User Interaction Design Patterns for Information Retrieval Systems

Martin Schmettow
Fraunhofer IESE
Fraunhofer Platz 1
67663 Kaiserslautern, Germany
martin.schmettow@iese.fhg.de

1 Motivation

The field of information retrieval systems (IRS), like web search engines, hypertext, database and also search facilities embedded in e.g. web shops and even desktop applications, has some special characteristics regarding the design of user interaction. On the first glance the interaction is very simple: Man asks, machine answers. When doing a conventional requirements analysis and design concept, you would draw two use cases and design two UI masks and you're done. But at a closer look the design of usable, enjoyable and successful IRS is by far not trivial.
Instead it seems that those factors that make up a good interaction design for an IRS are nearly as complex and subtle like human interpersonal communication itself [1]. Fortunately the field of information retrieval (IR) got a lot of attention since the early days of human-computer interaction research.
While more technically oriented scientists and designers engaged to optimize the hidden qualities of IRS, esp. the optimization of recall and precision, others concentrated on enhancing the user-system interaction. Of course there are shades and strong interrelationships between both directions and as a result, today, there are tons of scientific results on the design of IRS and their user interfaces and there are as well lots of successful solutions on desktop computers, information systems and in the WWW.
The aim of this pattern language is to collect known good solutions for interaction design of IRS. It is mainly inspired by recent pattern languages on UI design like [2-4].

2 Overview on the pattern collection

The pattern collection presented here consists of 10 patterns on two levels of abstraction. To say that clearly: It is by far not complete regarding the known best practices in UI design in IR. Instead it is meant as a starting point.

Figure 1 IRS UI Patterns arranged by primary patterns, with "Recommended to Use With" Relations and Support Points for User Activity.

2.1 Pattern Language Features

This starting collection should already show the most important characteristics of a more complete pattern language in this domain:

- Relations between patterns
- Several sublanguages defined by one primary pattern
- Secondary classification based on a psychological theory in the domain of information retrieval

A frequently used relation in UI pattern languages is "Recommended to use with". While this relation is often not stated formally, it can be derived from informal statements in the "solution" or "resulting context" (e.g. [3]). It is for example more explicitly stated in the "related patterns" sections of [4]. This relation helps to find patterns that fit nicely together or complete each other. Some relations could already be identified with the IRS UI patterns (see Figure 1).

The pattern language consists of two levels of abstraction (so far). The abstract level is based on two primary patterns (an idea taken from [3]), where the more concrete patterns are related to the primary patterns in the manner of the concretization of a principle.

2.2 Theoretical Background

Most UI pattern languages today are not based on any sound theory from HCI or any other related discipline. Still, a theory can be very handy, in that it at least gives a classification and related vocabulary of what is going on in a specific field.

The two abstract patterns basically suggest basing the design of an IRS UI on some properties of sound human conversation. That is: to be *expressive* (Engage in Rich Conversation), *with a historical and social context* (Exploit Social information) and *robust against errors.* The last principle is at the moment hypothetical, in that it is not clear, if this principle can be defined by a crisp sublanguage or if it is an additional aspect in other patterns. And indeed, the primary patterns' main purpose is to be a heuristic for finding new concrete patterns rather than being used in concrete design situations.

All other patterns can directly be arranged to one of the two primary patterns (see
Figure 1, where the primary patterns constitute the top row).

When designing an IRS for a specific purpose, it is often a good idea to ask potential users of what problems they have in using comparable systems. This could help to prioritize design solutions that compensate for specific user problems. To facilitate this I chose a theory on human cognitive activities during the use of IRS [5] This approach identifies the four activities:

1. **I**dentify Problem: The user forms a goal for interacting with the IRS from the information problem in an external task. Example: *"I have the idea of writing a paper about UI patterns for information retrieval systems. Does such a collection already exist?"*

2. **A**rticulate needs: From the usage goal the user derives more concrete needs expressed in concepts from the domain in question. Example: *"If such a publication already exists, I will find it by combining terms from the pattern domain (pattern, collection, language, problem, solution) with terms from the information retrieval domain (information retrieval, query, recall, precision).*

3. **Q**uery formulation and reformulation: The domain terms are used to build a query term using the specific syntactic, logical and linguistic features of the IRS at hand. Example: *"I will combine the terms inside each domain by "logical-or" and connect the resulting two terms by "logical-and". I will not use placeholders, because flexion is automatically handled by the IRS."*

4. **E**valuate results: The user reviews the results of his query and decides if it suffices his needs. If not, he proceeds with rearticulating his needs. Example: *"Lots of publications on patterns in information retrieval! But they all seem to refer to topics of pattern-matching algorithms. I consider a reformulation, this time explicitly excluding terms from the algorithm domain."*

Because nowadays IRS are very often integrated in other systems and processes, a fifth activity is added, namely:

5. **F**urther use of results: The user transforms the relevant results and queries for the use his external tasks or for later situations. Example: *"Information retrieval algorithms was not what I wanted right now, but it may be good for later use. I will export the results to my favorite bibliography tool."*

The patterns can now be classified regarding if they facilitate a user activity by e.g. making it easier, more efficient or less error-prone. The classification according to user activities is visualized in
Figure 1 as the colored dots beneath each Pattern.

2.3 Scope, limitations, complements

As already stated, this collection is meant as a starting point with a theoretical underpinning and heuristics to identify further patterns. Fortunately, there exist other pattern languages, which already fill some of the gaps. Especially the collection of Wellhausen on UI design for searching [6] is a good complement, in that it covers the topic of keyword based searching in detail. Most if the searching patterns in patterns can be classified to the activity "Query (Re)Formulation". The outsider pattern CORRECT ME IF YOU CAN would nicely fit in the language of Wellhausen as it clearly treats keyword searching. On the other hand it is one candidate for the hypothetical category of patterns for error robustness (cf. 2.2).

