A Meta Search Engine for User Adaptive Information Retrieval Interfaces for Desktop and Mobile Devices

Ernesto William De Luca and Andreas Nürnberger
[1] Otto-von-Guericke University of Magdeburg
School of Computer Science
Universitätsplatz 2, 39106 Magdeburg, Germany
{deluca,nuernb}@iws.cs.uni-magdeburg.de

Abstract. In this paper we present an information retrieval system (meta search engine) that provides web service based methods for user and query specific annotation of search results. The system currently supports two types of annotations: Categorization based on information extracted from user bookmarks (Intelligent Bookmarks) and semantic disambiguation of query terms in documents using an ontology that can be selected by the user (Sense Folders). We describe both approaches and present adaptive user interfaces for different devices (mobile and desktop) that use these services in order to improve the user's search process. Furthermore, we discuss the advantages of dividing query results set processing (the information to be presented) from the interface design (information presentation) using web services in order to simplify the development of retrieval systems for, e.g., different desktop as well as mobile devices.

1 Introduction

Over the last years a number of methods have been proposed and realized, that enable a user to search in large collections of documents which may consist, e.g., of web pages or text documents. Most of these methods are based on keyword queries which means that a user has to provide a list of keywords that describe the contents of the searched document. After performing the query the user obtains a list of documents, which is ordered by the degree of similarity to the applied query (see, e.g. search engines like AltaVista [10] and Google [4]). If the keywords are well chosen, these methods frequently provide an appropriate list of results due to their sophisticated ranking methods, which are usually not based on pure Boolean queries but take into account word frequencies, thesauri or even simple semantics [1]. However, if the result list covers, e.g., diverse meaning categories (if the search terms are ambiguous) or diverse topic categories (if the search terms are used in different domains), then these categories appear rather unsorted in the result list. Since this is the case for most queries, automatic categorization of the documents would strongly improve the retrieval performance for a user, since he can then select the intended category and thus reduce the result list to a subset of relevant documents.

First approaches based on this idea had been provided, e.g., by the web search engine Yahoo [24]. Here, the complete document collection was (manually) hierarchi-

cally organized and this structure was provided to the user to browse the document collection. The main drawbacks of this approach are the necessity of manual interaction for the arrangement of the documents. As the classification has to be updated regularly, maintenance can get rather costly. More recent approaches which try to categorize documents automatically are realized, e.g., by meta search engines like Vivisimo [22] or KartOO [14]. Vivísimo's search and clustering solutions are based on an approach that organizes search results into categories. There is no preexisting taxonomy. The document clustering and meta-search software automatically categorize search results on-the-fly into hierarchical clusters. The categories that are automatically selected from the words and phrases contained in the results or documents themselves. The Vivísimo algorithm is based on textual similarity. Kartoo is a search engine which shows results in visual interface. The results of several search engines are combined and represented in a series of interactive two-dimensional maps through a proprietary algorithm. The thematic relations between the results are indicated with annotated lines. Furthermore, colors are used to symbolize certain subject. Thus, this search engine provides information in context in order to allow the user to navigate in semantic graphs.

However, all currently available categorization techniques still have difficulties in providing appropriate categories. Both, the manually assigned categories (subjectively labeled from humans) or the automatically derived categories (usually obtained by clustering methods [20, 3]) only consider the word distribution in documents without taking into account criteria derived from the underlying query, such as different meanings of a term or information from a user profile. Thus, the query as well as user specific interests are neglected during categorization. Therefore the assigned categories usually do not represent the categories a user is expecting for the query at hand.

In the following, we present the framework for an information retrieval system that enables the use of user profiles in order to automatically annotate result set based on user and query specific information. Furthermore, the system enables the use of user specified ontologies for disambiguation purposes and supports the use of adaptive interfaces on different devices (Sect. 2). Before we present in Sect. 4 two adaptive interfaces, we discuss in Sect. 3 the role of content and user adaptivity in information retrieval in order to motivate the specific interface design. In Sect. 4 we finally present two application examples and clarify the importance of the differentiation between content and presentation.

