

Web Usability Patterns

Fernando Lyardet, Gustavo Rossi¹
LIFIA, Fac. de Informática. UNLP
{fer, gustavo}@sol.info.unlp.edu.ar.
(¹) also at UNLM and CONICET.

Introduction

During the last four years we have been developing hypermedia applications (in CD-ROM and in the Web) using the Object-Oriented Hypermedia Design Method (OOHDM). OOHDM explicitly separates navigation from user interface design; this means that design decisions related with the navigational topology of the application are (in a broader sense) independent respect to those regarding interface issues (See for example [Schwabe96, Schwabe98]). Separating navigational from user interface design allows us to define different interfaces for the same navigation structure and maximize modularity.

We have mined many patterns that show recurrent design problems and their solutions both while developing the navigational architecture of the application and when building its user interface [Rossi96, Rossi97, Rossi99, Lyardet99]. The patterns contained in this paper belong to the hypermedia design pattern language that we have developed while mining patterns for web applications for some years. What is interesting in these applications is that they implement different navigational metaphors for traversing the information space (based generally on the simple hypertext metaphor with nodes, links and indexes) such as paths, sets of similar nodes, landmarks, etc.

Furthermore, many of this applications provide an effective user interface design, which consists not only of a set of astonishing graphics, but rather in their ability in handling effectively the several opposing forces involved when displaying information in a particular device like the computer screen. Such forces when not handled properly at the user interface find their resolution at the expense of degradation of information structure and navigational overhead [Lyardet98]. Thus, if we want to build web applications we have to learn how to organize the information, the navigational space, and develop an effective user interface.

The concept of Design Patterns originated in Architecture with Alexander [Alexander77], who saw the need to identify 'generic solutions' to re-use design experience to support the user-centered design of spaces that enable living Patterns are gaining popularity in the HCI community, following a CHI'97 workshop, reported by Bayle et al. 'Towards a Pattern Language for Interaction Design' [Bayle]. Bayle et al. define an Interaction Design Pattern (IDP) as a 'pattern that describes a connection between a repeatedly encountered problem and a solution that has been proven in the field, across time and circumstance'.

The patterns included in this work do not address the general problem of how to design a complex interactive software artifact. Rather than focusing on the design of traditional user interfaces, video games, and other such things, these patterns cover specific issues found in the interaction with web information systems (WIS). The patterns included in this work do not attempt to address detailed implementation issues, however.

A common source of concern in the pattern community is the need to avoid the tempting and simplistic renaming of the elements already known in the particular domain or application paradigm rather than concentrating on novel applications and successful solution schemes widely available, which are the result of the interaction among the base elements that conform those problem domains. In this context, design patterns are confused with the guidelines of

good design because patterns have a positive bias since they are usually focused on successful solutions. The patterns presented in this paper are focused on special interaction solutions adopted to overcome common pitfalls on web usability.

The Format

We decided to move a little from the traditional GOF pattern format to allow introducing some sections relevant to the HCI area. These sections ease to clearly state the kind of problem addressed by the pattern. We will introduce some variations following the work of Martijn van Welie and Jennifer Tidwell.

URL Affordance

http://www.mysite.com/something/

Affordance: "a technical term that refers to the properties of objects - what sort of operations and manipulations can be done to a particular object"

Donald Norman.

Problem How to design a useful URL.

Usability principles Affordance, visibility, convention.

Introduction URLs are an important part of the current web user interface because they are the mean to specify what we want to see and where. As such, it has to be designed to make it work effectively.

When people access the web, they usually face the challenge of finding what they want. Many times, users know exactly what they are looking for, and they have a certain confidence of its existence, and may also know the website address. Unfortunately, all this knowledge is not enough to ensure a successful search.

Since domains are being assigned at an impressive rate, finding a 'good name' is a time-consuming task. Thus, many websites decide to simplify the name selection and concatenate their name/s in a long string. This leads to amazingly long or extremely hard to spell URLs.

Furthermore, many websites rely on the dynamic generation of contents and they tend to underestimate the use of URLs by the users to access the information they need. Rather, these sites force users to access their homepage and then navigate or use a search tool to go to the desired place.

As an *example*:

["http://www.amazon.com/exec/obidos/subst/home/music.html/103-1306596-6860655"](http://www.amazon.com/exec/obidos/subst/home/music.html/103-1306596-6860655)

It would be easier to remember a URL like this:

["http://www.amazon.com/music"](http://www.amazon.com/music).

This not only reveals a more manageable way of expressing a URL, but also let the URL become a mechanism to access not only the website homepage but also specify the specific information or activity of interest for the visitor.

