Patterns for Adding Search Capabilities to Web Information Systems

Fernando Lyardet *, Gustavo Rossi [+], Daniel Schwabe **[1]
*[+]LIFIA Depto. de Informática. UNLP.
La Plata, Argentina
E-mail: {fer,gustavo}@sol.info.unlp.edu.ar
**Departamento de Informática, PUC-Rio, Brazil
E-mail: schwabe@inf.puc-rio.br
([+]) also at UNLM and COINCET.

Abstract

We present in this paper several patterns that can be used to improve Web Information System with search capabilities. We first introduce and justify the need for adding search functionality to Web applications; next we briefly explain the context in which we discovered these patterns. Finally, Web-search patterns are presented illustrating them with examples of successful information systems in the Internet

Introduction

The World Wide Web has become a popular platform for developing widely accessible applications. Moreover, the development of better and cheaper communication and storage technology, together with the exponential popularity growth of the web, brought us this vast ocean of information, with a few tools to help us find what we want.

Designing applications that support navigation and multimedia interfaces is a hard task, as we must provide meaningful navigation paths and we must build interfaces in order to avoid cognitive overhead. We have been mining patterns for designing these kind of applications for the last 4 years [Rossi96, Rossi97,Lyardet98, Rossi99]. These patterns provide guidelines to organize the information hyperspace and to design usable interfaces.

However, pure browsing is not always the best way to access huge information bases; in some areas, such as E-commerce, we should also provide search facilities to ease the task of finding the desired item. In this context, Web search engines are the most popular tools to find information on the WWW.

While Hypermedia and Web design methods have evolved and already provide good strategies for the design and organization of the underlying information space, designing the search functionality of a Web application is difficult and it is also a common source of widely spread misconceptions. Some of them have their roots in the common belief that adding a search service is reduced to the interface between an indexing engine and the website. When providing search capabilities for a Web application, we need to understand that user expectations and needs often fall well beyond the simple keyword search.

When the user begins searching, there will be certain information in the information space which will be highly relevant to his information needs, some of which will be useful, some marginally relevant and some not relevant at all. Only some of this relevant information will be returned by a search engine. It is possible to measure the performance of a search using two ratios: ***recall ratio*** and ***precision ratio***. ***Precision*** is the ratio of the total number of relevant document (or nodes, pages, etc.) retrieved, to the total number of documents retrieved, while ***recall*** measures the ratio of the number of relevant documents retrieved, to the total number of relevant documents. These concepts are shown in Figure 1.

[1] Daniel Schwabe is partially supported by a grant from CNPq.

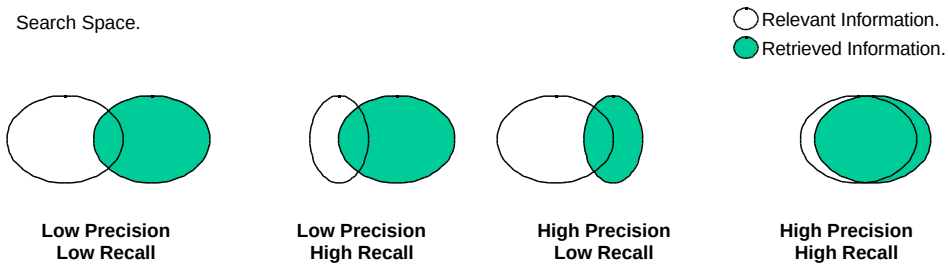

Figure 1- Precision and Recall.

In this paper we discuss some design issues that a novel designer must be aware of, and we show how expert designers solve them. We stress design issues that should be considered when adding search capabilities to a Web information system, by showing different kinds of search approaches. We show that these design problems are quite usual in current Web applications by illustrating them with currently accessible commercial information systems.

Most of these search strategies can be implemented using either standard technology or reusing existing search tools. We don't discuss the implementation of search algorithms: this issue may be the focus of a complementary pattern language addressing implementation aspects.

As previously mentioned, the patterns presented here belong to a hypermedia design pattern language that has been developed over these last years (See for example: http://pelican.info.unlp.edu.ar/patterns/); they address the more general problem of designing hypermedia applications (CD-ROMs, websites and web information systems (WIS))

These patterns are intended to form an Alexandrian pattern language, as found in [Alexander77], and not a catalog such as the one in [Gamma95]. This means in part that they are intended to be used together synergistically, in a way such that the whole is more than the sum of its parts. Like other such pattern languages, it does not break new theoretical ground or present innovative new techniques — more than likely, you have already seen examples of every pattern presented in this paper. Instead, it captures and encapsulates ordinary design wisdom in a practical and learnable way.