2 The Information Retrieval Framework

The main objective of user modeling in the area of information retrieval is to extract and store information about a user in order to improve the retrieval performance. A user model in this context usually consists of a list of keywords to which relevance degrees are assigned. More complex models [25] distinguish between different search contexts, or store additionally relations between keywords in order to model a more expressive search context. A user profile, based on the user interests, can be obtained, by extracting keywords from the queries performed or documents read by the user in

the past [22]. Furthermore explicit as well as implicit feedback information from a user can be collected in order to learn or refine a user profile [17, 18, 2].

In the system presented in the following, we use user and query specific information in order to annotate – and thus categorize – search results from other search engines or text archives connected to the meta search engine by web services (see Fig. 1). The system currently supports two types of annotations: Categorization based on information from user bookmarks (*Intelligent Bookmarks*) and semantic disambiguation of query terms using an ontology that can be selected by the user (*Sense Folders*).

The idea of *Intelligent Bookmark* annotation is to exploit information about the way a user is ordering, sorting or categorizing his documents in order to categorize so far unseen documents. This approach is described in more detail in Sect. 2.2.

The idea of the *Sense Folder* annotation approach is to use ontologies in order to disambiguate query terms used in the retrieved documents [8]. Thus it is possible to categorize documents with respect to the meaning of a search term. Further details can be found in Sect. 2.3.

In the following we briefly describe the query processing and annotation process of the system and its embedding in the meta searcher.

2.1 Annotating Result Sets

In Fig. 1 an overview of the system architecture is given. The search engines (e.g. Google or local searchers) as well as the user interface are connected to the system by Web Services. Thus the system can easily be extended by additional search engines or used by different interfaces. The annotation methods are implemented as modules within the meta search engine.

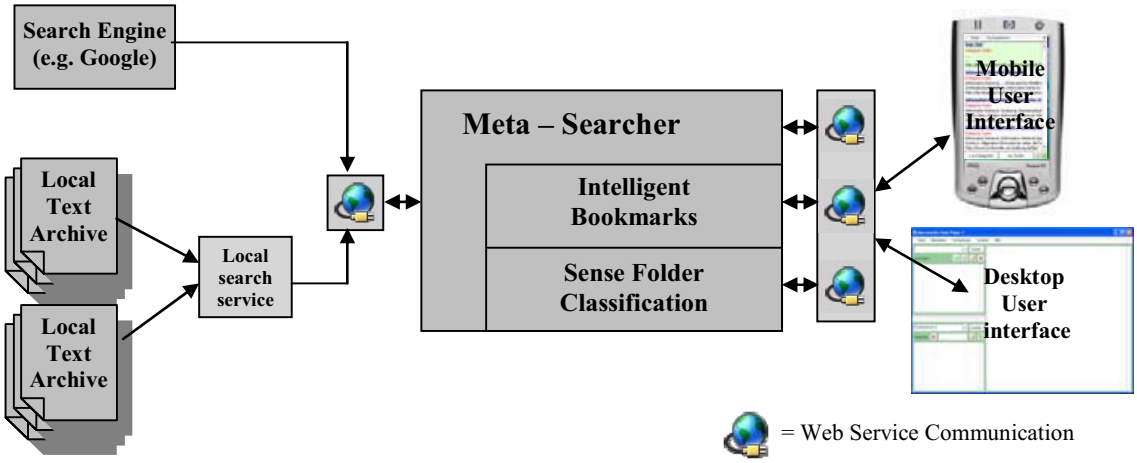

Fig. 1 Overview of the retrieval system

The main idea of the system is to provide additional disambiguating information to the documents of a result set retrieved from a search engine in order to enable the clients to annotate, restructure or filter the retrieved document result set. This idea is motivated by user studies have shown that category interfaces are more effective than list interfaces in presenting and browsing information. For example, in the studies

presented in [11] the authors have evaluated the effectiveness of different interfaces for organizing search results. It turned out that users favored the category interface over the list interface and were 50% faster in finding information organized into categories. In this study the authors provided additional semantic category contexts for the list interface using categories as described in [15] in order to also evaluate what interface elements were the most important for the search process.

The annotation process we have implemented into the meta search engine can be briefly described as follows:

1. First the user types his query (keywords).
2. Search results are indexed.
3. Search results are classified/annotated by its *Sense Folder*.
4. Search results are classified/annotated using the *Intelligent Bookmarks.*

Based on this process an annotated result set is obtained. This result set is forwarded to the client via the web services.

