

# Patterns for Adaptive Web Applications<sup>1</sup>

Nora Koch\*, Gustavo Rossi\*\*

\*University of Munich and  
F.A.S.T. Applied Software Technology GmbH, Munich,  
Germany  
kochn@informatik.uni-muenchen.de, koch@fast.de

\*\*LIFIA Facultad de Informatica, UNLP, La Plata,  
Argentina  
gustavo@sol.info.unlp.edu.ar

**Abstract.** In this paper we present some patterns for adaptive Web applications, i.e. those Web applications that may change their behavior regarding the current user. These patterns refine the coarse grained personalization patterns in [Rossi 01], focusing on how adaptation can be implemented through the manipulation of links, and content and presentation of nodes. We first introduce the problem of building adaptive Web applications; next we introduce the patterns Adaptive Anchor Annotation, Anchor Selection, Sorting of Anchors and Conditional Fragments.

## Introduction

Personalized Web applications tailor the information presented and the structure to the user's preferences, knowledge or interests. Adaptive Web applications perform this adaptation dynamically learning from the user's navigation and interaction behavior. Most of these applications perform a rule-based adaptation and include the management of a user profile as it is formally specified by the Munich Reference Model for adaptive Web applications in [Koch 02]. We focus on persistent personalization (adaptation) in the sense of the definition given in [Schafer 00], i.e. when two different users may perceive different information even when accessing the same object. The adaptation process may include changes such as the selection of pieces of information that are appropriate to the knowledge level of the user, or some guidance performed through the removal of links that the system considers not relevant to the user. The system takes such decisions according to the knowledge the system has of the user at given point in time, knowledge that is given by the current state of the user profile. Two different forms of adaptation (techniques) are distinguished by [Brusilovsky 96] for Web applications:

---

<sup>1</sup> Copyright (C) 2002 by Nora Koch and Gustavo Rossi.

adaptive presentation and adaptive navigation support. These are adaptation at content-level and at link-level. Another possible adaptation is a change at presentation-level, i.e. changes to the layout that do not affect the content, such as colors, font type or font size [Paterno 99]. Summarizing:

- *adaptive content* consists of selecting different information, such as different text, images, videos, animation, etc. depending on the current state of the user profile. For example, an adaptive Web application provides an expert in a certain domain with more information than a novice.
- *adaptive navigation consist of* changing the anchors appearance, the link targets or the number of anchors presented to the users as well as the order in which these anchors are presented. Note that we distinguish whether the link or the anchor is adjusted to make the application adaptive while in the literature very often both are treated as link adaptation.
- *adaptive presentation* shows different layouts of perceivable user interface elements, such as different type of media, different ordering or different colors, font size, font type or image size.

Making a Web system adaptive has not only advantages, it also implies a set of risks. First of all, moving partial control from the user to the system is quite the opposite of the hypertext paradigm philosophy , which is supposed to give the user full control to explore the hyperspace. The design of an adaptive interface means therefore that special attention must be paid to risk of disorientation due to over-personalization. The adaptive patterns to be used must be non-intrusive, motivating, non-disorienting, and helpful. One of the problems is that the interface of adaptive systems is less stable for the user. The user may be irritated by incomplete and/or hidden information or anchors. For example, if the user goes back to take another look at pages he has already seen, these pages very often look different to the first time he saw them, as they are generated dynamically according to the current state of the user profile. In such a case usability is not increased through personalization. This risk is eliminated by systems like SmexWeb [Albrecht 00], which keep a history of changes to the user profile. In this way one page has the same look and feel throughout a session for a particular user.

In this paper we present a set of patterns that complements those in [Rossi 01] dealing with personalized Web Applications. These Web patterns are:

- Adaptive Anchor Annotation, shows different annotations according to the user interests or preferences.
- Adaptive Anchor Selection, reduces the navigation space.
- Adaptive Sorting of Anchors, indicates the relevance of links for the user.
- Adaptive Conditional Fragments, personalizes the content of the information nodes.

Figure 1 shows the dependencies between these patterns used in Adaptive Web Applications. How they are related is explained in the section Related Patterns of each pattern description.