3 The pattern collection

3.1 Engage in Rich Conversation

Alternative Names

- *The answer is 42. You have to pose the right question.*

Problem

In order to efficiently solve the users information problem the user and the IRS have to gain a mutual understanding of what the user wants and what the system can do.
A fundamental problem of IR is that information problems of users are usually open-ended and the knowledge needed to solve the problem is distributed between the IRS and the user.
In more detail the user in general knows about:
- his tasks and goals
- domain concepts and their relationship
- his own level of domain expertise

Whereas the system might "know" about:
- what domains are covered in the database
- how specific a term is for one or more domains
- what alternatives are there to use the system
- how terms correlate in the set of documents

If the IRS behaves in a monotonous and (in a social sense) reserved way to the attempts of the user, both resources of knowledge lie idle instead of gaining a "common understanding" between system and user. As a result the user has to work out his mental model of the system and accordingly the correct formulation of needs and queries tediously by trial and error. This is very demotivating, so the user might loose interest and give up early.

Forces

- Users like their systems to show natural aspects of conversation.[1][2]

---

[1] At the time of writing this ELIZA celebrates its 40th anniversary.

- Information problems are often much underspecified.
- The user performs best, when he exactly knows the capabilities of the system
- The system performs best, when it exactly knows the goals and interests of the user
- Initially the user and the system have very poor knowledge about each other

Solution

Let the system behave like a human partner, who hypothesizes, poses questions and offers alternative in order to find out what the users goals, problems and questions are and to expose it's capabilities to the user.

The system should react to the user in a conversational way. It should try to understand, what the user really meant. Since this is usually not unambiguous from the view of the system, it should try to hypothesize, what the user means and pose its hypothesis as additional questions to the user.

The system should then be able to give the user advice for alternatives and for selecting the right search strategy.

In a more technical sense the system has to analyze the user input and its own output and sense certain patterns typical for user problems. It should then have "knowledge" about how the user could achieve better results. This knowledge should not only be "declarative", in that the system gives the user some abstract advice (as classical expert systems would do), but the system should be able to sketch out or even initiate the alternative action.

This is not restricted to critical situations, where a user might fail, but also, for example, to arouse interest of occasional users passing by.

Nevertheless, the self-initiated behavior of the system to maintain the conversation with the user should be carefully balanced with the primary function of the system, which is to deliver content. While in critical situations (where the user most likely is prone to failure) the hints of the system can be more revealed (like the door sign of an emergency exit), in normal situations this is clearly "smalltalk" and should be considered as such.

Resulting Context

- You end up with a significant more complex user interface and dialogue structure.
- You also need a lot of additional logic behind the scenes.
- But you have an IRS, which motivates users to have a second look and not to give up to early.

Related Patterns

As this is a primary pattern it is concretized in the patterns:
- TEACH THE TAXONOMY CASUALLY
- RICH RESULTS
- GOOD HIT GOOD EXAMPLE

---

[2]   This is admittedly a claim of the author. The interactive help function of Microsoft Word© is often given as a counterevidence. But in the opinion of the author the dislike of "Karl Klammer" is mainly caused by its annoying behavior and not its approach to interact naturally. How much would anyone working concentrated on a task like another person who notoriously interrupts and tells how to make things the right way?

Known Uses

- Ebay always offers alternatives and refinements to their users:
  - suggests categories, where many hits are found
  - suggests refining search by several attributes (price range, location etc.)
  - suggests several broader searches, when the result set is too small
- Pandora.com is a music database, where artists are classified according to several criteria. The service engages the user in a dialogue to find out, what the user's preferences regarding music are:
  1. The user initially enters an artist he likes or is interested in
  2. The service plays a song from this artist
  3. If the user likes this sample, she can ask for similar artists
  4. The service presents samples from other artists based on similarity. It also gives a rationale why this sample was chosen.

Figure 2 The conversational character is claimed to be the basic idea behind pandora.com.
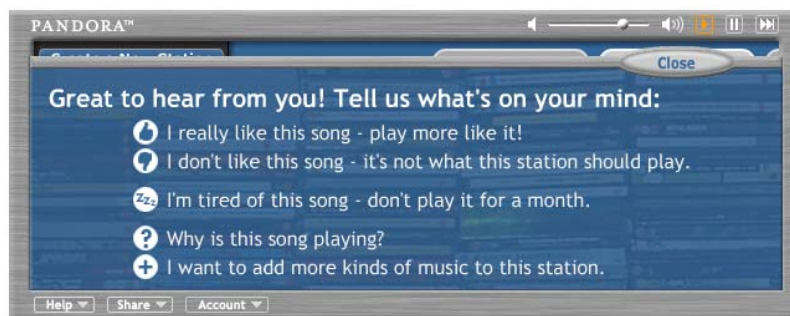
Figure 3 The user always has the chance to tell the music retrieval system his opinion on the content presented latest.
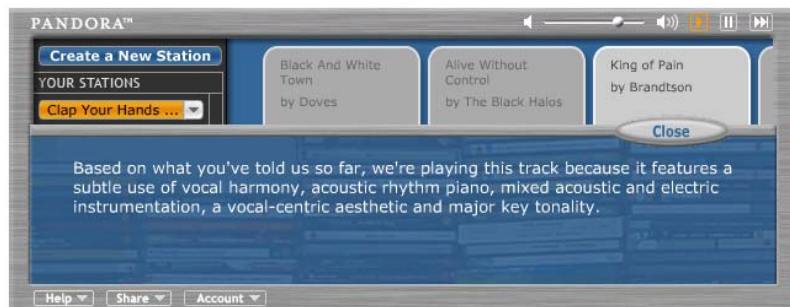
Figure 4 As the music retrieval system also explains its rationales, system and user can gain a mutual understanding.