2.2 The Intelligent Bookmark Approach

Most users use the bookmark functionality of their web browser to store relevant websites in a more or less structured way. The main idea of the *Intelligent Bookmarks* is to provide users additional benefit from storing bookmarks. Therefore we extended the functionality in different ways: The system supports hierarchical bookmarks in a tree structure, the bookmark hierarchy can be accessed and managed from any device and furthermore the structural information stored by the bookmark hierarchy and the assigned web pages is used to annotate search results. All functions can be accessed via web services. Thus, the bookmarks can be visualized by clients in different ways, e.g., as a folder structure like in the interfaces discussed in Sect. 4.

Based on the web pages stored in the bookmark structure, a classifier is trained that uses the folder names as category labels. Thus, the more results are stored and assigned to a category, the better does the system learn something about the way a user is structuring information. We assume that every category folders describe groups of web sites that implicitly define the categories for the support system of the search interface.

This classification approach can be used in order to annotate search results or to filter documents. Since the system just provides the annotation, the visualization for the user can be realized by the client systems. A more detailed description of the functionality of this process is given in [13].

2.3 The Sense Folder Approach

Similar to the *Intelligent Bookmark* annotation, the classification terms obtained by the disambiguating classes of the *Sense Folders* are then added to each document listed in the result set of a query and are forwarded to the search client for further use. In the *Sense Folder* approach integrated in the retrieval system we consider the different linguistic relations that describe the context of the searched word in the ontol-

ogy in order to recognize the meaning of the user query [8]. Currently we use these linguistic relations from the WordNet ontology [23] in order to create prototype vectors that are defining the *Sense Folders* for the different meanings of the query terms. Obviously, this approach is restricted to query terms that appear in the ontology.

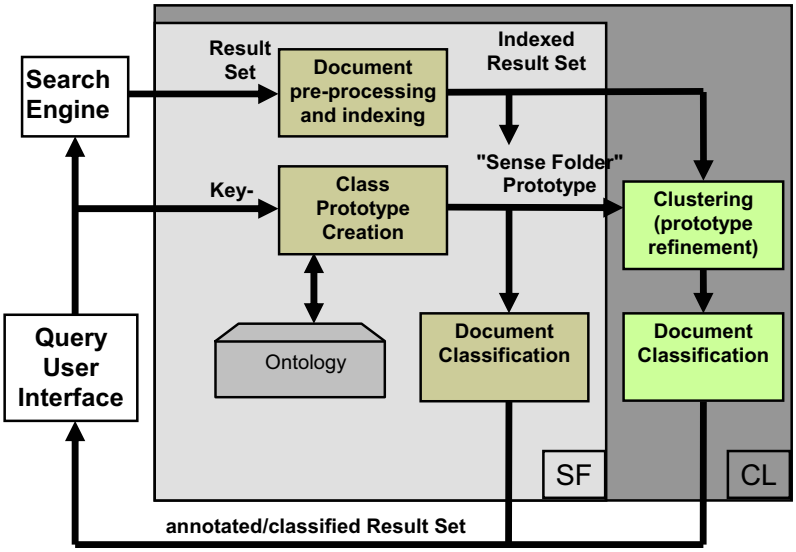

Fig. 2 Overview of the Sense Folder (SF) and Clustering (CL) classification process

For our disambiguation problem, we first assign semantic information (retrieved from the ontology) to the result sets we get from the search engine. Afterwards, we use a clustering algorithm in order to fine tune the initial prototype vectors of each *Sense Folder* using the distribution of documents around the initial prototype vectors (see Fig. 2), i.e., we expect that in a web search usually a subset of documents for each possible meaning of a search term is retrieved. Thus, each subset forms a cluster in document space describing one semantic meaning of this term. In other words, every document is first assigned to its nearest prototype vector derived from the ontology and afterwards this classification is revised by the clustering process [9]. This approach has shown to strongly improve the classification (or disambiguation) performance [9]. The semantic information assigned is appended to the document as additional information in order to help the user in finding the relevant documents, without loosing too much time in browsing all documents.

