range, so the age of the user could determine the order in which the links are presented. Links are grouped in a Widget Grouping in such a way that the ones which are most accessed by users with the same age as the current user can be displayed at the top of the list. *Concrete application*: This pattern is already supported by Amazon, when the user visits a page presenting a specific product, he/she is also presented with a long list of other recommended products, as Fig. 3a shows. As an example of applying the Link Ordering pattern, this list could be sorted according to the age of the user. The links most used by users with the same age as the current user could appear first. Fig. 3b shows an example of an Object Model. The object Widget Grouping defines the order criterion based on the User Characteristic Age which is equal to the age of the current user.

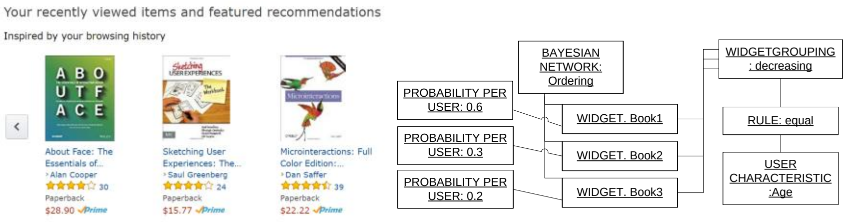


**Fig. 3.** a) Example of Link Ordering in Amazon; b) Object Model to apply Link Ordering.

### Link hiding

*Problem*: The existence of irrelevant links in a Web application increases the complexity of the navigation space and the cognitive overload of the users. *Context of application*: Contextual links (disabling them), non-contextual links and hyperspace maps. *Solution*: Hide, remove or disable links to irrelevant pages according to the User Model. *Motivation*: Some systems include large collections of links, resources, options, form fields and navigational elements on a single Web page, which creates a huge quantity of information and different choices that may confuse the user.

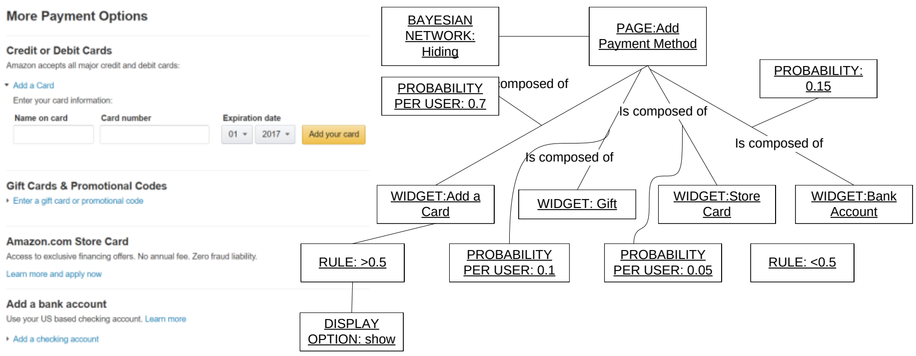


**Fig. 4.** a) Example of Link Hiding in Amazon; b) Object Model to apply Link Hiding.

*Function*: The Bayesian Network saves the percentages of use of each link, in such a way that the system can be adapted by hiding the less used elements, and showing them only on demand. Another option is to hide (through Display Option) the least used widgets for users with the same Users Characteristics as the current user. For example, the least used links among users of the same age as the current user. *Concrete application*: This pattern is not supported by Amazon yet, but we can explain how it could be applied. When the user has to select the payment method in the Amazon checkout process, it is presented with four choices: *Credit or Debit Cards*, *Gift*

*Cards & Promotional Codes*, *Amazon.com Store Card* and *Add a bank account*, as shown in Fig. 4a. Three of these options could be hidden by default, showing only the one most frequently used: *Pay with Credit or Debit Card*. Fig. 4b shows an Object Model with the examples of Probabilities and a Rule to define that the widget of *Add a Card* must be shown when its probability is greater than 0.5 (for example).