- Other examples are given in the more concrete patterns below.

3.2 Teach the Taxonomy Casually

Problem

Users often stick to keyword search despite there is a category search available, which might be more appropriate to solve the user's problem.
So how can you motivate users to switch to the category browsing mode?
Many users start by keywords if you let them and it is always a good idea to let them take the initiative. But keywords are likely to fail because of different reasons, especially:
- the keyword has homonyms, which gives a lot of irrelevant results
- the keyword has synonyms, which lets lots of treasures buried

With only keywords the user is left alone to find the optimal keyword expression, which includes finding the right keywords and combining them in an optimal query expression. This leads to, what most users have severe problems with: Debugging.
As an alternative you can provide some way of browsing the content through a category system (or more general: classification). But this requires the user, who may be unfamiliar with the categories, to explore and learn the database instead of just quickstarting.

Forces

- occasional or new users want to quickstart instead of first learning
- category systems require users to learn the taxonomy
- keyword search requires the user to learn the syntax, otherwise it is quite restricted

Context

- the information source is large
- and has a keyword search
- and has some kind of classification (e.g. a taxonomy)

Solution

Guess categories based on their frequency of occurrence in the keyword query result and offer them as entry points to category browsing.
When the user makes his best guess keyword search, give her the chance to learn about the classification of topics. Calculate the most frequent classification categories from the result set. Suggest them to the user for exploration and refinement of search.
If the user chooses an offered category, switch the UI into browsing mode. This will most times be a hierarchical list of items presented according to the patterns SITE MAP [4] or OVERVIEW BESIDES DETAIL [7].
Make the suggestions clearly visible, but at a place clearly separated from the main result set. Otherwise the user might confuse it with the actual results or experienced users are annoyed, because they have to scroll to find their results.
Screen space is rare: So you should restrict the suggested categories to just a few (or use MORE OF THIS). You can of course gain your suggestions from more than one category system. If they have a quite distinct meaning or structure from the viewpoint of the user (e.g. taxonomy vs. keyword) separate them, otherwise you can also mix them up.
There are many ways to have a classification system on board your IRS. The most popular are taxonomies, where sets of categories are refined in a hierarchical manner. In library systems

keyword vocabularies are well established. In these days keyword systems are brought to a broader audience in the form of folksonomies[3].

Resulting Context

- Users will notice your suggestion and some of them will switch to an intertwined search strategy of keyword and browsing. So you should at first design the browsing mode of your IRS very carefully - it will gain more attention.
- Second you should explicitly enable this intertwined search strategy by also enabling a smooth switching from browsing to keyword search.
- Another good effect can be, that more users learn your category system, because it is revealed to them piece-by-piece. This will make their use of the system much more pleasant and efficient. But you are in the obligation to design and maintain your classification accurately.

Related Patterns

- This pattern could also be derived from EQUAL OPPORTUNITY [2]. There it is argued with giving feeling of control to the user.
- Use keyword search together with Only here! to achieve a round-trip experience between keyword and category search.
- If there are so many categories to offer, use MORE OF THIS in favor of screen space.
- OVERVIEW BESIDES DETAIL, SITE MAP and several other more general UI patterns are ways to organize the browsing mode (see Solution)

Known Uses

- Ebay presents the categories, where many items are found, with each result. This is even working iterative: When the user chooses one of the presented categories, again the best matching subcategories are presented. This is excellent for finding the category, where the best balance between recall and precision of results is. A special variant is used in the last step. In addition to the suggested categories, a form pops up, which enables the user to refine his search according to category-specific product attributes.

---

[3]   From wikipedia.org (06/03/29): A "**folksonomy**" is a collaboratively generated, open-ended labeling system that enables Internet users to categorize content such as Web pages, online photographs, and Web links. The freely chosen labels -- called tags -- help to improve search engine effectiveness because content is categorized using a familiar, accessible, and shared vocabulary. The labeling process is called tagging.
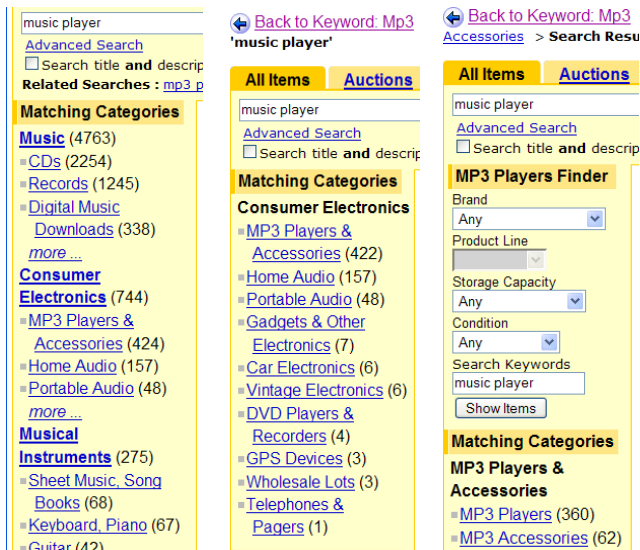
Figure 5 Ebay teaches the category to the users

3.3 Rich Results

Problem

Since the search request is never perfect, the user has to review the result set in order to pick out valuable results or - in an extreme case - decide to start a reformulated query.

