Active Support For Query Formulation in Virtual Digital Libraries: A case study with DAFFODIL*

André Schaefer, Matthias Jordan, Claus-Peter Klas, and Norbert Fuhr

University of Duisburg-Essen
a.schaefer,mjordan,klas,fuhr@uni-duisburg.de

Abstract DAFFODIL is a front-end to federated, heterogeneous digital libraries targeting at strategic support of users during the information seeking process. This is done by offering a variety of functions for searching, exploring and managing digital library objects. However, the distributed search increases response time and the conceptual model of the underlying search processes is inherently weaker. This makes query formulation harder and the resulting waiting times can be frustrating. In this paper, we investigate the concept of proactive support during the user's query formulation. For improving user efficiency and satisfaction, we implemented annotations, proactive support and error markers on the query form itself. These functions decrease the probability for syntactical or semantical errors in queries. Furthermore, the user is able to make better tactical decisions and feels more confident that the system handles the query properly. Evaluations with 30 subjects showed that user satisfaction is improved, whereas no conclusive results were received for efficiency.

1 Introduction

It is a well known fact, that query formulation for digital libraries is a difficult task. While web search is mostly based on ranked retrieval, most digital library interfaces and OPACs (Online Public Access Catalogues) offer field-based boolean search interfaces. The average user has to learn the conceptual implications of this search modality and has to understand the underlying vocabulary problems. In *virtual* digital libraries, which search in a distributed way over several underlying digital libraries, this conceptual burden tends to increase.

The digital library system DAFFODIL[1] is such a virtual digital library, targeted at strategic support of users during the information search process. For searching, exploring, and managing digital library objects it provides user-customisable information seeking patterns over a federation of heterogeneous digital libraries. Searching with DAFFODIL makes a broad range of information sources

* The project DAFFODIL was funded by the German Science Foundation (DFG) as part of the research program "Distributed Processing and Delivery of Digital Documents"

[1] `http://www.daffodil.de`

easily accessible and enables quick access to a rich information space. However, the response times are well above that of single digital libraries, since DAFFODIL acts as meta-search engine for the 14 digital libraries currently included, so the result can only be shown and ranked after the slowest digital library answered the user's query. In such a situation, the problem is aggravated if the query is syntactically or semantically incorrect, therefore retrieving no results and wasting expensive time.

DAFFODIL already provides a broad range of tools to help the user in query formulation, e. g. a thesaurus, a spell-checker, a classification browser, and a related-term service. However, all these tools are currently separated, and are not integrated in the context of the query formulation (cf. [1]). The previous system version offered automatic suggestions of search terms that where shown in a separate window, a feature generally well-liked by users. However, our own findings in evaluations of DAFFODIL [2] indicated, that actually using terms provided in such a manner constitutes a high cognitive burden for many searchers as this feature was underutilized. Thus, in order to improve the usability of DAFFODIL, we decided to bring the suggestion more to the focal point of the interaction.

As a solution to reduce the workload and cognitive work of users, mixed initiative and interface agent-based approaches have been proposed in [3,4,5] for other application areas. In this paper, we want to investigate the question if these approaches can be applied to query formulation for searching in digital libraries. It is well known that in information seeking tasks, the user feels uncertain, has an ill-defined goal and a very limited attention span. Thus, the mental costs of pushed suggestions may be counterproductive in this situation.

These considerations led us to a user-oriented iterative redesign of the query form in the user interface of DAFFODIL [6]. The specific goal was to provide a mixed-initiative query formulation interface, which should help the user

1. to make fewer low-level errors,
2. decrease the cognitive load in early stages of the query reformulation cycles,
3. increase confidence in the constructed query,

and therefore increase user efficiency and effectiveness.

In the remainder of the paper we will first describe the problem in more detail and provide a brief overview on related work, followed by a description of our concept for proactive user support. Then we present the iterative design and evaluation process for integrating this feature into DAFFODIL in a user-friendly way. In addition, we evaluate the extension of the mixed-initiative approach for suggesting tactical or strategic moves and actions. A summary and an outlook conclude this paper.

2 Problem Description and Related Work

When users attempt to instantiate *known items* by specifying known facts about the documents, e.g. author and time range, the field-based boolean modality suits

the users' needs quite well. But in user tests [7,2] it is observed that users still need an average of four to five queries to find the information they were looking for. One major cause of the need for repeated querying is faulty queries. Most of these "errors" fall into the category of misspellings or typographical errors. Sometimes problems with query syntax lead to unexpected results.