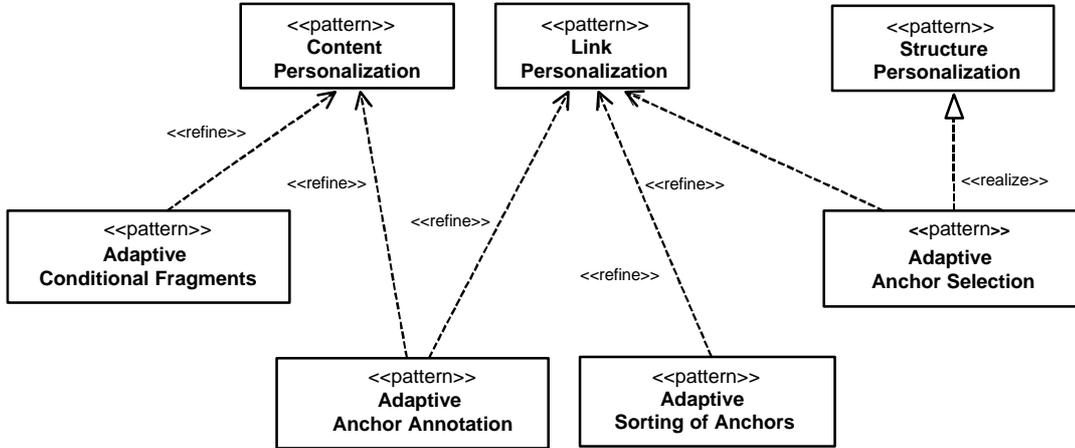


Figure 1: Dependencies between Patterns used in Adaptive Web Applications

Our patterns are based on the adaptation techniques supported by the UML-based Web Engineering approach for development of Web adaptive applications [Koch 01]. There are two main differences in philosophy between the set of patterns in this paper and the patterns Link Personalization, Content Personalization, Structure Personalization and Client-Side Personalization in [Rossi 01]. First the new ones are finer grained and second they provide a more dynamic way of personalization through dynamic adaptation of the contents and adaptation of the application’s topologies. We include an Appendix with the previously mentioned patterns.

## 1 Adaptive Anchor Annotation

### 1.1 Intent

Offer more information before the user selects an anchor about what the user will find as target of the link. It gives a positive measure of how appropriate the “following” node may be so the user can evaluate before he decides which link he will follow.

### 1.2 Motivation

A Web application is a complex net of related pages connected by links. Some anchors for links indicate the target of the link implicitly. For example in an e-commerce application (as in [www.amazon.com](http://www.amazon.com) in Figure 2) when you click an anchor named “your recommendations” you can expect what you will find. However, it is not clear what should we expect to find when clicking the “Why was I recommended this?” anchor (See Figure 1).

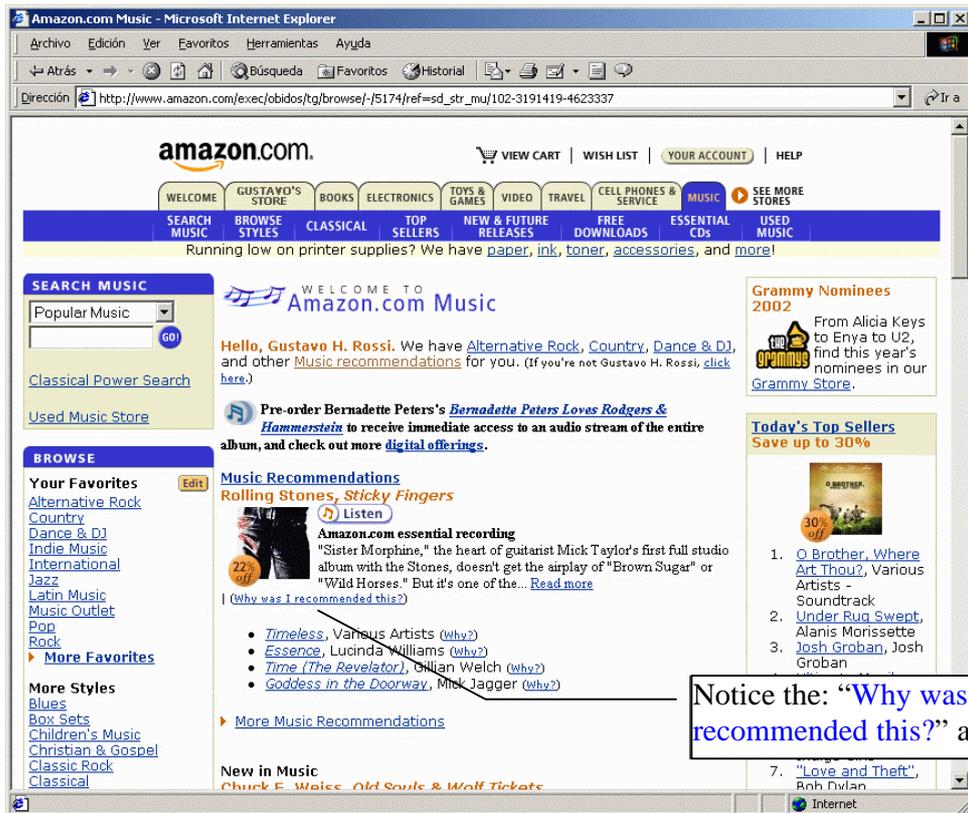


Figure 2: Annotations in Amazon.com

### 1.3 Forces

- Users of Web applications deal with many anchors of links and deciding which one to follow is not easy.
- Different users may have different interests and this interest will manifest in which links are more important for them.
- We may want to take into account the user's preferences to suggest to them which items he should explore.
- Simple annotation (not personalised) of a link, however, does not completely solve the problem.