In many cases it can be expected, that the user is confronted with large result sets when searching in an IRS. This has several reasons:

- The database is very large
- The user doesn't have expert skills with formulating very exact queries
- The user doesn't want to miss any hit, so he uses broad queries

The resulting problem is that users have to decide for what is a hit by looking at every single result. This is cumbersome, when the user has to open every single document (e.g. by clicking on it) to get sure.

This problem becomes even unacceptable, when the user has to pay for the actual documents (e.g. on a pay per view).

Forces

- Identifying a result as hit or miss is effort for the user
- Users avoid effort
- The use of the system could be paid on a pay-per-view base, what makes opening the wrong documents expensive.

Solution

Make sure you know the users common criteria to determine if a result is a hit and show these in the result set. Also make it possible to manipulate the result set based on users' requirements.

The results should be displayed in an expressive manner. It should be carefully considered, which elements or properties enable the user to quickly decide, which item is promising and which not. There are some very common elements, which should most times be considered, namely titles (more general: identifiers) and full text fragments where the users keywords were found. But, depending on the kind of information, user preferences and even query mechanisms other properties can be appropriate (have a look at the Known Uses).

The properties should of course be selected carefully, because screen space is always rare and strange properties could confuse the user. Especially if there are expert or technical users, their interest to see (or show) technical details of the system should not be confused with details that really help the common user to find their things quicker. Make sure to exactly know the users' goals, tasks and expectations!

Here are some examples, of what properties should definitely be shown in several domains:

- *Unclassified text documents*: If you are only dealing with unclassified text documents this patterns reduces to the pattern RESULT SET [4]. Show title and text fragments.
- *Webshops* should at least state: article name, article type, price and availability.
- A *database of patterns* should at least give the patterns names, their category, if one exists, and the main problem and solution statements.

Resulting Context

- As the shown properties require space, the result presentation will as well be longer and might make navigating the result set more cumbersome.
- If several properties are shown at every single result, the presentation becomes more complex. The layout grid and naming of properties should be carefully designed.

Related Patterns

- When showing meta data properties with results, these could be made active links for further search (cf. GOOD HIT GOOD EXAMPLE). This is also supported by the fact, that with rich results the user gets a good picture of the specific hit.
- This pattern generalizes SEARCH RESULT [4] in that it generalizes the latter to all kinds of data items, structured and unstructured.
- When the data is more of a tabular than a full-text type you could consider a SORTABLE TABLE [7] to show the results.

Known Uses

- In EBay, where time is an important factor of business, start time, end time and remaining time is displayed. This nicely adapts to the users different tasks: When he wants to know, which offers have been added since his last visit, he chooses "sort order: newly listed" and the start time of offers are shown in the result set. If instead he wants to put a bid on the soonest available item, he chooses "ending soonest" and the time column switches to the remaining time.
- The ACM Digital Library has a promising results presentation, where the most useful information on a document is presented in a concise, yet compact manner. However, there is room for improvement. They could highlight the search term and make the

meta data (authors, keywords etc.) links.

3.4 Good Hit Good Example

Problem

The formulation of an information need requires the user to reason in "free-recall"-mode about the domain concepts. This is error prone and demanding.
When the user has identified her information goal, she has to work out an idea of what specific properties of a document might fulfill her needs. Therefore she has to first identify, which properties are relevant at all. Then she has to consider a possibly large number of combinations of these properties to identify those, which are promising. For text documents this can require a sound knowledge of terms and concepts in the domain at hand. And additionally the whole cognitive process has to be conducted in the so called "free recall" mode[4], which is highly demanding and error-prone.

Forces

- users are not always able to state precisely what they are looking for
- users have restricted domain knowledge
- free recall tasks are hard
- advanced meta data search is difficult for many users
- it is easy for users to identify a hit if they actually see one

Context

- The IRS can determine the similarity of documents (e.g. based on metadata or text similarity measures)

---

[4]   Example for a *free recall* task: "Tell 10 kinds of fruit, you know!" versus a *cued recall* task: "Which of those 20 words are fruits? (apple, cucumber, lettuce, cherry, tomato, steak, …)"

Solution

Let the user reformulate his query by just pointing at a single result, which makes up a good example of what she was looking for.

Users (or humans in general) are very good at pointing at something and say: "This is what I want!" when they actually see it. So you can make each single hit from a result set an example for at least one set of documents with similar properties.

There are basically two variants to achieve this:

1. *Singular:* Treat a single property of each result as a query. The similarity is then defined by this property alone. For example: Show the author of the document and make it "clickable".
2. *Holistic:* Treat the whole result with all its properties as an example for another query. This can be done with a link called "Find similar documents". This of course requires a complex and well calibrated algorithm to compute similarity between documents. It is important that this is a measure corresponding to the user's sensation of similarity and not just a technical gimmick. In other words: Know your users requirements and mental models to find the right similarity measure.

Further on there are (at least) two variants of presenting the set of similar documents:

1. *Instant show:* The list of similar documents is presented instantly at the result in question. This can be done for example by a pop-up window, a section in the detail view of the document or by using MORE OF THIS. This variant works best with the holistic similarity or when the expected set of similar documents is small.
2. *New query:* The set of similar documents is presented as a new result set. This works better with large sets and when you have a singular property similarity and thus can tell a clear query for this result set (Example: "You searched for: author Weizenbaum").