The situation becomes even more complicated in the case of *topical search*. If the topic is not well known and the user needs to explore the information space, she is immediately hit by the vocabulary problem, e.g. *which search terms give the desired results, which related terms or synonyms are needed here.* This adds to the uncertainty the user already experiences due to his *anomalous state of knowledge* [8] that led to the topical search in the first place.

The psychology behind these problems [9] is well known and existing models of information seeking behaviour [10,11,12,13] explain the situation quite well. The users start with a high degree of uncertainty and learn about their original need during the iterations of their query reformulations and result inspections. Low level errors, like spelling mistakes or inadequate boolean logics, can be explained as an expression of uncertainty or fear when starting a search.

Bates [10] proposed a medium level of system support where the user is supported doing her task, but still remains in control. She differs between four levels of search activities: MOVE, TACTIC, STRATAGEM and STRATEGY. While moves can be any activity associated with searching, tactical activities try to drive the search in a specific direction, e.g. BROADER or NARROW. A medium level of system involvement means that the system should suggest corrections and tactical term additions or substitutions. Additionally activities outside the search form could be suggested, e.g. doing a journal run or computing a coauthor network for a more efficient strategy.

Together with Bates' concepts of system support, where users enter search terms, it makes much sense to present corrections, mark errors and suggest search terms or other possible moves right from the beginning of a search cycle[14,15]. Other systems suggest mostly terms for query *reformulation*. In our approach, we want to increase the system support significantly, by helping the user already in the important step of initial query formulation, thus decreasing the user's high uncertainty at the beginning of the search. Since the *Anomalous State of knowledge* (ASK) (cf. [8]) problem has been recognised, it is clear that term suggestions are a useful addition to information systems' search interfaces. Schatz et al. [14] have demonstrated and analysed the usefulness of term suggestions based on a subject thesaurus and term co-occurrence list. Brajnik et al.[15] have conducted a case study to investigate the value of query reformulation suggestions, terminological and strategic help, and the best way to provide them. *User-controlled interaction* appears to be preferred by most users, and they find support for Bates' hypothesis that users do not want fully automated search systems to which they can delegate the whole search process (cf.[11]).

In a similar effort to support users in formulating their queries, Google[2] provides a selection of interesting methods, one of which is "Did you mean".

[2] http://www.google.com

This facility suggests reformulations of queries that Google regards as a possible misspelling. The experimental *Google Suggest*[3] interface provides users with an online prefix completion facility. According to Google's online documentation, this facility suggests term completion based on overall popularity of search strings, not on the user's preferences or search history. The main difference to "Did you mean" is that Google Suggest supports the user during the initial query formulation. Altavista[4] also provides this kind of help with the variation that Altavista conducts searches both for the misspelled keyword and its automatic correction. The Scirus search engine for scientific information[5] provides term suggestions, based on a term co-occurrence analysis in form of clickable links that modify the original query.

We argue for incorporation of as many help features as possible into the query formulation part of the interface, as this is the most difficult cognitive work for the user. It is also most important for successful reformulation and the effectiveness of the search process. Existing systems still leave much room for improvements. While approaches exist, they still lack integrated efforts with more than one method. Thus, a modular system of support components seems to be a reasonable approach.

3 Proactive Support

As described above, users require terminological and strategic help while searching. As they are often unaware of what they can do and what is possible within a system, they fail to actively use tools like thesauri, subject classifications, or dictionaries. The implication is that the system should observe the user's search tactics and offer help and suggestions at the right time and the right spot.

Thus, trying to support users *proactively*, means presenting suggestions or spelling corrections *without their explicit request*. In the given scenario of query formulation this means that some agent observes the query construction process and takes appropriate action if necessary or useful. This should happen in a way that matches the users' expectations and does not leave them puzzled or distracted by the offered suggestions. These methods have been successfully applied in other domains like office software or integrated development environments for quite some time and have proven to be very popular. Therefore we argue that these methods will also provide benefit for search interfaces and should always be considered when designing them. For this purpose all tools in Daffodil for entering and submitting queries are extended by an observation service. The proactive service should achieve several goals to improve efficiency and effectiveness:

- Mark potential errors on the typographical and syntactical level
- Suggest useful search terms, marking the partially constructed query

[3] http://www.google.com/webhp?complete=1&hl=en
[4] http://www.altavista.com
[5] http://www.scirus.com

– Give useful hints on semantic query improvement
– Suggest useful actions and moves to add valuable facts to the query, e.g. looking up co-authors for the already known authors