### 1.4 Solution

Annotate anchors with adaptive information according to the user interest or preferences (user profile). Annotations can help the user to decide which link to follow. Anchors are "annotated" to present a different visible aspect, such as different color, underlying, bullet or additional text to show the relevance of the destination. Personalizing the link [Rossi 01], i.e. pointing the link to different targets according to the user profile, does not solve the problem either because we still want to help the user decide whether he has to navigate or not. The most widely known variant of Adaptive Anchor Annotation is

changing the anchor's color to show the alternatives visited/not yet visit. Another example of annotation is the use of special icons, such as colored bullets or different symbols for the anchors to show a degree of their appropriateness to the user. Special cases of anchor annotation are anchor highlighting and anchor hiding. Hidden anchors can be represented by annotation that show the anchor text in the same way as the text surrounding the anchor, i.e. the link of such an anchor is still available but the formatting used to distinguish it as an anchor has been removed.

Adaptive anchor annotation is a pattern that is used to support adaptive navigation through the pages of a Web application.

## **1.5 Examples**

Traffic light is a well known example of anchor annotation used to show relevance, e.g. to the current state of user knowledge, of the information hidden behind a link. In such a case, anchors are annotated with the colors green, yellow and red. In [Campbell 99] the authors use traffic lights added around the anchor text of each link to indicate its connection speed. Another examples are the tutoring systems ELM-ART [Schwarz 96] and SIGUE [Carmona 02]. SmexWeb [Albrecht 00] annotates links with smilies with three different shapes to show the appropriateness and relevance of the link for the user. [Bieber 97] provides meta-information with annotation to help users to decide to follow a link or not. The PersonalWebWatcher is a personal agent that searches the Web structuring the hits according to the interests the system believes the user has [Mladenic 00].

## **1.6 Consequences**

- Each user receives information customized to her/his profile.
- Overloaded information because the user has to read (if annotation is a text) or observe and analyze more information (if annotation is given by images or icons).
- Orientation through additional information; disorientation if the annotation changes dynamically too often.
- From the design point of view, we need to administrate a user profile if a user profile is used for the annotation.

## **1.7 Related Patterns**

Behavior Anticipation [Rossi 00] helps the more general problem of anticipating the result of an interface object activation. Link Personalization [Rossi 01] is usually refined with Adaptive Anchor Annotation to connect the current node with different ones according to the user profile. Adaptive Anchor Annotation may be thought as a refinement of Content Personalization (see Appendix). In this case the content which is personalized is the (meta) information about the meaning of an anchor.

## 2 Adaptive Anchor Selection

### 2.1 Intent

Adapt the navigation topology to the current user interest and/or preferences. This way the application selects the anchors that the system considers they are appropriate for the current user at that given point in time. Adaptive Anchor Selection can be seen as the addition of anchors to an empty list of anchors or as the removal of anchors from the list of all possible anchors.

### 2.2 Motivation

Web applications usually offer so many navigation possibilities on each Web page that there is a high probability that the user gets overwhelmed by the amount of information. For example in [www.amazon.com](http://www.amazon.com) when you select one kind of music (e.g. rock) you enter into an immense sea of possible CDs to buy. You may end browsing through hundreds of CDs to discover that only one is of interest to you. The problem with this approach is that it takes you away from the page in which you are (e.g. the Music Home Page) and it may be cumbersome to include this link in each different level of the taxonomic structure of the store (in fact it is not included in Amazon). One possible solution is to use Link Personalization [Rossi 01] as is done in the Amazon home page to recommend CDs according to the user profile.

Personalizing the structure of the home page as is done in [my.yahoo.com](http://my.yahoo.com) [Rossi 01] is also not a good solution; it works well for letting the user select those musical genres (or more generally subjects) in which he is interested. But once inside a specific genre the problem remains.

### 2.3 Forces

- Web pages are overloaded with anchors and deciding which of them are relevant is not always straightforward.
- We may want to take into account the user's preferences to help to find some items of his interests.
- We do not want to distract the user by moving him to a specific recommendation page.

### 2.4 Solution

Reduce the set of anchors shown to the user. Only provide him with those anchors that will let the user navigate to the items of his interest [Albrecht 00]. Given a particular user profile, anchors that the system considers inappropriate for the user are not included, i.e. they are no longer available. Anchors of these links may be replaced by text, or just eliminated.