In general singular property examples with a new result set is a good starting point. It maintains flexibility in that the user can choose what property is relevant. When time goes by, you might get a better picture of what a good example really is to your users and can derive a holistic similarity measure.

Resulting Context

- As the shown properties require space, the result presentation will as well be longer and might make navigating the result set more cumbersome.
- If several properties are shown at every single result, the presentation becomes more complex. The layout grid and naming of properties should be carefully designed.
- If variant 2 (holistic) is used, there might be no intuitive way to explicate the query behind the emerging result set.
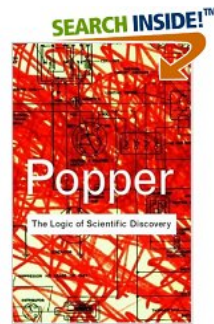
Related Patterns

- If you have holistic similarity criteria, you can use More of this to give the user an idea of this function by a few examples and at the same time access to all similar documents that exist.

Known Uses

- Ebay offers similar articles, if the users bid was unsuccessful (holistic, instant show)

- Amazon.com lets the user find other books of the same author (singular, new query) or to related topics (holistic, new query)

3.5 Only here!

Problem

You have a category mode and a keyword query mode. How is it to establish, that users can first browse, then refine by keyword?
Category browsing is a nice thing, because it is a cued-recall situation to the user (see above) and makes it easy to eliminate irrelevant results without risking possible hits. But even in much elaborated category systems the user might in the end have a large set of possible hits left. But users often search for very specific items. With a category mode alone, the user would then stuck to manually inspecting the whole set of items in the category.

Forces

- Users refine their search stepwise
- It is hard to maintain a very detailed classification or ontology
- Users needs are often very specific

Context

- a category system, which can be browsed by the user
- a full-text search engine

Solution

In browsing mode make it an option, if a query is executed on the whole set or just the category at hand.
Put a simple search field at the top of your window (or reuse an existing one). Let the choice to the user, he might also want to escape from this category by keyword search. There are different options to enable this choice:
- You could have a checkbox beneath the search box, which says something like "search only in this category". Have it checked by default.
- You could provide two search boxes: One general (e.g. at the right top of the screen), which always searches the whole database. Another one right beneath the classification path or category name

Make sure to state very clear (at least in the result set), that the search was restricted to a specific category.

Related Patterns

- To establish a roundtrip-experience combine this pattern with Teach the Taxonomy Casually, this describes how to smoothly switch from query to browsing mode.

Known Uses

- With Ebay the user can browse into (or otherwise reach) a category and has then the choice to search by keyword there (which is the default). What is especially well designed here is that there is an always present simple search box at the top right corner. So users can always escape by keyword search without worrying, if the only-here mode is active or not. Another interesting feature is that the user cannot only decide between "here and all", but can also choose one of the next ancestor categories.
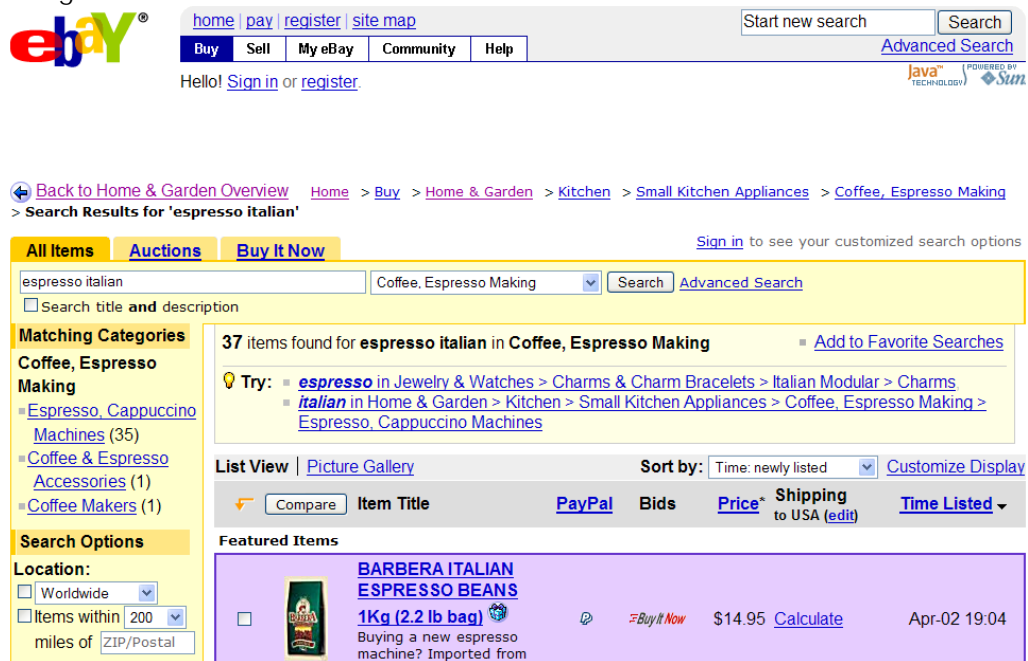
3.6 More of this

Problem

How can you convince the user to follow suggestions on similar documents or alternative queries without wasting too much screen space?

There are many ways to infer similar or related documents for result sets as well as single documents (e.g. Teach the Taxonomy Casually, Good Hit Good Example). Because these related results were not what the user explicitly asked for, it is not a good idea to use a lot of screen space to show the whole list of them. On the other hand just placing a link to e.g. a list of similar documents might not be convincing enough.

This is especially true for holistic similar documents (see GOOD HIT GOOD EXAMPLE), because the concept, these related documents were identified with, often cannot be precisely communicated to the user. (E.g. you should not call it "Find documents which are very close

in a Bayessian Belief Network). But just saying "Find similar documents" might not be convincing enough, so the user would avoid it.