These suggestions and markings have to appear

– at the right time, e.g. while the user is hesitating or thinking about possible extensions of the query
– in an unobtrusive manner – the user should be able to ignore the suggestions and to concentrate on the task
– in context – suggestions too far from the current focus of attention often will not be considered
– with clear semantics – the user needs to comprehend the goal of the suggestion and the reason the system is presenting the suggestion

3.1 Design of the Interface

Based on the goals described above, the subtasks of the design are to decide

– how to mark errors and words that the system can provide help on, and
– how to present suggestions (e.g. terms and actions)

The proactive functions should show the user *markers* and *suggestions* at the point of her work, which can be any query form.

– The user's focus of attention is at that place[6] and she should not be distracted with out-of-focus annotations and suggestions.
– The suggestions should appear in form of popup lists near the current cursor or mouse position, because they can be placed near the focus of the user without too much interruption and are a familiar concept. Icons, markup, and colours will be used to visualise for which items the proactive functions can provide suggestions.
– To provide semantical explanation some additional text may be useful, but it should not exceed the absolutely necessary level of verbosity.

The cognitive burden of the user will increase significantly if she is forced to think about the suggestions, e.g. by a separate dialog which *gets in the way*. It is desirable, that the cognitive model of what is displayed and why, is easy to learn and maintain, and also easy to ignore, if the user is sure about her input.

In the design of the *markers* we tried to use well-known symbolisms. Red curly lines (see figure 2) are used to mark erroneous words in a text. We expanded this scheme to mark words where the system can provide help, even if the word itself is not wrong. We used orange as a code for "warning", marking words that might be, but not necessarily are wrong. We choose blue as a code for "information", to mark words the system has suggestions for. We were aware

[6] most users do in fact look at their keyboard instead of the form on the screen, but that is another matter

that colour codes might be misleading, because some colours like blue don't have a widely accepted meaning in the context of computer messages.

Another possible kind of "error" in our setting is located on the field level. These are mostly semantic errors, that we decided to mark this kind of errors by red square brackets around the affected text field. An example for this would be a query for documents of a specific author in a range of years where the author did not publish. While there is no error within any field that could be marked, the combination of the two field contents would lead to an empty result set. A similar error that is marked in this way is the conjunction of two year terms (see below). We chose this marker to avoid the visual overload resulting from using red boxes around a text field and to help colour blind people notice the markers by their distinctive shape.

The design of the *popup list* had to provide a means to present different kinds of information to the user – preferably in the same list at the same time. An additional goal was to structure the information to reduce the mass of visual stimuli the user has to process at once when the list pops up. Some ideas for this have been suggested in [16]. To implement such a hierarchical popup list we first wanted to develop and test a design for a flat list to expand it later after improving the shortcomings of the flat design.

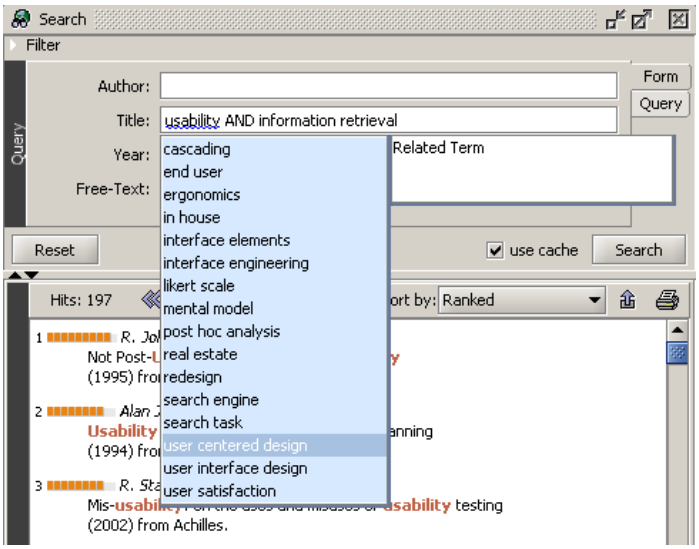

Figure 1. The popup list presenting related terms and an info message.

Figure 1 shows the popup list presenting a selection of co-occurring terms related to the term `usability,` which is marked by a blue curly line. The user selected the term `user centred design` by keyboard navigation, which resulted

in the informational message on the right side of the popup list. Accepting the selection would add the selected term to the query, connected to `usability` by an OR operator. The function would also put parentheses around the two terms, to be more specific on the query.