Selectable Search Space.

Intent.

Specify a category within which the search should be made or restricted to.

Motivation.

Many times users try accessing information through navigation. Nevertheless, as the information spaces become bigger, users are not always successful searching for the desired information. Therefore, having reached a certain point through navigation, they perform a search to find the desired information.

Every search activity pursues the goal of providing search results with high precision and high recall. It is also important to note that, with most search engines, it is possible to trade off precision against recall. If we increase the number of the documents retrieved, it is possible that the number of relevant documents retrieved also rises. At the same time, however, it is also likely that the number of irrelevant documents retrieved is increased, thus decreasing precision. At this point, the task of tuning a search can be greatly simplified with an analysis about the information being published in the website and how it is structured, in order to find groups of semantically related documents/nodes.

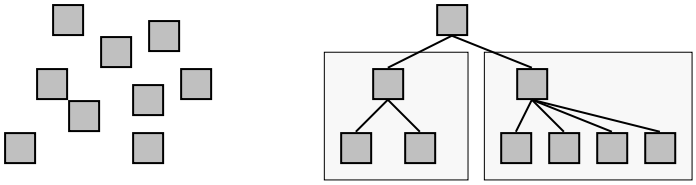

Figure 2. Being able to group semantically related information, allows better search results.

Many companies offer their products through their websites. When a user is looking for something concrete, it is useless to search in product areas different from the one the desired product belongs to. Suppose a person is interested in buying a certain book, and the selected website also sells videos and gifts; it would make sense to allow the user to select the category of products in which the search should be performed. In Figure 3 we can see how these categories can be set in a search server like Netscape Enterprise.

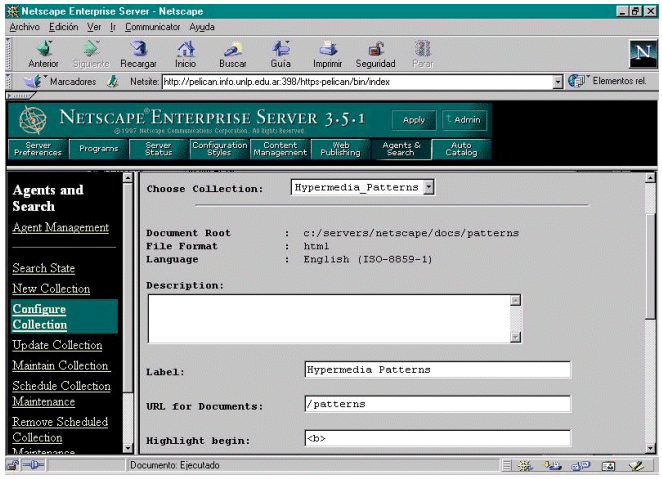

Figure 3. Setting a search category in Netscape Enterprise.

A more complex case appears when it would be helpful to combine search spaces or categories. Consider a website that already has its information organized into several groups. It is possible that some information relevant to a given topic may appear in more than one of these groupings. It is the case of many companies that apply a given knowledge or technology in several areas or products, and therefore there may be reports about several aspects of the topic of search covered in different regions of the websites or, even in other websites of the same company. In such a scenario, additional flexibility is needed [Explicar qual: combine groups], and since the website already has several categories, it would be impractical to consider all possible combinations of areas.

Forces.

- Users want to search over large search spaces.
- Precision and recall are mutually opposing measures, whereas effective search should have high precision ***and*** recall.
- Effectiveness of search can be improved by restricting the search space to known relevant subspaces.

Solution.

Provide the user with a mechanism to select which category (sub-space) the user is going to search into. There are 2 common variants to this solution, according to the desired functionality:

· Allow users select only one search category at a time. The requirement to implement this solution is very simple: it should be possible to split the information space into disjoint

sub-spaces. The same piece of information should not belong to more that one group at a time.