So how can you strongly invite the user to use the side trails without cluttering your screen space?

Forces

- Screen space is rare
- Examples are a good way of giving an idea of a concept
- Users tend to follow paths they can understand and fully control
- Related result sets can lift treasures for the user

Context

- There are some features to suggest alternative searches or results to the user.

Solution

Show an example of the feature and put the full list behind a link.
Present related result sets in the following style:

- Headline expressing the concept (e.g. "Users who visited this document also visited")
- Show 3-5 results in very short (meaning no extensive meta data etc.)
- Link to a complete result set (e.g. "Show more items")

Related Patterns

- The underlying principle of is similar to Good Hit Good Example, in that a more or less abstract concept is replaced by some examples or placeholders, which the users can easily evaluate.
- This pattern should be used, whenever the IRS makes proactive suggestions to the user (TEACH THE TAXONOMY CASUALLY, GOOD HIT GOOD EXAMPLE, SHOW THE TREASURES)
- It is the favorite way of instantiating the instant view variant of GOOD HIT GOOD EXAMPLE.

Known Uses

- A9 search engine does it in a mouse-over tool tip to find similar documents.
- Ebay does it sometimes: Less restricted queries are presented together with some examples.
- Citeseer (citeseer.ist.psu.edu) shows several categories of related documents this way and uses two steps: More and All.

Figure 6 "More of this" and "All of this" at Citeseer

3.7 Exploit Social information

Alternative Names

- NNUP's Not User Profiling

Problem

Previous knowledge helps a lot to guess the optimal response regarding the user's expectations. But previous knowledge of the individual user might not be available.
An IRS always has to present information to the user, which he most likely finds interesting. This is not trivial in case personalized user profiling is not allowed (e.g. due to privacy policies) and thus the system doesn't know anything about the user's preferences. Another case is, when the usage is typically very infrequent and doesn't allow robust user profiles. As a consequence the system is left alone with the vagueness and impreciseness of the user's actual query and cannot attract new users with "surprisingly" good guesses.

Forces

- Occasional or new users need special guidance or attraction.
- It is hard to hypothesize based on a single user's behavior.

Solution

Track and analyze the behavior of many users in order to bring anticipatory features to your IRS. Base the IRS anticipation of users needs upon what many users preferred in the past.
You can exploit the fact that humans are social beings. When they are unsure what to do, they often start by imitating others. In a more technical speaking there is a high probability that preferences or strategies shown by many users in the past are as well useful for the current user.
Therefore you should try to track the preferences of users in your IRS, analyze them and distill some good suggestions, optimized results or other guidance for the current user (see SHOW THE TREASURES).

This might be a very general guidance, for example offer very often accessed content to new users first, and also very situation specific, in that the systems gives some hints to the user of what past users did or preferred in exactly the same situations. Additionally the systems knowledge of users behavior can be used to optimize the system response completely hidden to the user, e.g. by "social" ranking algorithms. In other words: Not only the behavior of end-users of the IRS counts, but also the behavior of content providers, if they can refer their content to other pieces of information (see RANK BY AUTHORITY).

You will probably achieve the best results, if you not only track user's trails, but make use of a utility function for each trail. This can be done by asking the user about the success of his current trail, but you should prefer non-reactive criteria to not interrupt users.

Resulting Context

Dependent on how you exploit social knowledge and how explicitly you bring it to the user interface, you end up with a user interface, which

- Better guides the user in that it makes recommendations, when the user might have no goal or plan himself
- Prevents user failure, in that successful paths of others are suggested more prominently
- Delivers more often valuable results from the viewpoint of the user

In the first two cases, the system will be perceived as a more "intelligent" interaction partner, because it shows its capability of social reasoning.

You should also be very aware of privacy issues.

Another problem is to create local maxima, that is, when an acceptable but suboptimal choice is promoted by the system and thus stabilizes too early – or just think of it as the lemmings situation.

At last the collection may take some time until it leads to an improvement, so be patient or deactivate the feature during the learning period.

Related Patterns

This pattern is nearly as abstract as the Bayesian probability logic[5]. It is concretized here to the patterns RANK BY AUTHORITY and SHOW THE TREASURES. These are at the moment representatives for a larger set of patterns derived from the ideas of EXPLOIT SOCIAL INFORMATION.

RANK BY AUTHORITY represents the aspect of exploiting the content providers behavior and Show the Treasures emphasizes end-user behavior.

Known uses

See concrete patterns below for examples.

---

[5] Bayesian probability is here meant as a synonym for "The more prior knowledge, the better your guess!"

3.8 Rank by Authority

Problem

The sort order of result sets is an implicit filter to the result and is thus quite critical for the perceived precision of retrieval. The IRS has to guess about that without knowing the individual user's understanding of relevance.
When users are searching by keywords in large full-text sources (like the web), they often get an overwhelming result set. Most of the results cover the topics the user is looking for, but are too specific to be of interest to the user. In many cases, what the user wants is an introduction or overview of the topic. And, especially when he is not a domain expert, he relies on the validity of content.

Forces

- General topics are by chance of interest to more people
- Validity of content cannot always be guaranteed

Context

- A very large full-text information source
- The information pieces are connected by references or links

Solution

Rank results high, which have the most references from other sources drawn on them.
You should exploit the opinion of the content providers they implicitly express, when link their content to another source. In general one can assume that a link to another piece of content is a positive appraisal. Accordingly, a piece of content having drawn many links onto it is most likely a valuable source of information.
Additionally a piece of content that covers a topic in broad is more likely to be referenced than a very specific document.
Authoritative sources are those, which are referenced or linked to by many other sources. When the references are done by independent authors (like in the web or scientific literature) the authority is based on social acknowledgement, which is probably the best choice.
If the references are constructed by controlled content classification (like in an editorial process), this is also good, because the source is then likely to be a good starting point to discover the topic in all its facets.