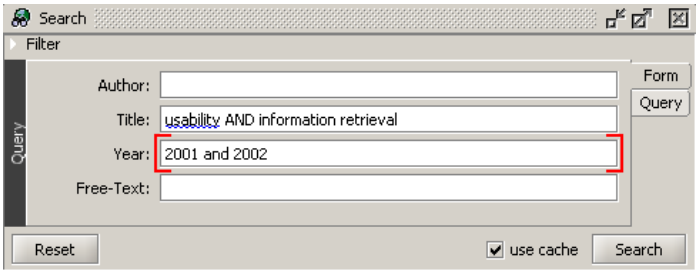

Figure 2. DAFFODIL marked the year field with red square brackets to notify the user about the over-constrained year clause

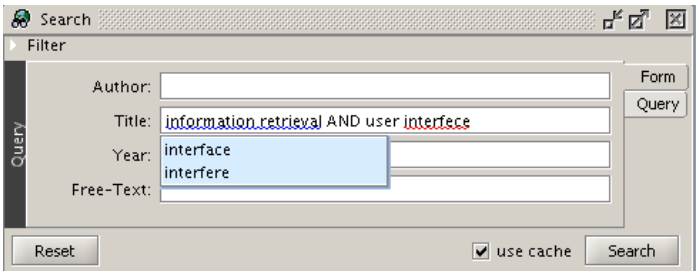

Figure 3. DAFFODIL suggests spelling corrections.

Figure 3 shows the first term marked in blue to notify the user about available suggestions and the second term in red because it is a possible error. The popup list suggests corrections. In this situation the user can only tell which term the suggestions are for based on the cursor position.

3.2 Design of the Functions

The agent observing a query form has several modules with different subgoals, providing an easily extensible set of functionality. The observer takes action if the user hesitates for a short moment, but the action can also be triggered actively with a key shortcut. When the cursor is on or directly behind a word

for which suggestions are available, a popup list with the information (currently only terms) is shown. When pointing or navigating to a term, by mouse or keyboard, a short explanation of the reasons behind the suggestion is shown to the right. The list vanishes if the user continues typing, clicks somewhere else, or selects a term. Some aiding functions which are more expensive in terms of computation time (compared to user interaction) will take action when the user submits a query – to offer their suggestions at query reformulation time. The form input is checked for typographical errors. If a misspelling is detected, terms are highlighted immediately with a red curly line. The query history module provides a convenient method to search the query history for old queries. Each list item presents the query and three example-hits that were found using the query for episodic memory support as in in [17]. The history of the user is also scanned to suggest terms which complete the currently entered prefix to previously entered terms. The related-terms module fetches terms, based on term co-occurrence, for the given query from the associated DAFFODIL service. Since it is quite expensive and fetching the results typically takes longer than half a second, the results are only being offered for query reformulation, after submission of the initial query. The thesaurus module delivers synonyms for given terms. It also offers broader or narrower terms if the number of results indicates that the query was too broad, respectively too narrow. Finally, the author-names module suggests correctly spelled author names, from a list of known authors.

4 Evaluation

The design of the proactive support methods was accompanied by three different qualitative user tests: heuristic evaluation of slide show mockups, controlled tests with two groups of students with and without the proactive functions, and single user loud thinking tests with video recording. Our expectations were that:

- the users accept the proactive functions
- the delay timing, until the proactive functions are triggered, is a good choice
- the users like the enhanced DAFFODIL more than the baseline version
- the users use the proactive functions often and even invoke them explicitly, once they found out how to do so
- the users have a clear understanding of the proactive functions

4.1 Methodology and Results

Heuristic evaluation with mockups For the heuristic evaluation (HE) we displayed mockups of our proposed ideas in a mockup slide show. We wanted to see if users understood the displayed suggestions. We asked if they would accept this kind of suggestion while specifying some query. The users commented on the mockups, and on problems and ambiguities they found. They were asked to assign their problems to a heuristic problem category, as supposed in [18].

This heuristic evaluation showed that the design of the proactive functions was generally accepted; however, some aspects had to be re-worked, one of which was the form and colour of the error markers. Information which was displayed out of focus, e.g., hit count predictions in the status bar at the screen bottom, was requested to be made more visible. The markers inside of the form fields were generally accepted.