3 User and Content Adaptivity for Information Presentation

Standard keyword based search engines retrieve documents without considering the importance of user oriented information presentation. It means that the user has to analyze every document and decide himself which are the documents that are relevant. In order to support the user the retrieval system assigns additional information – currently retrieved from the ontology or the bookmark structure as described above – to the retrieved documents. Furthermore, this information should also be adapted according to the individual user needs.

The need to obtain a good adaptivity [5, 6, 7] to user needs should be based, on an appropriate user model in the retrieval system, since users expect individual information depending on their interests and knowledge. In order to achieve this, we need different user profiles [19, 21] that cover (almost) all user needs, including the relations between the user search words (content adaptivity), and the different user characteristics (depending on the user interests, knowledge, experience, language, culture, etc.). In this sense the *Intelligent Bookmarks* as described in Sect. 2.2 are only a first step. However, since the system exploits information the user is anyway willing to provide, this approach is non-obtrusive and thus more easily accepted then explicit feedback techniques. By storing bookmarks the user implicitly provides information about his interests and the way he likes to have information structured.

Besides the information describing a user we have to consider certain limits of the hardware when we implement a user interface for a specific, e.g., mobile device, and have to adapt our interface accordingly. We also have to observe, that information has to be independent from the interface and has to be presented individually; it means that users should have a unique support depending on their needs as independent as possible from the hardware they are currently using.

Implementing our retrieval system, we decided to divide query results set processing (the information to be presented) from the interface design (information presentation) using web services in order to simplify the development of retrieval systems for, e.g., different desktop as well as mobile devices. We have chosen to implement web services in order to give the possibility to access information in a way that is platform and hardware independent. Thus the communication between the search-client and the meta search engine as well as the meta search engine and search engines like Google or local search services is realized trough web services that represent a standardized interface. For the current implementation of the meta search engine we used Axis – an open source SOAP server and client – on a Tomcat Web Server. As an example, in Figure 3 a subset of the web services that are necessary for accessing the basic search functionality and the bookmark management is shown.

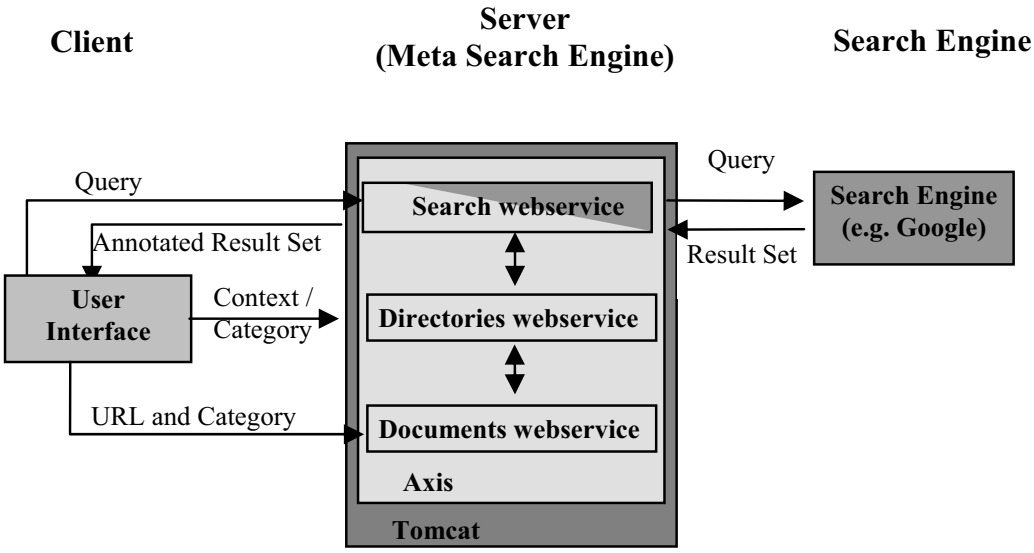

Fig. 3 Web Services interaction overview using the bookmark based classification

The first web service (*Search*) connects to a search engine (here Google using the Google API) and returns a set of results that contains additional annotations as described in Sect. 2. The *Directories* web service is used to manage the bookmark categories created from the user. Therefore it provides methods to add and remove folders for the bookmark hierarchy. The *Documents* web service is used to store and remove links (URLs) to web pages in the bookmark hierarchy.