Resulting Context

The result of this pattern does not change anything to the user interface. Thus it is not a feature, which immediately catches the users' attention. But in the long run, users will experience that the intelligent ranking of results makes their searches more efficient.

Related Patterns

This pattern greatly enhances the quality of pure keyword based search. Because it does not require an explicit user request to sort in a useful way, it is especially well suited, when a minimum of user input is likely – that is the case with a simple keyword search.

Known Uses

- Many web search engines use the Kleinberg algorithm [8], esp. Googles PageRank® is a variant.
- Citeseer (citeseer.ist.psu.edu) counts the references of scientific papers for ranking. It even offers to choose between authorities (documents most often cited) and hubs (documents, which cite many authorities). While often cited documents are likely to have a high reputation, documents citing many authorities are have often the character of an entry point, an introduction or an overview, which is a useful property as well. Citeseer also shows a graph of citations over the past years. That helps the user to estimate the up-to-dateness of a document.

Figure 7 Citeseer uses references between documents to compute relevance measure. There are even different measures available for different user goals.
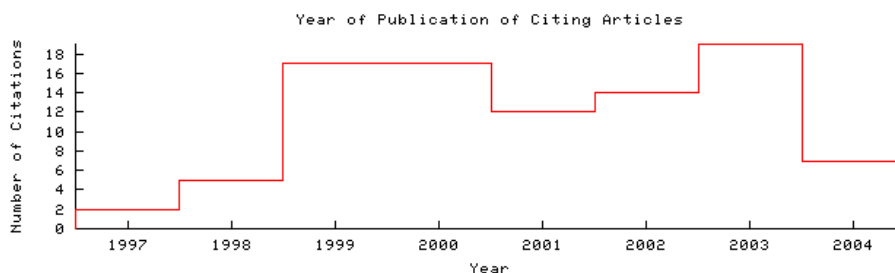
The graph only includes citing articles where the year of publication is known.

Figure 8 Citeseer's timeline graph of citations gives a historical view back on fields of interest.

- In science the citation index is an accepted measure of a persons or organizations reputation. This is also associated with the so-called impact factor of scientific journals.

3.9 Show the Treasures

Problem

Users might use the IRS for very narrow-minded tasks. But you want them to have a further look and develop new goals to achieve with the IRS.
When users are looking for something very specific, they will most likely follow a very narrow search strategy (e.g. looking for a specific book by author and title). In these cases, they will not appreciate the other treasures in the IRS. This behavior of users can also occur, when they have been using the IRS for a longer time and didn't notice it's grow (a paradox of the active user [9]). Also users with a low attitude to technology might be too focused on their task instead of showing explorative behavior.
Of course, the IRS could randomly suggest pieces of content to the user. But, when the content offered does not catch the user's interest, it will be perceived as (probably unwanted) advertising. So how can you catch these users attention on other content without intimidating them with what they could consider unwanted advertising?

Forces

- Task orientation helps users achieve their goals quickly
- Explorative behavior helps users to set new goals
- Interruption annoys the users

Context

- Your IRS is capable of fulfilling are large bandwidth of interests.
- You have a measure of popularity for the content.
- You have previous information of the user's interests.

Solution

Make a best guess of what a user might be interested in and proactively offer some results. Choose a moment, where the user is not engaged in a task.
At first you need a method to guess, what proactive results the user might enjoy or find interesting. You can easily achieve that by a general measure of popularity (like best sellers, top 100). If you have tracked past queries of the user or have a user profile, you can enhance the chance of offering something of interest to the user. But the suggestions should *not* be based on the most recent queries. Instead, select pieces of content, that are not too close to what the user recently searched, as this would not broaden his mind.
The second challenge is to offer the "treasures" in the right moment. When the use is eagerly searching for something, he is most likely not in the mood for exploring. But, especially, when he has successfully finished a particular task, he might be open for suggestions. You can also consider choosing the moment, when the user enters the IRS, because then he might have a goal but is not already involved in action. The additional advantage is, that users, who enter with the intention to explore get good starting points.

Resulting context

- You might need a starting page, where proactive content is offered.
- Users, who are very aware of privacy issues, might get skeptical.

Related Patterns

- This pattern is an alternative to GOOD HIT GOOD EXAMPLE , when the primary goal is not to support the current engagement of the user, but to make him find new goals with the IRS.
- When screen space is rare it can be used together with MORE OF THIS.

Known Uses

- Amazon makes proactive offers based on bestseller lists and user profile when entering the portal and offers based on user profiles, when the user has put an item into his basket.

Known abuses

- Amazon makes proactive offers after selecting a book for the basket, but this might be too early, as it happens before the user sees the effect of his action – namely the content of the basket.

3.10 Correct me if you can

Problem

Correction of spelling errors is easy for an IRS, important for the quality of results, but potentially unexpected and confusing for the user.[6]
Perceived low retrieval rate of keyword search because of spelling errors has always been a problem with query based information retrieval. As users usually don't know, if there actually exist content matching their query, they cannot easily decide, if an empty result set comes from spelling error or simply missing content.