Solution Design the URL as a complementary tool for the user to find and access the contents of the website. The following is a set of guidelines to help the designer to achieve a better usability:

- A domain name that is easy to remember and easy to spell.
- Short URLs.
- Easy-to-type URLs.
- Persistent URLs that don't change to keep your customer bookmarks valid.
- Single case name (DoNotMixTheM.com)
- Use a spelling-checking web server to absorb common typos.
- Design URLs that help visualize the site structure.

Implementation **Design URLs that help visualize the site structure**

There is simple heuristic to organize URLs to help visualize structure according to the contents in the website:

1. Identify all the relevant keywords and kinds of elements (books, papers, music, etc).

2. Structure the content of your website in different directories, and name them with the kinds of elements you identified.
3. Set the default page of each directory to a page containing the index for that category.
4. Define mappings among keywords and elements, and establish your web server URI settings to enforce the correct redirection.

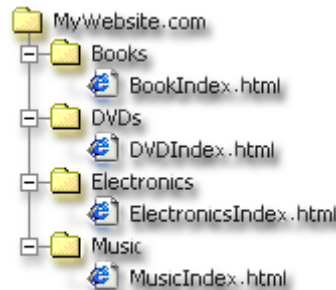


Figure 1. An example how to organize the contents of a website.

Use a spelling-checking web server to absorb common typos.

An example of such web server is the Apache web server [Apache]. If you have it installed, try checking http://localhost/manual/mod/mod_speling.html or search for *Module mod_spelling* on the documentation available.

Consequences *Positive*

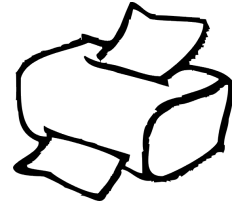
- Easier to remember URLs
- Reduces required amount of navigation to access a determined page.

Negative

- Force a wide hierarchy organization, based on a breath of choices. Its main disadvantage is that it may represent too many options. While the user only has to click once or twice to reach content, the time spending through all the initial choices in the home page may be counterproductive. Nevertheless, some web servers provide the possibility of setting virtual directories, helping providing simple URLs without forcing a specific organization of the information.

Known Uses There are several examples. One of them is Microsoft.com. In its website, Microsoft provide simple URLs to access the different products and technologies, regardless their actual organization, thus providing useful links (like www.microsoft.com/sql/) without setting constraints to the navigational structure.

Printer Friendly



Problem Provide the user with information in a suitable format for printing.

Usability principles
Introduction

The www, as the practical implementation of the mythical Ted Nelson's Xanadu Hypertext System where all the information and all the versions of the documents that contain such information were always available, should have turned the need of printing documents an obsolete practice. Far from that, due to the uncertainty of information persistence across time, users still print documents available on the web for archival purposes. Printouts are also preferable if the user has a paper-based filing system with folders or binders for all the information related to a project.

Furthermore, it is widely demonstrated that reading large documents of text from computer screens is an unpleasant task; forcing users print the documents for offline reading. Finally, users printout web pages to give to others to read, perhaps as handouts in meetings or seminars [Nielsen99].

Web browsers provide a limited support for printing web pages, with several limitations when printing websites with frames. For example, most browsers use the same typeface and font size or online viewing and printing, even though it is known that the two media require different typefaces. Also the preferred layout of a document is different: single column for online document display and two-column layout for printed media. [Nielsen99, Schneiderman94].

The capabilities of the users' browser generally dictate how a Web page will appear when it is printed. However, there are many things that you can do as a page designer to make printed versions of your pages look better.

Solution There are two strategies to solve this problem, and several considerations in common according to the designer needs.

Strategies:

1. If your site has several layers and contains information that you think would be useful to collect and print all at once, then take the initiative and create a single concatenated page of your entire site.
2. Another approach to supplying this functionality is to offer a downloadable, formatted version of the document to be printed. For instance, you could offer to download a Microsoft/Word or PDF version of the entire document. That way, you can have complete control over the printed appearance.

General Improvements:

- *If you need accurate document layout and spacing, use tables.*
You cannot control the fonts your audience will choose, the type of printer they have, or whether they decide to print in *landscape* or *portrait* mode. If you really need to control the size and spacing of text blocks, use tables. Tables let you specify the relative positions of text blocks and images to each other.
- *Pay attention to contrast.*
The majority of people will be printing your pages on a black and white PostScript printer. Color images are transformed into half-tone black

and white, so it is important that any critical information in your images be **high contrast**. That is, if your background is dark, make your text light and visa versa.

Contrast: difference in luminance between an object and its background on a display screen [Preece].