- Allow users to combine search areas. A grouping of checkboxes is presented to the user, with all possible areas of search. The user may select any combination of them. Although this approach is more flexible, it should be used only when the previous one is not feasible, since it requires extra navigation to a search page to select the desired categories

Consequences.

Advantages:

- Higher recall and precision rates. Surprisingly, a number of websites with huge amounts of information neglect to provide their users with ***selectable search spaces***. Rather, these sites rely on a simplified input field and offer only full searches over a website, which usually perform poorly since the search space is much larger. Furthermore, it forces the user to repeatedly refine the search by providing more detailed keyword information each time, in order to retrieve the desired information.
- Fewer searches are needed to find the desired topic, thus consuming less computing resources, and increasing user satisfaction.

Disadvantages:

- Categories usually must be determined manually.
- Having too many categories impacts in its usability Therefore, a certain balance among the granularity of the search spaces and their quantity must be achieved.

Examples.

Two examples of ***Selectable Search Categories*** are shown in Figure 4. The first one is amazon.com, which implements selectable spaces with a combo-box, while the second implementation in microsoft.com uses radio-buttons. Both implementations allow only one category to be selected at a time.

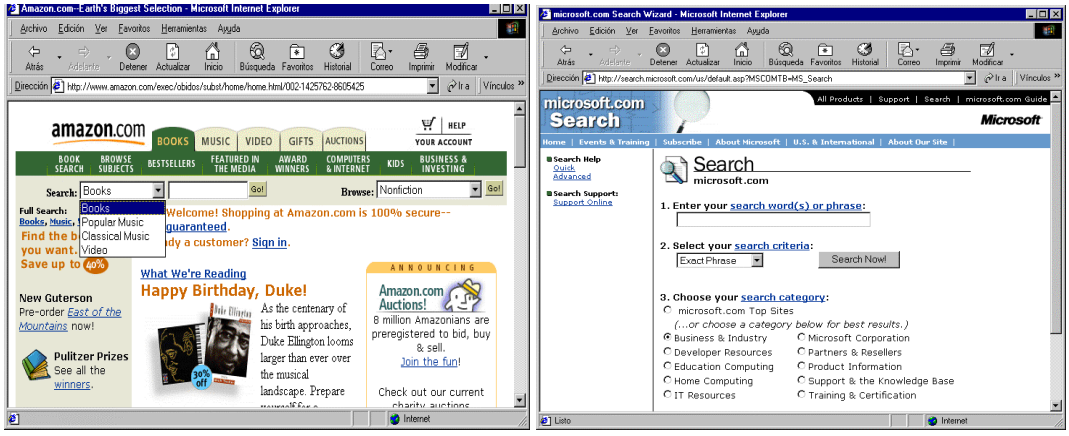

Figure 4. Two examples of *Selectable Spaces*.

An example of combinable search areas in www.microsoft.com is shown in Figure 5.

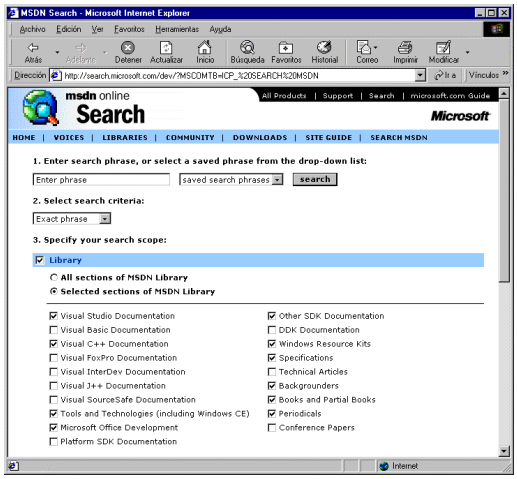

Figure 5. An example of *Selectable Spaces*, that provides the ability of combining categories.

Related Patterns.

Simple Search Interface, is usually enhanced with ***Selectable Search Spaces.*** It is also used in combination with ***Node In Context*** [Rossi99] to automatically set the searching category.

Selectable Keywords

Intent.

Suggest a set of keywords that are relevant to the results of previous searches, in order to improve the quality of subsequent search results.

Motivation.