Adaptive anchor selection is a pattern that is used to support adaptive navigation through the pages of a Web application.

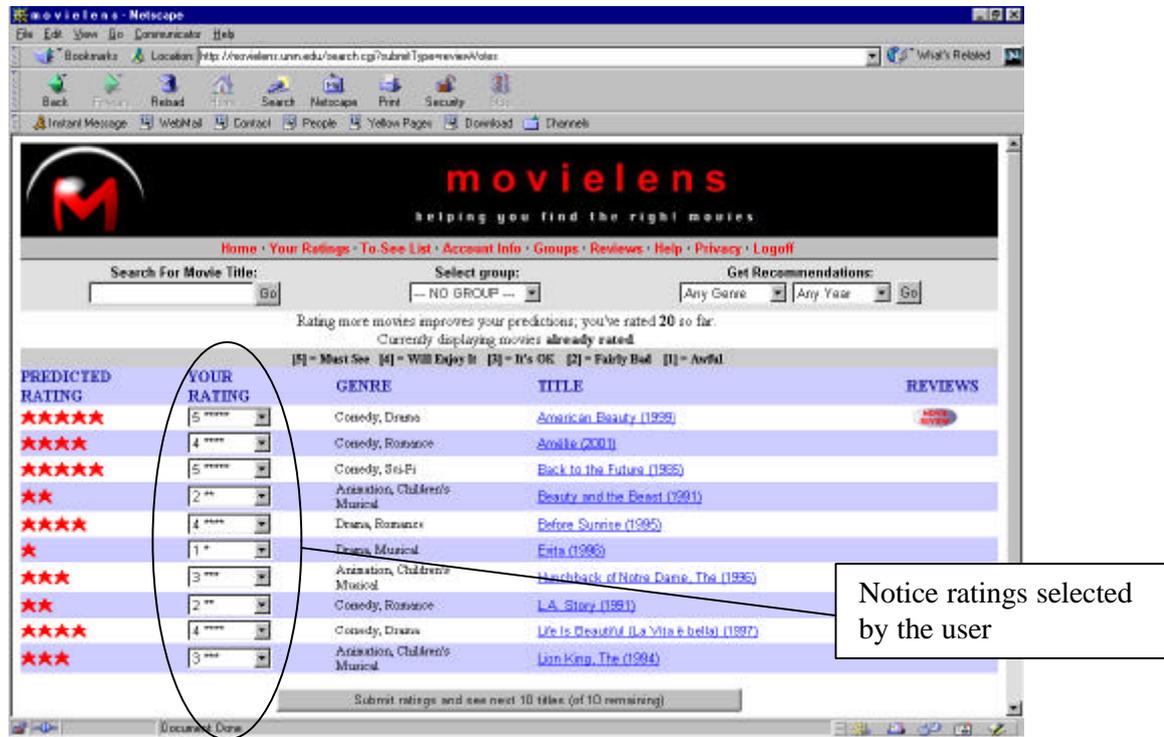


Figure 3: Film ratings for user profile in movielens.umn.edu

## 2.5 Examples

The mostly wide known examples of Adaptive Anchor Selection can be found in recommender systems. These systems provide a personalized list of recommendations based on the current state of the user profile. For example, the movielens recommender of the University of Minnesota ([movielens.umn.edu](http://movielens.umn.edu)) is an adaptive system that provides film recommendations based on ratings for a set of films that the user enters (see Figure 3).

Note that the user can access and modify his profile (Figure 3) changing ratings or adding new ones. For example selecting a rating 5 for the film *Evita* and a rating 3 for *L.A. Story*. If then the user starts again a recommendation request for action films of the 2000s, the “movielens” system’s response is shown in Figure 4. We can observe that some anchors have been removed and some new ones have been added. At the same time anchors are presented in a different ordering, i.e. link are sorted according to the ratings the user entered. This occurs because both patterns Adaptive Anchor Selection and Adaptive Anchor Sorting are often used in combination (see next section).

## 2.6 Consequences

- Local guidance is incremented as the user has less anchors to choice between.

- The disorientation factor will decrease.
- If the system evaluation of link relevance is inappropriate, the user will not be able to see and select some significant links at a given point in time.
- From a design point of view, complexity increases as the pages are not uniform and the user profile must be maintained and updated.