- *Do NOT make the background and informative element of the screen.* Remember that background images are not included on the printed page
- *Avoid setting a fixed font size to either paragraphs or titles.* Remember that users may have very different display resolutions. Therefore, what is clear to read at 800x600 may be almost invisible at 1024x768 or higher resolutions. Browsers have a common feature: "text size". This feature, that allows the user set the preferred text size, becomes completely useless when absolute font size is established. Rather, use the <H1...5> </H1...5> tags.

Implementation

Consequences *Positive*

- Users can print online documents without the limitations of visual displays.

Negative

- Increased complexity for building sites to provide such functionality.

If the website is not designed properly and relies on tight coupling of information and presentation (ie. the information is stored in html format), there is a risk of duplicate information to support this facility.

Examples There are dozens of examples available on the web, like Borland.com. zdnet.com, sun.com, *ibm.com*, delphi3000.com

Subjective Access



Problem Provide a customized view of the website according to the user's origin

Introduction E-commerce has flourished in the last three years, and a range of complementary services has been built around it. Some of these services, such as payment, are delegated to other more specialized companies, which already have their own websites.

As a result of this, users often get disorientated since the actual payment for a purchase or auction is made somewhere probably unrelated site. An example of this are auction sites, where the intricate mechanisms to ensure the payment on the one side, and the delivery of the offered merchandise on the other side, be accomplished in a successful manner.

International users always have to set up the correct language, even though the web server already knows which is the preferred language. As a result of this, if national language support is not very clearly visible, users may choose other, more simple or local website.

Solution Provide the users with a customized view of the website, according to their precedence and preferences, as a smooth transition to the new website.

Implementation To automatically set up the preferred language, most popular web servers already provide this functionality as a standard feature. This feature usually appears with the name of "Content Negotiation". This is because the web server considers the idea of resources.

A resource may be available in several different representations. For example, it might be available in different languages or different media types, or a combination. One-way of selecting the most appropriate choice is to give the user an index page, and let them select. However it is often possible for the server to choose automatically. This works because browsers can send as part of each request information about what representations they prefer. For example, a browser could indicate that it would like to see information in French, if possible, else English will do. Browsers indicate their preferences by headers in the request [Apache].

Consequences *Positive*

- Websites can focus in their business, delegating complementary functionality or services to other third-party, more specialized websites to accomplish a given task, leveraging the whole website architecture and maintenance.

Negative

- Potential disorientation of users, who don't expect to be redirected since they used to the traditional shop paradigm, where the purchase AND the payment is done as a whole, in the same place.

Examples www.ebay.com delegates in www.iescrow.com much of the complexities of payment and delivery. The iescrow website appearance is different is the user is accessing from an outbound link from Ebay, or by typing the iescrow URL in the browser.

Other interesting example is the IMP web mail that automatically detects the user-client char set and sets the default language automatically.

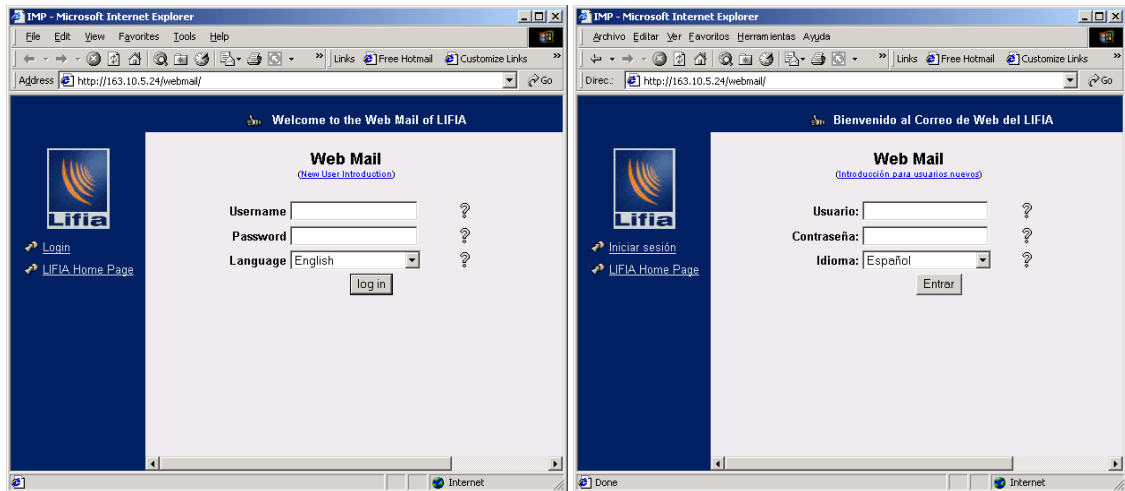


Figure 2. The language is set automatically according to the browser native language.

Typed Link



Problem Help the user differentiate among different kinds of links.

Usability principles Convention.