The most common interaction that users have with search engines is typing relevant keywords about the topic of search. Nevertheless, since the web search space is very large, there may be several possible matches for a given word, thus a more sophisticated search must be specified. This almost always requires further keywords, in order to reduce the search space and allow better matching. For example, suppose we want to find information on databases; if we use a search interface with this keyword, we will obtain hundreds of related pages when perhaps we only want information on one particular kind of database. Unfortunately, using more than one keyword in this first search may not yield reasonable results.

Forces.

- Users want to search over large search spaces.
- Arbitrary keywords may yield undesirable items, whereas pre-defined ones can increase effectiveness.
- Users don't always know what keywords may be used.

Solution.

Provide the user with a list of the possible keywords, according to the search already performed by the user. He should be able to include any combination of the suggested keywords without typing, since typing is an error-prone activity.

The usual interface strategy for this solution is the following: place a list of keywords near the field in which he user enters the keywords to perform search; this list should be selectable and combinable (e.g.: by using check boxes).

Consequences.

Positive:

- Users may complete a search without typing.
- More successful results, thanks to the improved matching.

Negative:

- Users must perform at least one search, before the system can be able to suggest any further keyword.

Examples.

Excite.com provides the user with a set of keywords related to a search previously made, (see Figure 6). In Figure 7 we see a less flexible implementation in Altavista.com

Figure 6 . An example of *Selectable Keywords* in Excite.com.

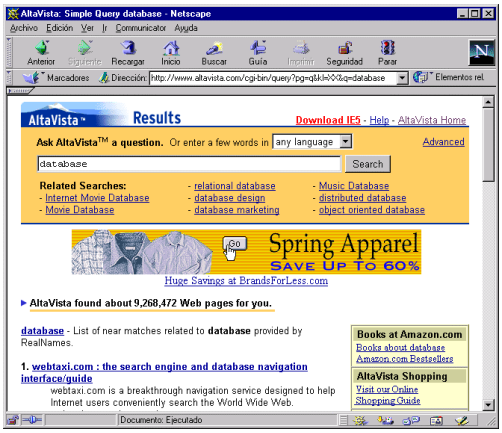

Figure 7. Another implementation of *Selectable Keywords*. The search engine proposes several related pre-built searches.

Related patterns.

Behavioral grouping [Garrido97]. The search/keyword field and the list of selectable topics should conform/be arranged in a unit logical unit. Other related pattern is ***Structured Answer.***

Structured Answer

Intent.

Provide the user with an organized information report as the result of a query.

Motivation.

The Human Computer Interaction research field has studied for years communication between computers and their users. One of the essential aspects of interacting with computers is the problem of providing meaningful information to users, rather than “raw data”. This aspect has been always an issue, because human communication requires an important piece of information: the notion of context. Context is vital in human communication, to allow the interlocutor a chance to correctly understand a given information [Schneiderman].

In the vast extent of the web space, search results may include a huge amount of relevant information links. Nevertheless, since the amount of information is so large, further explanation regarding the nature of the information being retrieved must be provided in order to allow people to conform themselves a context to understand the data.

Suppose for example that we submit a query using the keyword: “Objects”. In a standard search we would obtain thousands of links pointing to different kinds of pages that include the word “Objects”. Even if we use ***Selectable Keywords,*** the search result may be rather flat.

Therefore, a list of links is not enough. Further information regarding the information being retrieved is needed to assist a user deciding on the right link.

Forces.

- The results of queries may include many items.
- Summary information is needed to help users choose between alternatives.
- When examining one item, easy access to the other items should be available.

Solution.

Provide the user with a more structured result that organizes the information such that it is easier to help the user decide which choice is the most suitable. Such additional information may be grouped in different categories:

· Information about the item that has been found: Include a brief abstract of the target page, which may be automatically done by the search engine.

- Information about links to other information that may be related but that is automatically chosen by the system.
- A taxonomy organizing the information space in meaningful sub-spaces.

Consequences

Positive:
- Users have a more complete view upon the information retrieved and therefore are able to do reduce navigation and search effort.

Negative:
- Since extra information is provided, the amount of information received by a novel user may be overwhelming.

Examples.

In www.excite.com, and www.yahoo.com we can find good examples of Structured Answer, where lots of information about the topics found is provided like precision ratio, category, related topics, news involving some of the keywords, links to related books and where to buy them. In www.altavista.com we also find a graphical representation of the result space, and related topics. In Figure 8 we see the solution in www.yahoo.com for the query on objects. Notice that the results are grouped in grouped in sub-sets.