**Link annotation**

*Problem*: The users may have access to every link in lists such as menus, tables of contents and indexes, in order to look for the target information. In such lists, there are links to information which is more or less suitable depending on the user profile. *Context of application*: Contextual and non-contextual links, menus and sub-menus, tables of contents, indexes, hyperspace maps. *Solution*: Provide more information on the target page of a link through a graphical annotation, such as using different colors, adding an icon, attaching a tooltip with a small text or a number, altering the font format, or adding an extra description on the browser's status bar or on a pop-up that appears when the user passes the mouse pointer over the link. *Motivation*: The annotation mechanism lets the user know more about the current state of the target page of an annotated link. Unlike link hiding which can distinguish only two states (available or not available), this pattern can distinguish more than two states.


**Fig. 5.** a) Example of Link Annotation in Amazon; b) Object Model to apply Link Annotation.

*Function:* The Bayesian Network stores information about the most used links in such a way that the state of each link can depend on the level of use of each one of them. The states can be defined through Rules and Display Options, which defines the layout. For example, the most used link may appear in green, links used sporadically may be displayed in yellow, and the least used ones may be represented by red. We can also define Rules to assign a state to any link depending on User Characteristics. Links that are frequently used by users with similar characteristics can be represented in different states depending on the Probability per User. *Concrete application*: Fig.

5a shows an illustrative example of an application in Amazon (this example is not real). The link with the circle in green means that it is the most accessed for users with the same User Characteristic ("Profession") as the current user. Yellow means that the link is accessed with less Probability per User and red means that it has not been accessed yet by people with the same "Profession". The colors and how they are displayed, can be defined in Display Options. Fig. 5b shows an example of an Object Model to represent this scenario. Even though this pattern is not currently supported by Amazon, it implements a similar feature to highlight best seller products by adding an orange "Best seller" tag to them (this is also shown in Figure 5a).

**Link generation**

*Problem*: The abundance of resources and the existence of many links in a Web application, make it difficult to look for a specific item. *Context of application*: Non-contextual links. *Solution*: Creating new links as shortcuts to other pages or generating similarity-based links that may be useful within the current context according to the user model. *Motivation*: Most of the previous patterns such as annotation or hiding, adapt the presentation of the existing links on a page to the characteristics or preferences of the user, but they do not generate new links nor create shortcuts.

*Function*: The Bayesian Network with the representation of the user model stores the Probability of Use of each link. This way, the most used ones can be added as shortcuts to other pages. We can also define Rules to create shortcuts to the pages most frequently used by users with similar User Characteristics to the current user. For example, we can create shortcuts of links most frequently used among users with the same "Profession". *Concrete application*: Fig.6.a shows an example of shortcuts in Amazon that have been created depending on information stored in previous interactions. This functionality is already supported by Amazon. Fig. 6.b shows an example of an Object Model that supports this feature. The Bayesian Network saves the Probability of Use of each link during the interaction of any user. Most users who access the UML book next navigate to the link to the Design Patterns book, so Amazon creates Design Patterns link inside the page of the UML book.

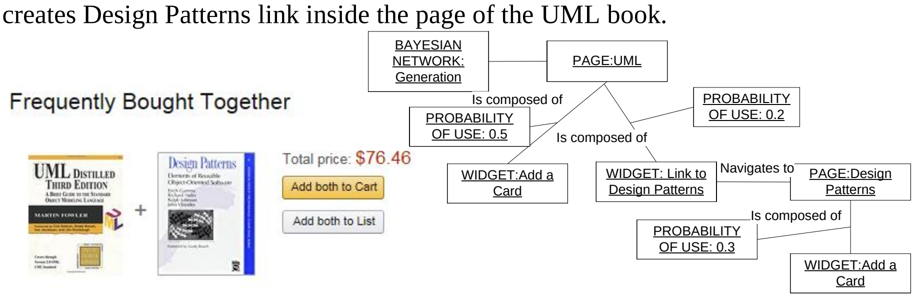


**Fig. 6.** a) Example of Link Generation in Amazon; b) Object Model to apply Link Generation.

## 5 Use of Interface Adaptation Patterns in Real Web Sites

This section analyzes the top 7 Web sites in the world (excluding Chinese Web sites) according to [26]. For each Web site, we study whether or not the functionalities of our proposed adaptive interaction patterns are supported. Note that the fact of supporting the functionalities does not mean that their design follows our patterns. All Web sites store preferences through logs or cookies in the client browser and some of them, such as Amazon, save information about the purchases. However, there is not a common repository that reports the preferences of all subjects and actions beyond logs, as our proposal aims to do through the user model. If users' preferences are saved in the client browser through logs, they cannot be shared among different devices. Moreover, systems cannot adapt interfaces by analyzing information of all users, which implies that adaptations are exclusive for each subject. This makes impossible to adapt interfaces regarding user preferences with characteristics similar to the current user.