Forces

- Spelling errors are frequent slips of humans
- Humans perform poor at detecting spelling errors, esp. their own
- Spelling errors are fatal for stupid machines

Context

- You have a full text query over a large database
- There are occasional users, who don't know which content is actually there

Solution

Inform the user about possible spelling errors and suggest a correction.
Analyze the query at least by a spellchecker, which signals, if it doesn't know a word and proposes a spelling correction. It is even better to analyze the frequency of a term. If it's very rare and there is a similarly spelled word with a much higher frequency, this should be

---

[6]   Number of books published by Jacob Nielsen: *0*

proposed. The highest value can theoretical be achieved, if the context is taken into account for identifying spelling errors, e.g. the other terms in the query or the users previous queries. But do not silently correct user queries. Especially with rare words (e.g. product vendors) this can lead to completely unexpected results (e.g. "Rebach" –> "Rehbock"[7]). As a paradox consequence users will suspect that he made a spelling error: This is because typing a word is such a highly automated process which can at the earliest be evaluated after execution.

Related Patterns

- This pattern is especially useful together with keyword searching, as this mode is extremely prone to spelling errors. Thus it is tightly connected to the patterns of searching [6].

Known Uses

- A9 and Google execute queries with spelling errors, but give a good visible and polite hint with most times useful correction.
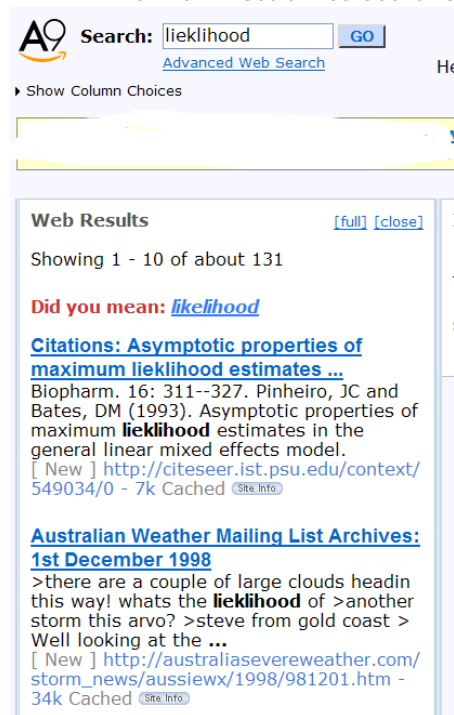

Figure 9 The A9 search engine executes queries with typos, but makes a best guess suggestion (www.a9.com). Notice, that in the two top results the content author made the same mistake as the searching user!

Known Abuses

- When Ebay introduced spelling corrections of users' query in early 2006 it used silent error correction for a short period of time. This was sometimes quite confusing, because many Ebay users search for brands. As these often are proper names or even fantasy words, the automatic error correction was prone to "false alarms".

---

[7] Rebach: computer vendor
Rehbock: german for roebuck

4 Future prospect

It is myth, that interaction design (in general) can be treated and developed independent of other layers of a system, e.g. application logic or storage [10]. In other words: UI design decisions will most likely influence other layers, which can result in a catch-22 of what to design first. In [11] an approach is presented to solve that problem, in that several UI design patterns can be linked to distinct architectural patterns. For the IR patterns it is planned to classify them according to what technological concepts are required to realize the interaction solution. This will be done by consulting the more technically oriented references in the discipline of information retrieval.

The initial patterns presented here are believed to be quite obvious or proven by some well known existing implementations. When the collection proceeds, there will be pattern candidates that are not so clear. These solutions should then be verified (or disqualified) by investigating existing empirical studies or by at least finding a plausible rationale based on general HCI findings or cognitive psychology. The latter can partly be achieved by intensifying the reference to the theory of cognitive activities in IR, which for the moment was mainly used for the purpose of classification and guidance.

References

[1]   P. Watzlawick, J. Beavin, and D. Jackson, *Pragmatics of Human Communication*. New york: Norton, 1967.

[2]   I. Graham, *A Pattern Language for Web Usability*. London, 2003.

[3]   J. Tidwell, COMMON GROUND: A Pattern Language for Human-Computer Interface Design. http://www.mit.edu/~jtidwell/interaction_patterns.html (last accessed: 03 2006)

[4]   M. v. Welie, Patterns in interaction design. http://www.welie.com/ (last accessed: March 2006

[5]   A. Sutcliffe and M. Ennis, Towards a cognitive theory of information retrieval. *Interacting with Computers*, vol. 10, pp. 321-351, 1998.

[6]   T. Wellhausen, User Interface Design for Searching A Pattern Language. Presented at Tenth European Conference on Pattern Languages of Programs, Irsee, Germany, 2005.

[7]   J. Tidwell, *Designing Interfaces*: O'Reilly, 2005.

[8]   J. M. Kleinberg, Authoritative sources in a hyperlinked environment. *Journal of the ACM*, vol. 46, pp. 604-632, 1999.

[9]   J. M. Carroll and M. B. Rosson, Paradox of the active user. In: *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, J. M. Carroll, Ed., 1987, pp. 80-111.

[10]   E. Golden, B. E. John, and L. Bass, The Value of a Usability-Supporting Architectural Pattern in Software Architecture Design: A Controlled Experiment. Presented at Proceedings of the 27th international conference on Software engineering, St. Louis, MO, USA, 2005.

[11]   N. Juristo, M. Lopez, A. M. Moreno, and M. I. Sanchez, Improving software usability through architectural patterns. Presented at ICSE '03 International Conference on Software Engineering Proceedings: Bridging the Gaps Between Software Engineering and Human-Computer Interaction, Portland, Oregon, USA, 2003.

Acknowledgements