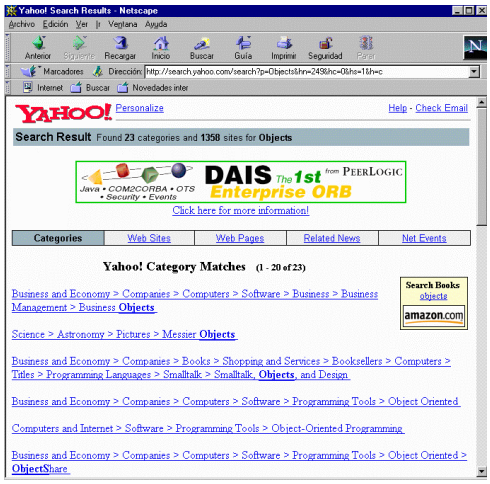

Figure 8: A taxonomy organizing the result space

Related Patterns:

Structured Answer plays a dual role with respect to Selectable Keywords. While Selectable Keywords allows reducing the search space, Structured Answer gives more information about the result space, by organizing results in different categories.

Selectable Search Engine

Intent.

Enhance a website search capabilities by delegating the actual search service to more sophisticated engines, which may be able to produce a ***Structured Answer*** with more detailed or better formatted information.

Motivation.

The web offers a huge space of information that is far from being organized, and it grows at extremely high rates, suffering unforeseeable number of changes, where information is added or removed, classified and reorganized, with highly heterogeneous criteria, To help users find a given topic in such amount of data, the web is being scanned continously by indexers, info-bots, spiders, worms, etc. in order to update their own information databases and provide the users with accurate search results.

Nevertheless, as the amount of information is so high, no single search engine is nowadays able to claim complete web knowledge; therefore the same search query performed in different search engines provides different results. When a topic is not found within a given searcher, it may be necessary to repeat the same task with other search engines. This is already a well known fact for experienced web users, and as a result of this, several companies offer in their homepages the ability to perform a search using the users favorite web search engine, without having to navigate to other sites. Even web browsers such as MS-Internet Explorer already provide this feature, as a way to leverage the task of searching on the web. The most recent versions of the Macintosh OS includes this feature as part of the operating system search functions.

Forces.

- Powerful search engines are expensive and resource consuming
- Index maintenance is also resource consuming
- Search technologies are rapidly evolving, and it may be time consuming to keep track of them.

Solution.

When developing a Web information system, link it to an existing search engine that allows external websites. Doing this, it is possible to add full search capabilities to a website by delegating the search to those engines that allow being used remotely. Therefore, the scenario described is twofold: users are able to select search engines without having to navigate to their respective websites and, on the other hand it also allows implementing a sort of "zero-cost" search service in a website.

Consequences

Positive:

· Development times and costs are reduced and we can trust on the power of existing engines

Negative:

· Existing engines may be rigid for the task in hand. We should evaluate the cost/benefit relationship of delegating searches in terms of functionality of our Web site or application.

Examples

Many Search engines can be used from other applications. For example: Amazon, Infoseek, Altavista, Excite, etc.

This pattern can be found in many websites that implement ***Selectable Searchers***: NetscapeCenter (www.netscape.com), MSN (www.msn.com), www.webtaxi.com, etc. In addition, www.uiuc.edu, where users are able to search over the campus website using an external search engine. In Figure 9 we can see an example in MSN

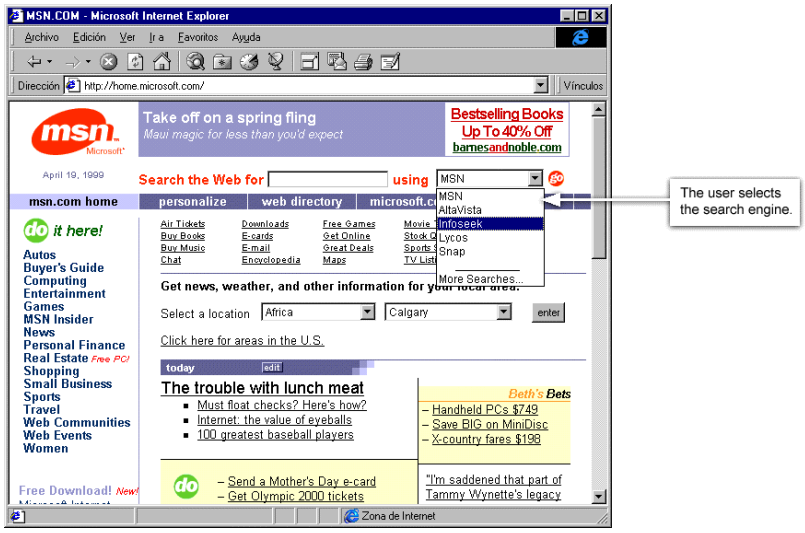

Figure 9. An example of *Selectable Searchers*.