Controlled user test Based on the revised design following from the heuristic evaluation, we implemented a first version of proactive support. This system version was tested with a group of 20 undergraduate students of computer science. They had to perform a set of five tasks, similar to those described in [2]. The tasks were a mixture of known item instantiation and topical searches. In addition to system logging of the users' interactions, we asked the users to fill a questionnaire after finishing the tasks.

The baseline evaluation, where half of the users performed the same tasks without any proactive support by the system, showed that approximately 45% of the submitted queries were erroneous. There were several classes of problems which occurred frequently:

1. Spelling mistakes in author names and query terms
2. Inadequate use of boolean operators or phrase syntax
3. Inadequate use of language (e.g. German instead of English)
4. Over-constrained queries – queries which cannot have answers, because of contradictory expressions in conjunctions
5. Under-constrained queries – queries using highly frequent search terms or only time ranges without any further expression

In the second part of this phase, the other half of the students had to perform the same tasks with proactive support enabled. In comparison to the baseline group, these test subjects worked in a more confident way, asked less questions, and had fewer negative comments. They were not more successful, however, and still lacked strategic skills to perform the topical task efficiently. These results are consistent with the findings of Brajnik et al., who observed that the searchers are often unaware of their need for strategic help, while they are actively requesting help at the terminological or syntactical level. Search times and success rate did not vary significantly between the two groups, but the number of participants was not sufficient for good quantitative results anyway.

Single user loud thinking tests The findings of the first two test runs formed the basis of a third, refined implementation which was tested with ten single users in controlled loud thinking tests with video recording for in depth analysis of the process and the remaining problems. These users performed the same tasks as the twenty students in phase two. This group was mainly formed by nine research assistants. Furthermore a librarian, who normally teaches students to search in bibliographic databases.

Since the users were thinking loudly, a more in depth inspection about their cognitive models and expectations was possible then in the HE test at the beginning of our evaluation process. In addition we inspected how the timing of the suggestions worked. Timing is an important attribute of the proactive suggestions, as it turned out that the expected behaviour varies notably between users. Some users still requested lists which stay longer on the screen and are more eye catching, as they didn't notice them right away — mostly because they had to concentrate on the keyboard instead of the screen. One user had a general disapproval of proactive functions, also in word processors or similar tools, and requested the ability to stop the observer (which is already possible, through the personalisation of DAFFODIL).

Overall Results

General acceptance Overall the acceptance of the proactive functions varied. While most users started to generalise (requesting and expecting further proactive help in more situations), some refused to use the proactive functions. The conclusion is that each proactive function needs to be personalisable, so the user can turn the feature on or off.

Users used them multiple times during each task, depending on their level of confidence. In general, with growing confidence test subjects used the proactive functions less. Many users noted that they liked not having to type each word themselves. Users also stated that knowing that there are no spelling errors in their query makes them feel more confident, and that it saves them time because they don't have to reformulate a misspelled query.

Visual markers The similarity between the marks for misspelled words and words for which term suggestions are available (red underline vs. blue underline) was not helpful for most users. While the design goal was to let users implicitly use the same mechanisms to open the popup list and choose the right substitution, it did not work smoothly. As the semantics are obviously clearly different, the graphical representation has to be as well.

Timing and Usage The timing seemed to be a good compromise between a high level of disturbance and a long waiting time. No user complained about popup lists showing up too frequently, and no user stated that the time to provoke the list is too long. In fact, one user triggered the popup list during a short typing break but later answered that he did not see it. Some users wanted the lists to be on the screen longer. Currently, it closes immediately when the user continues typing and this was often done before realising that the suggestions might have been helpful in the current context.

Most users who actively provoked the suggestion lists, did so by simply deleting and re-typing a part of a word and waiting until the list opened. An easy way to open the suggestions with the mouse is needed, e.g. mouse-over or click events. None of the users tried to employ the context menu (right mouse button

click) for this purpose, although this feature offers additional functions in several other DAFFODIL tools.

Understanding Users generally require an explanation of the reasons behind a term suggestion, especially if the terms suggested are not known to the searchers (e.g., the system suggested the term "simulator sickness" after users searched for "cybersickness"). Some users were reluctant to use suggestions which they could not understand. To improve this, more semantics and reasoning should be offered in the comments that are given for each suggestion; however, since most of the related terms suggested by the system are based on statistical correlation, this may not be very helpful for the user.

Overall An assessment of the overall satisfaction after using the DAFFODIL system was requested as part of the questionnaire after the session. On average, users gave 7-8 out of ten points for retrieval quality and 6-7 points for usability. Considering this result and the comments of the users given during our loud thinking tests we are confident that proactive functions significantly improve the satisfaction of users searching for publications in DAFFODIL.