4 User Interface Design

Since an information retrieval interface that is developed based on the meta search engine discussed above should help to accelerate the search process of a user, it is important to design the interface as user friendly as possible. Therefore in this section we discuss two prototypical search interfaces we implemented for the visualization of the available data. We also discuss briefly the different requirements that have to be considered implementing user interfaces for diverse devices.

4.1 Desktop User Interface

A screenshot of a prototypical search interface for desktop systems is shown in Figure 4. The interface basically consists of three parts: The query area, the contexts and categories area, and the result list area.

The query area is in the upper left side of the interface. Here, the user can enter the search terms that will be used from the system in order to obtain a list of results from a connected search engine. The obtained results are then stored by the interface in a specific folder that is labeled with the search terms given by the user. Every time a user starts another query, a new "query" folder will be created, thus the user can navigate through them without writing the same query twice.

The bookmark area is placed underneath the query area. The bookmarks are user specific. Here we used the first level in the bookmark hierarchy to distinguish between search context and categories within a given search context. Thus the user can store bookmark hierarchies for different search purposes, e.g. work and recreational activities. This supports on the one hand the user in structuring the bookmarks and on the other hand simplifies the classification of documents in search results, since we already have separated the documents in different domains and can learn individual classifiers for the respective subsets [13].

The result list area is situated on the right side. Here both, the web search results and the list of web sites contained in a selected category, are presented. For each item its title, its hyperlink, and a snippet briefly describing the content of the belonging web page as provided by standard search engines is given. In addition, categories derived by the *Intelligent Bookmark* and *Sense Folder* annotation methods (see Sect. 2) are displayed. For the *Sense Folder* annotation additional label information is retrieved from the ontology and displayed to give the user a more detailed description of the content of the annotated document entry. Thus documents belonging to differ-

ent semantic categories are labeled with different senses derived from the ontology (see also Section 2.3 and [8]).

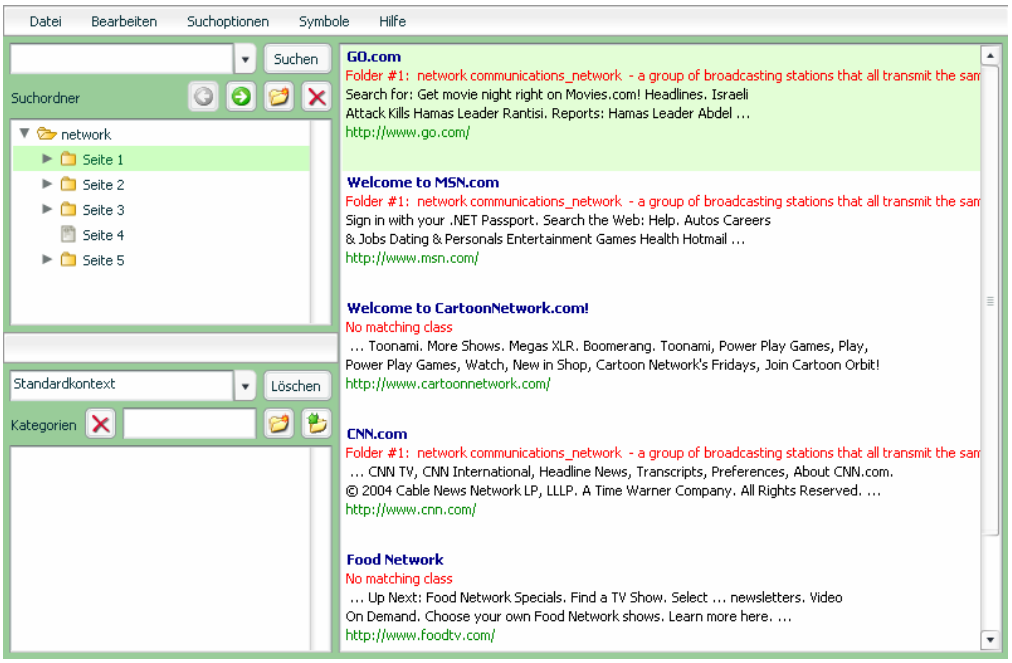

Fig. 4 Desktop User Interface. The information from the Sense Folder annotation is provided directly below each title line.