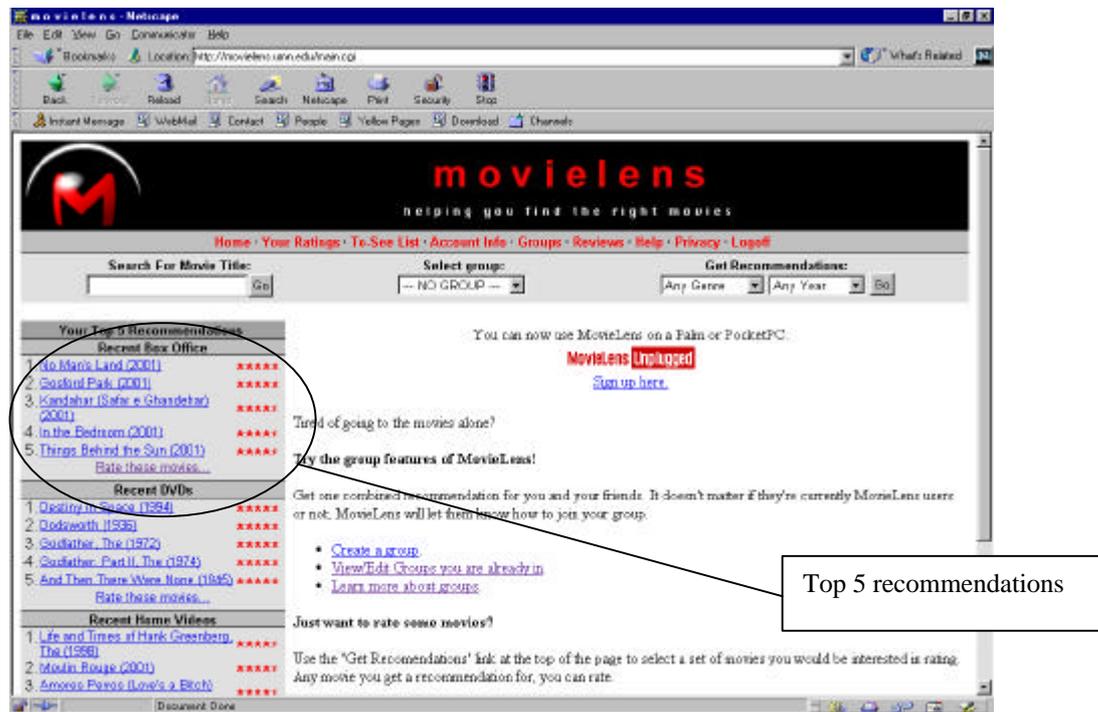


Figure 4: Anchor selection in movielens.umn.edu

## 2.7 Related Patterns

Adaptive Anchor Selection refines Link Personalization in its ability to adapt the navigation topology to the user profile. It also improves Structure Personalization to circumscribe the navigation space to the aspects the user prefers or he is interested in (see Appendix).

## **3 Adaptive Sorting of Anchors**

### **3.1 Intent**

Organize anchors so that they are presented in decreasing order of link relevance to the user.

### **3.2 Motivation**

As mentioned in previous patterns, many Web applications have dense connection topologies; a single page may point to dozens of other pages. We can reduce cognitive overhead applying Structure Personalization [Rossi 01] and the Adaptive Anchor Selection pattern that take into account the user profile. However, even in this case the set of recommendations may consist of a huge number of anchors. The user will be forced to read them all to find what he wants.

### **3.3 Forces**

- A single page may have many anchors pointing to different pages; the user may get distracted or overwhelmed.
- Selecting those anchors of his interest reduces the set, but it may still be too large.
- Dividing the page in parts, each one containing a sub-set of the anchors, does not solve the problem; it may complicate things further as it requires more navigation steps.

### **3.4 Solution**

Show the anchors in a page by sorting them according to the user profile i.e. for example the current user's interests. This way the user's attention will concentrate on the anchors that conduce him to the nodes that are more relevant to him at search time. This solution complements the one showed in the Adaptive Anchor Selection pattern by imposing a stronger constraint: not only we must select which anchors should be shown but also the order in which they appear in the screen. Notice that in different sessions of the same user, the same anchors may appear in a different order as shown in the examples.

Adaptive sorting of anchors is a pattern that is used to support adaptive navigation through the pages of a Web application.

### **3.5 Examples**

The movielens application offers to a registered user "Top 5" recommendations in the categories Box Office, DVD and Video. Here we can observe Adaptive Anchor Sorting in the first category when recommendation is get before and after user profile change (see Figures 4 and 5).