5 Summary and Outlook

DAFFODIL provides strategic support for searching in digital libraries. In this paper, we have focused on the usability of the most crucial functionality, namely query formulation in the search tool. In an iterative design and evaluation process, we have integrated proactive functions for supporting query formulation at the focus of the user's attention. The experimental results show that user satisfaction is increased and uncertainty is reduced.

The evaluation also showed a general acceptance and the request for even more proactive support, like faster query refinement and quick indication of empty result sets; we have started working on the development of appropriate methods for implementing these features.

A deeper personalisation, e.g. in form of individual user term-spaces instead of global ones, will aid the user even more. These term-spaces can be created from the user-specific data-flow in the DAFFODIL system; this will also avoid too large lists of synonyms or author name completions.

The presented evaluation of the proactive functions used qualitative methods, because of limited resources and the lack of a standardised document base. The next step is a quantitative evaluation, in order to show that proactive functions not only improve the user-friendliness of a digital library systems by reducing the typing effort during query formulation, but also increase the average retrieval performance in general and especially for novice users.

References

1. Kriewel, S., Klas, C.P., Schaefer, A., Fuhr, N.: Daffodil - strategic support for user-oriented access to heterogeneous digital libraries. D-Lib Magazine **10** (2004)

2. Klas, C.P., Fuhr, N., Schaefer, A.: Evaluating strategic support for information access in the DAFFODIL system. In Heery, R., Lyon, L., eds.: Research and Advanced Technology for Digital Libraries. Proc. European Conference on Digital Libraries (ECDL 2004). Lecture Notes in Computer Science, Heidelberg et al., Springer (2004)
3. Maes, P., Wexelblat, A.: Interface agents. In: CHI 96. (1996) 2
4. Maes, P.: Agents that reduce work and information overload. Communications of the ACM **37** (1994) 30–40
5. Horvitz, E.: Principles of mixed-initiative user interfaces. In: CHI'99, Pittsburgh, PA, USA, ACM (1999) 159–166
6. Fuhr, N., Klas, C.P., Schaefer, A., Mutschke, P.: Daffodil: An integrated desktop for supporting high-level search activities in federated digital libraries. In: Research and Advanced Technology for Digital Libraries. 6th European Conference, ECDL 2002, Heidelberg et al., Springer (2002) 597–612
7. Tombros, A., Larsen, B., Malik, S.: The interactive track at inex 2004. In Fuhr, N., Lalmas, M., Malik, S., Szlavik, Z., eds.: INitiative for the Evaluation of XML Retrieval (INEX). Proceedings of the Third INEX Workshop. Dagstuhl, Germany, December 6–8, 2004. (2005)
8. Belkin, N.J.: Anomalous states of knowledge as a basis for information retrieval. Canadian Journal of Information Science **5** (1980) 133–143
9. Borgman, C.L.: The user's mental model of an information retrieval system. In: Proceedings of the 8th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, ACM (1985) 268–273
10. Bates, M.J.: The design of browsing and berrypicking techniques for the online search interface. Online Review **13** (1989) 407–424
11. Bates, M.J.: Where should the person stop and the information search interface start? Information Processing and Management **26** (1990) 575–591
12. Marchionini, G.: Information seeking in electronic environments. (1995)
13. Kuhltau, C.C.: Developing a model of the library search process: Cognitive and affective aspects. RQ **28** (1988) 232–242
14. Schatz, B.R., Cochrane, P.A., Chen, H., Johnson, E.H.: Interactive term suggestion for users of digital libraries: Using subject thesauri and co-occurrence lists for information retrieval. In: DL'96: Proceedings of the 1st ACM International Conference on Digital Libraries. (1996) 126–133
15. Brajnik, G., Mizzaro, S., Tasso, C.: Evaluating user interfaces to information retrieval systems: A case study on user support. In Frei, H.P., Harman, D., Schäuble, P., Wilkinson, R., eds.: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, ACM (1996) 128–136
16. Sanderson, M., Coverson, C., Hancock-Beaulieu, M., Joho, H.: Hierarchical presentation of expansion terms. (2002)
17. Ringel, M., Cutrell, E., Dumais, S., Horvitz, E.: Milestones in time: The value of landmarks in retrieving information from personal stores. (2003) 184–191
18. Nielsen, J.: Heuristic evaluation. In: Usability Inspection Methods. John Wiley and Sons, New York (1994)