The results of the study are shown in Table 1. Youtube is the Web site that best supports the adaptability, while Google or Wikipedia are the least adaptable to navigation's preferences. Patterns that depend on the historical data of several users are not supported by any of the Web sites, for example there is no Web site that implements Annotation since it needs to report information on users with similar characteristics to the current user. In general, supported patterns are those that depend on previous actions of the subject (independently of other subjects).

These results show that most used Web sites in the world support some adaptation mechanisms, but there is still a wide scope to include new techniques in order to enhance user experience. All adaptation mechanisms depend exclusively on information extracted from the current user. The user model that we propose contributes, with sufficient flexibility, to report data from different users depending on their characteristics or behaviour during their interaction. Moreover, existing Web sites have their own solution which is not made public. One of the advantages of using patterns in our approach is the reusability that they offer.

**Table 1.** Ad-hoc implementations of our proposed patterns in the top seven Web sites in the world.

| | Guidance | Ordering | Hiding | Annotation | Generation |
|---|---|---|---|---|---|
| **Google.com** | X | X | X | X | X |
| **Facebook.com** | ✓ | X | ✓ | X | X |
| **Youtube.com** | ✓ | ✓ | ✓ | X | ✓ |
| **Yahoo.com** | ✓ | X | ✓ | X | X |
| **Wikipedia.org** | X | X | X | X | X |
| **Amazon.com** | X | ✓ | X | X | ✓ |
| **Twitter.com** | ✓ | X | ✓ | X | ✓ |

## 6 Conclusions

This paper proposes a user model to represent users' preferences in such a way that Web applications can be adapted according to them. The user model reports information from any interaction of any user, allowing the sharing of adaptations of users with similar characteristics. The user model is abstract enough to support any type of preference, but for space reasons we focus on the mechanisms to adapt navigations.

To this aim, we have defined a set of patterns that specify how to adapt navigations through our user model.

Note, importantly, that our approach focuses on how to model users, but we do not discuss how to model interfaces to support adaptability. How the information reported in the user model is transferred to interfaces is beyond the scope of our work. The definition of interaction patterns aims to reduce the cognitive load to use the user model, but how interfaces are described depends on the software development method. There are already widely used notations such as UsiXML or IFML.

We have detected some limitations of the proposed user model. First, the user model becomes very large for a real Web application. Moreover, the use of Bayesian Networks increases the size of the user model since each page and link is represented in it. Second, the user model fits with any mechanism of adaptation, so it must be instantiated to a specific mechanism. Third, the storage of users' preferences in a Web server allows for the customization of the pages but it involves more traffic on the net.

As future work, we plan to build a tool to support building models through the user model and a validation of the approach implementing the patterns in a real Web site. We also plan to define patterns for adaptation mechanisms different from navigation and the inclusion of accessibility mechanisms. We aim to study optimization algorithms to better look for information in huge Bayesian Networks needed for real web system with millions of users.

## Acknowledgement

This work has been supported by Generalitat Valenciana under Project IDEO (PROMETEOII/2014/039) and the Spanish Ministry of Science and Innovation through project DataMe (TIN2016-80811-P).