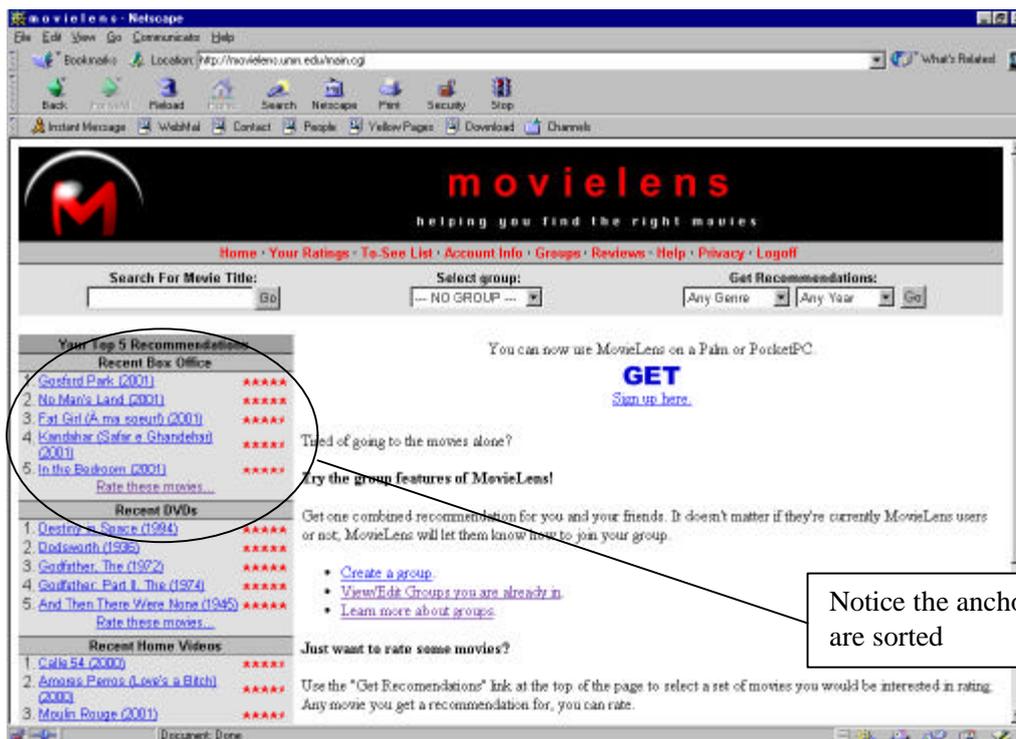


Figure 5: Sorting of anchor in movielens.umn.edu

### 3.6 Consequences

- The user can find his “best match” anchor easier.
- We need to provide both the algorithm for maintaining the user profile and the algorithm for ordering the anchors.
- The disadvantage of adaptive ordering is that each time the user enters the same page, the ordered anchors may be different. In such a case it is important that the system manages a history of pages presented to the user and assures that the user will see exactly the same page and with the same presentation when he goes back as supported by SmexWeb – a framework for Web learning applications [Albrecht 00].

### 3.7 Related Patterns

Adaptive Sorting of Anchors can be used together with Adaptive Anchor Selection to improve the list of anchors shown to the user. It can be also used in combination with Link Personalization (see Appendix) in the target page of the personalized link. Adaptive Sorting of Anchor is also a finer grained version of Content Personalization as what the user perceives depends upon his user profile.

## 4 Adaptive Conditional Fragments

### 4.1 Intent

Provide the user with dynamic personalized content in the information nodes. Appropriate content is selected or generated dynamically [Zdun 02].

### 4.2 Motivation

Many times the user access to start pages or portals that include a huge amount of information trying to reach a wide spectrum of users, i.e. trying to offer something for everybody or for every time a user access to the portal. But the part of a Web page that is visible at the first glance is a restricted area. If we want to include so much information it results in an overloaded page using very small fonts.

If we want to keep the user interested in the page, we should provide him with information that adapts to his interests. Adapting nodes content or structure is a good solution for solving this problem. In an electronic store for example the same product may show different prices according to the user shopping history (See [Rossi 01]).

The problem with content personalization is that it focuses on providing different values of the same node's attribute (e.g. special discounts on prices according to the user buying history as in [www.half.com](http://www.half.com)); however, it may happen that two different users are interested in slightly different attributes of the same object (more than in different values).

We can also present him with a customized node structure, i.e. two different users may also see different chunks of information as usually found in [my.xx.com](http://my.xx.com) sites ([mycnn.com](http://mycnn.com), [my.yahoo.com](http://my.yahoo.com), etc). The problem with this solution is that is rather static, i.e. it doesn't change while the user is interacting with the application and that it is coarse grained.

### 4.3 Forces

- Users want to read or see what they are looking for.
- Cognitive overhead in overloaded pages.
- Even defining different structures for different users may prove to be not enough.

### 4.4 Solution

Adapt node content selecting the information that is more appropriate for the user according to the knowledge the application has about him. In this way two different users will see completely different chunks of information (not only different links or values). Keep this changing content in the same place for every page in such a way that the user perceives a regular structure of every section of the screen and he does not feel disoriented.