Related Patterns

Behavioral Grouping [Garrido97]**.,** since the ability to select a search engine must be given locally to the keyword search field.

Simple Search Interface

Intent

Provide users with powerful, yet simple, search mechanisms.

Motivation

Effective search mechanisms often involve complicated, sophisticated query formulations, using logical operators, query refinements, field qualifiers, etc... On the other hand, very often a simple query satisfies the user needs. As an example, consider a developer accessing a website looking for some technical specification, although it may be possible to access the page using navigation, a quick search is done to help going straight to the required information.

A similar scenario appears when accessing a website to buy for instance, a book which he knows in advance. In this last case, the usual task is not navigation but searching, since it is more natural to think of a search of a book in a bookstore than browsing all categories of all books in order to find a given title. In the described examples, there is a common characteristic among both websites: search capabilities are used often, or even more than navigation to access the information required. Therefore, if some extra navigation is always necessary to access to the search facility, it becomes a source of unnecessary navigational overhead. Furthermore, as search engines become more sophisticated, the amount of options available to the user to perform a search may transform a simple activity into a more complex and time-consuming effort for simple queries.

Forces.

- Naïve users typically do simple searches
- Simple searches are often enough for the needs of the user
- Advanced search capabilities require more expertise from the user
- Formulating advanced queries takes more time, and requires a more sophisticated interface that is justified only when really needed

Solution

Provide a very simple, straightforward search interface – one field, no operators - that will satisfy a very large proportion of users; in addition, provide a link to a more refined search

interface, where all the "bells and whistles" are available. Provide a consistent way to move from one to the other.

Consequences

Positive:

- Quicker access to the search capabilities provided by the website
- Less navigational overhead for simple queries.

Negative

- Adds an extra element to the user interface, which in turn may result in more complex interfaces for novel users.

Examples

There are several examples of the Simple Interface pattern on the Web. For example Altavista.com, yahoo.com, or softseek.com websites provide a simple search interface but also more sophisticated search options are available to the user in a separate page. We can see examples of those Web-sites in Figures 10 y 11.

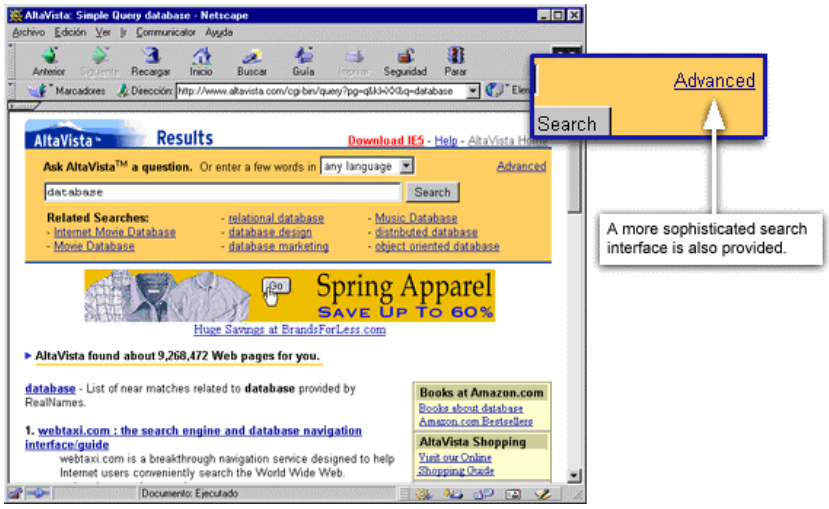

Figure 10: An example of *Simple Search Interface*: in www.altavista.com . Note that it is possible to access to a more sophisticated search facility if desired.