## References


1. Brusilovsky, P.: Adaptive Hypermedia. User Modeling and User-Adapted Interaction **11** (2001) 87-110
2. Park, J., Han, S.H.: Complementary menus: Combining adaptable and adaptive approaches for menu interface. International Journal of Industrial Ergonomics **41** (2011) 305-316
3. Brusilovsky, P.: Adaptive navigation support. In: Peter, B., Alfred, K., Wolfgang, N. (eds.): The adaptive web. Springer-Verlag (2007) 263-290
4. Knutov, E., Bra, P.D., Pechenizkiy, M.: AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. New Rev. Hypermedia Multimedia **15** (2009) 5-38
5. Brusilovsky, P., Millán, E.: User Models for Adaptive Hypermedia and Adaptive Educational Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Springer Berlin Heidelberg, Berlin, Heidelberg (2007) 3-53
6. Charniak, E.: Bayesian networks without tears: making Bayesian networks more accessible to the probabilistically unsophisticated. AI Mag. **12** (1991) 50-63

7. Alexander, C., Ishikawa, S. & Silverstein, M.: A Pattern Language: Towns, Buildings, Construction. Oxford University Press (1977)
8. Tidwell, J.: Designing interfaces. " O'Reilly Media, Inc." (2010)
9. Reinecke, K., Bernstein, A.: Improving performance, perceived usability, and aesthetics with culturally adaptive user interfaces. ACM Trans. Comput.-Hum. Interact. **18** (2011) 1-29
10. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. Knowledge-Based Systems **46** (2013) 109-132
11. Chen, G., Chen, L.: Augmenting service recommender systems by incorporating contextual opinions from user reviews. User Modeling and User-Adapted Interaction **25** (2015) 295-329
12. Hollink, V., van Someren, M., Wielinga, B.J.: Navigation behavior models for link structure optimization. User Modeling and User-Adapted Interaction **17** (2007) 339-377
13. Hammer, S., Wißner, M., André, E.: Trust-based decision-making for smart and adaptive environments. User Modeling and User-Adapted Interaction **25** (2015) 267-293
14. Nascimento, V., Schwabe, D.: Semantic Data Driven Interfaces for Web Applications. In: Daniel, F., Dolog, P., Li, Q. (eds.): Web Engineering: 13th International Conference, ICWE 2013, Aalborg, Denmark, July 8-12, 2013. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (2013) 22-36
15. Kim, H.-N., Ha, I., Lee, K.-S., Jo, G.-S., El-Saddik, A.: Collaborative user modeling for enhanced content filtering in recommender systems. Decis. Support Syst. **51** (2011) 772-781
16. Papatheocharous, E., Belk, M., Germanakos, P., Samaras, G.: Towards Implicit User Modeling Based on Artificial Intelligence, Cognitive Styles and Web Interaction Data. International Journal on Artificial Intelligence Tools **23** (2014) 1440009
17. Dim, E., Kuflik, T., Reinhartz-Berger, I.: When user modeling intersects software engineering: the info-bead user modeling approach. User Modeling and User-Adapted Interaction **25** (2015) 189-229
18. Sato, M., Izumo, H., Sonoda, T.: Modeling Individual Users' Responsiveness to Maximize Recommendation Impact. Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization. ACM, Halifax, Nova Scotia, Canada (2016) 259-267
19. Benyon, D., Murray, D.: Applying user modeling to human-computer interaction design. Artificial Intelligence Review **7** (1993) 199-225
20. Biswas, P.: A Brief Survey on User Modelling in Human Computer Interaction. Speech, Image, and Language Processing for Human Computer Interaction: Multi-Modal Advancements. IGI Global (2012)
21. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc. (1988)
22. Vanderdonckt, J., Limbourg, Q., Michotte, B., Bouillon, L., Trevisan, D., Florins, M.: USIXML: a User Interface Description Language for Specifying Multimodal User Interfaces. Proceedings of W3C Workshop on Multimodal Interaction WMI'2004, Sophia Antipolis, Greece (2004)
23. OMG: Interaction Flow Modeling Language (IFML) : http://www.ifml.org/.
24. Kaplan, C., Fenwick, J., Chen, J.: Adaptive hypertext navigation based on user goals and context. User Modeling and User-Adapted Interaction **3** (1993) 193-220
25. Yahoo: Yahoo Developer Network. URL https://developer.yahoo.com/ypatterns/.
26. Alexa: http://www.alexa.com/topsites.

Debevc, M., Meyer, B., Donlagic, D., Svecko, R.: Design and evaluation of an adaptive icon toolbar. User Modeling and User-Adapted Interaction **6** (1996) 1-21