In the best case what the user sees or reads is exactly what he is looking for, it may include pieces of information that are of his interest, but without chunks of information that disturb his attention. A special case is the devotion of spaces on pages to present content related to the user interaction.

Adaptive conditional fragments is a pattern that is used to support adaptive content in a Web application.

## 4.5 Examples

There are many examples of conditional fragments we can find in Web applications. Some of them show different variants of a subject to the user depending on the user profile, e.g. longer or shorter explanations, files in different formats for downloading, etc. However, other personalized conditional fragments are more sophisticated, like “page you made” in [www.amazon.com](http://www.amazon.com) as it is shown in Figure 6. The system keeps track of the products the user has visited and show them in the left part of the current page. This is a nice way of giving the user some kind of navigation history customize to the application logic.

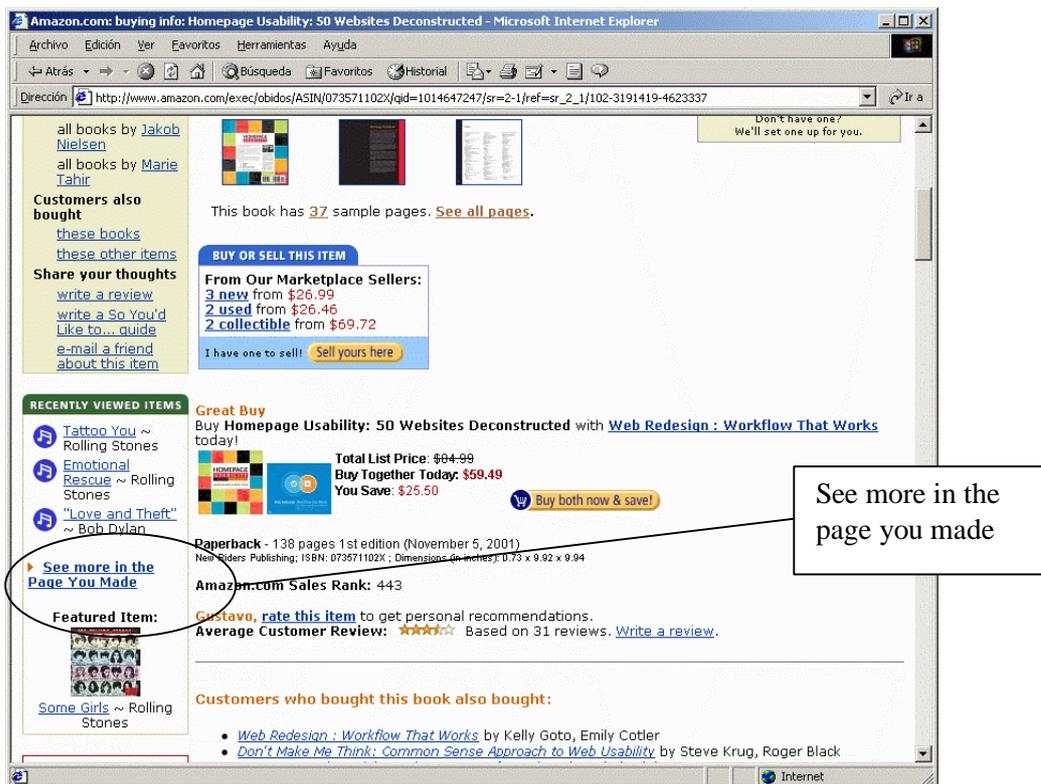


Figure 6: Conditional fragments in [www.amazon.com](http://www.amazon.com)

## 4.6 Consequences

- “Less is more” as the user is not overwhelmed with an overloaded page.
- The user may get confused as the same page has different content when he revisits the page.

## 4.7 Related Patterns

Adaptive Conditional Fragments is another finer grained version of Content Personalization (see Appendix).

# 5 Conclusion and Further Work

In this paper we have presented several patterns that address different problems related with building Adaptive Web Applications. These patterns convey design structures that are defined during the process of navigational design [Koch 02] and involve navigation objects such as nodes, links and anchors. Understanding the problems that we must face when building Adaptive Web Applications and being aware of good solutions to those problems is a key strategy for improving their quality and quality of use.

We are currently working in the definition of a more consistent system of patterns collecting design knowledge at different levels of abstraction (See for example [Rossi 01]) and in different design activities, such as conceptual and user interface design [Rossi 00]. This system should provide the background for a full-fledged pattern language for Adaptive Web Applications addressing from the early definition of requirements of personalization to the most technological implementation aspects of the adaptation process.

## Acknowledgements

We would like to thank our EuroPlop 2002 shepherd Paul Dyson who guided us through several iterations of our submission and provided many constructive comments. We thank also the members of the Writer’s Workshop for their helpful remarks.