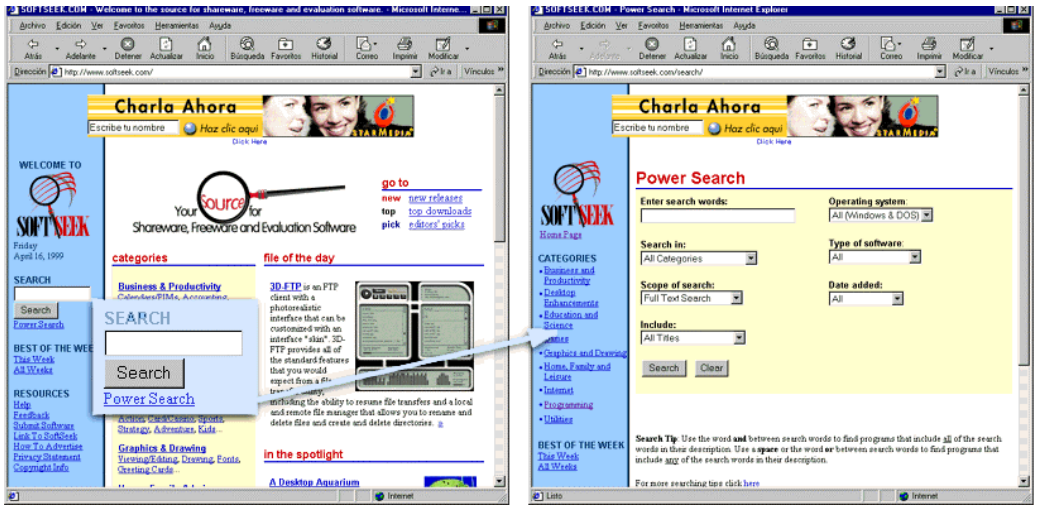

Figure 11. Another example of Simple Search Interface in www.softseek.com. A simple search interface is always present for the user, but also a more complete search engine in provided.

Related Patterns

Behavioral Grouping [Garrido97].**,** since the ability to select more complex search options must be clear and visible in the context of the search field.

Bibliography

[Alexander 77] Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King e S. Angel: "A Pattern Language". Oxford University Press, New York 1977

[Garrido 97] A. Garrido, G. Rossi and D. Schwabe. " Pattern Systems for Hypermedia" In Proceedings of PloP'97, Pattern Language of Programming, 1997.

[Gamma95] Gamma E.,Helm R., Johnson R., Vlissides J."Design Patterns. Elements of Reusable Object Oriented Software.", Addison Wesley, 1995.

[Lyardet98] Fernando Lyardet, Gustavo Rossi and Daniel Schwabe. "Using Design Patterns in Educational Multimedia applications". AACE EDMEDIA Conference, Freiburg Germany.

[Lowe] Lowe D.,Hall W., "Hypermedia & the Web. An Engineering Approach". John Wiley and Sons. 1999

[Rossi96] G. Rossi, A. Garrido, S. Carvalho: "Patterns for object-oriented hypermedia applications". In Pattern Languages of Programs II, Addison Wesley, 1996.

[Rossi 97] G. Rossi, D. Shwabe and A. Garrido : Design Reuse in Hypermedia Design Applications Development Proceedings of ACM International Conference on Hypertext (Hypertext'97), Southampton, UK, 1997, ACM Press.

[Rossi99] G. Rossi, D. Schwabe and F. Lyardet: "Patterns for designing navigable information spaces". To appear in Pattern Languages of Programs IV, Addisson Wesley, 1999.

[Schneiderman] Schneiderman, Ben. "Designing the User Interface. Strategies for Effective Human Computer Ïnteraccion". Second edition. Addison Wesley, 1992.

[Schwabe 96] D. Schwabe, G. Rossi and S. D. J. Barbosa. "Systematic Hypermedia Application Design with OOHDM". Proceedings of Hypertext'96 (HT96). Washington, March 1996.

[Schwabe96] D. Schwabe, G. Rossi and S. D. J. Barbosa. "Systematic Hypermedia Application Design with OOHDM". Proceedings of Hypertext'96 (HT96). Washington, March 1996.

[Shubin97] Hal Shubin and Margarrt M. Meehen: "Navigation in Web Applications". 1997, ACM Interactions. Volume iv number 6.