Introduction Oversimplified link anchors descriptions may cause hypertext readers to avoid links we very much want them to follow or vice versa.




Link anchors must simultaneously explain what will happen after they are followed and why readers ought to follow them; this dual message may be conveyed through multivalence, montage, or collage.

Unfortunately, in the www, links are also used to denote objects that the user may download, not navigation.

This also adds a new level of confusion since the same visual appearance is used to denote a section of a document (relative links), documents in the same website or to another website, and an arbitrary number of objects like Word, pdf, ps, txt, zip files, etc.

Solution Classify the kind of links by destination or object type, and assign an iconic decoration to provide the user with a visual cue.

Examples Thousands of websites instance this pattern.

The most common implementation is to provide a visual representation of the file format (.pdf , .txt , .zip , etc.)

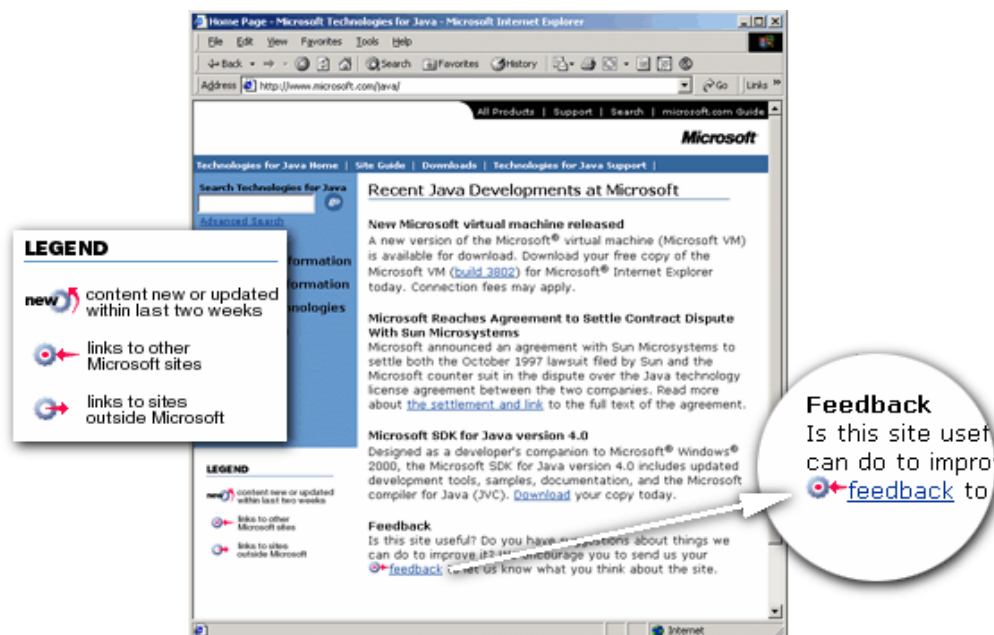


Figure 3. An example of Typed Link at Microsoft.com. In this example, the website defines a set of icons to help visualize outbound links and inbound links.

Bibliography

- [Alexander77] B. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel, "A Pattern Language," Oxford University Press, New York 1977.
- [Apache] <http://www.apache.org>. Apache Web server documentation.
- [Bayle] J .Bayle et al. 'Towards a Pattern Language for Interaction Design'. CHI'97 workshop report.
- [Preece] Jenny Preece et al.: "Human Computer Interaction". Addison Wesley, 1994.
- [Rossi97] G. Rossi, D. Schwabe and A. Garrido: *Design Reuse in Hypermedia Applications*. 8th ACM Conference on Hypertext Technology. Southampton, Inglaterra 1997, pp 57-66, ACM Press
- [Rossi99b] G. Rossi, F. Lyardet and D. Schwabe: "Patterns for designing navigable spaces" To appear in Pattern Languages of Programs 4, Addison Wesley, 1999.
- [Rossi99c] G. Rossi, D. Schwabe, F. Lyardet: "Improving Web information systems with navigational patterns" International Journal of Computer Networks and Applications, May 1999.
- [Rossi99d] G. Rossi, D. Schwabe, F. Lyardet: "Web application models are more than conceptual models". Proceedings of the First International Workshop on Conceptual Modeling and the WWW, Paris, France, November 1999.
- [Schwabe98] D. Schwabe, G. Rossi: "An object-oriented approach to web-based application design". Theory and Practice of Object Systems (TAPOS), Special Issue on the Internet, v. 4#4, pp.207-225, October 1998.
- [Schwabe 99] Schwabe, D.; Pontes, R. A.; Moura, I.; "OOHDM-Web: An Environment for Implementation of Hypermedia Applications in the WWW", ACM SigWEB Newsletter, Vol. 8, #2, June 1999.