## Appendix: Patterns for Personalized Web Applications

In this appendix we summarize those patterns that are heavily referenced in the paper; for the sake of conciseness we only describe the intent of the problem and the solution. A complete description can be found in [Rossi 01]. All of them consider a Web Application as a hypermedia network formed out of nodes and links. Nodes have perceivable attributes and anchors for links.

### Link Personalization

**Intent:** Adapt the navigation topology to the user's needs or preferences.

**Solution:** Define personalized links for connecting the nodes in the application. Though the information space does not change, some nodes may be easier to access to some users than to others.

**Example:** Recommendations in [www.amazon.com](http://www.amazon.com)

### Content Personalization

**Intent:** Provide the user with personalized contents in nodes.

**Solution:** Make nodes' attributes vary with the user. Treat them as a function of the current user.

**Example:** Personalized products' prices in [www.half.com](http://www.half.com)

### Structure Personalization

**Intent:** Circumscribe the navigation space to the aspects the user is interested in.

**Solution:** Personalize the structure of the Web site (or let the user do it); select which information objects will be shown, and which of these objects' attributes will be perceivable. Let the user access only those modules in which he is interested.

**Examples:** Home page organization in [www.my.yahoo.com](http://www.my.yahoo.com) and [www.mycnn.com](http://www.mycnn.com)

## References

- [Albrecht 00] Albrecht F., Koch N., Tiller T. (2000). *SmexWeb: An Adaptive Web based Hypermedia Teaching System*. Journal of Interactive Learning Research, Special Issue on Intelligent Systems/Tools in Training and Lifelong Learning. Kommers P. & Mizoguchi R. (Eds.), 367-388.
- [Bieber 97] Bieber M., Vitali F., Ashman H., Balasubramanian V., Oinas-Kukkonen H. (1997). *Fourth Generation Hypermedia: Some Missing Links for the World Wide Web*. International Journal of Human Computer Interaction, Vol. 47, 31-65.
- [Brusilovsky 96] Brusilovsky P. (1996). *Methods and Techniques of Adaptive Hypermedia*. International Journal of User Modeling and User-Adapted Interaction. Kluwer Academic Publishers, Vol. 6, 2-3, 87-129.
- [Campbell 99] Campbell, C., Maglio, P. (1999). *Facilitating Navigation in Information Spaces: Road-Signs on the World Wide Web*. International Journal of Human-Computer Studies, Vol. 50, 309-327.
- [Carmona 02] Carmona C., Bueno D., Guzmán E., Conejo R. (2002). *SIGUE: Making Web Courses Adaptive*. Proceedings of Second International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, LNCS 2347, Springer Verlag, 376-379.
- [Graham 02] Graham I (2002). *Three Patterns from a Pattern Language Concerning Web Usability*. Proceedings of EuroPlop'02.
- [Koch 01] Koch N. (2001). *Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Process*. PhD. Thesis, Uni-Druck.
- [Koch 02] Koch N., Wirsing M. (2002). *The Munich Reference Model for Adaptive Hypermedia Applications*. Proceedings of Second International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, LNCS 2347, Springer Verlag, 213-222.
- [Mladenic 00] Mladenic D. (2000).  
<http://www.cs.cmu.edu/afs/cs/project/theo4/text/learning/www/pww/index.html>
- [Paterno 99] Paterno F., Mancini C. (1999). *Designing Web Interfaces Adaptable to Different Types of Use*. Proceedings of Workshop Museums and the Web.
- [Rossi 00] Rossi G., Lyardet F., Schwabe D. (2000). *User Interface Patterns for hypermedia applications*. Proceedings of the ACM Conference on Advanced Visual Interfaces Sorrento, Italy, May, 2000.
- [Rossi 01] Rossi G., Schwabe D., Danculovic J., Miaton L. (2001). *Patterns for Personalized Web Applications*. Proceedings of EuroPlop'01, Rüping A., Eckstein J., Schwanninger C. (Eds.) 423-436.
- [Schafer 00] Schafer J.B., Konstan J., Riedl J. (2000). *Electronic Commerce Recommender Applications*. Journal of Data Mining and Knowledge Discovery, Vol 5 (1/2), 115-152.
- [Schwarz 96] Brusilovsky P., Schwarz E., Weber G. (1996). *ELM-ART: An Intelligent Tutoring System on World Wide Web*. Proceeding of Third International Conference on Intelligent Tutoring Systems ITS-96, LNCS 1086, Springer Verlag, 261-269.
- [Zdun 02] Zdun U., Vogel O. (2002). *Content Conversion and Generalisation on the Web: A Pattern Language*. Proceedings of EuroPlop'02.