4.2 Mobile User Interface

The first consideration has to be done working on mobile devices is why such an effort is so different from the interface design for a more typical desktop based Web application. In the following, we mention some problems faced by user interface designers in wireless application development: Typically a wireless device has a more limited bandwidth than a wired device. Transmitting and receiving huge amounts of data is a problem. Therefore, the data containing the information to be presented should be well preprocessed and redundant, and unnecessary data should be removed.

Furthermore, the connection to a wireless device is intermittent and there is no persistent point-to-point connection. Mobile devices are compact in size and have the problem of the limited battery life. We have also the problem of limited memory – that, however, sometimes can be resolved expanding it with memory cards. Another fundamental difference between mobile devices and standard workstations is the user interface. A “normal” interaction (it means through mouse and/or keyboard) for such devices is usually not available. Furthermore, the screen area is almost always very small and thus data can be viewed, navigated and manipulated usually only in a very cumbersome way if the user interface has not been adapted for this environment [12, 16].

Using mobile devices, we had to consider the different possibilities of user interaction. We transferred the given functionality provided by the desktop interface, as good as possible to the PocketPC. The realized interface is shown in Figure 5a-c.

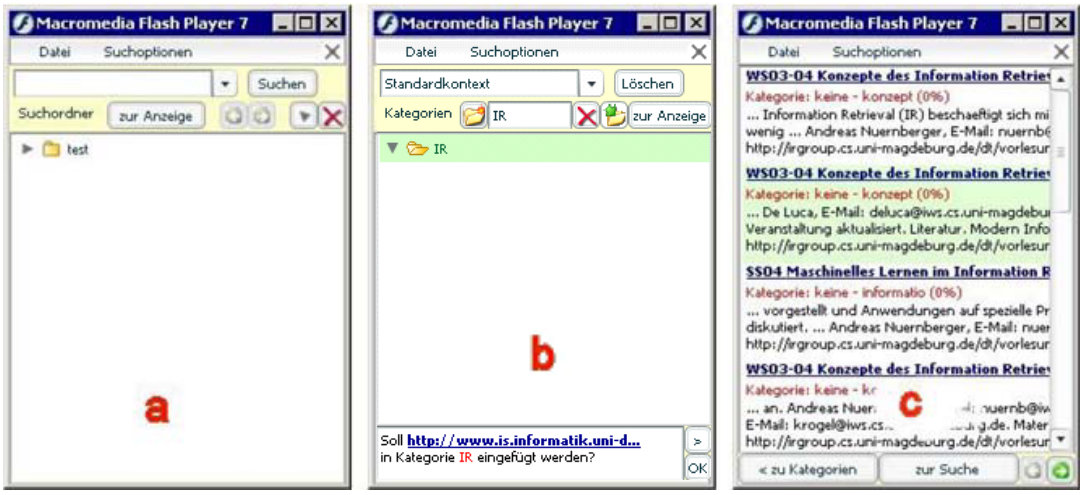

Fig. 5 Mobile User Interface: a) Search window, b) bookmark management and c) result list

In order to ensure an intuitive use, we provide three different adapted views of the user interface. In the desktop user interface, for example, we have a drag and drop functionality, in order to store or restructure the bookmarks. This is not possible in the mobile environment, because of the limited interaction functionality given from the use of the pen. For this reason we had to readapt the functionality of the user interface.

The navigation through the result list on a mobile device is done with a pen. If the user clicks on a document, the content will be then shown as a normal web page in the window of the standard web browser of the device.

Taking into account all software and hardware limitations, we developed an adapted mobile user interface. The basic components are three. We adapted their functionalities for mobile devices. These components (views) of the program are listed as it follows:

a) Search and presentation of the search results in a tree-form.
b) Results window: Search results are annotated/categorized using the different classification approaches (see Sect. 3) and are presented with the additional information as first information after the title.
c) The third component is used for the user specific (private) categories and contexts (bookmarks).

The user can type the query in the search window (Fig. 5a) using standard PDA text input methods. The system retrieves the documents presenting automatically the results, switching to the results window (Fig. 5c). Once a user gets results, he can choose to see one document by clicking on it with the pen. He can view the next results choosing the “next arrow” or can start a new search clicking on the search button.

Buttons (Fig. 5) provide the possibility to switch between the views. In every view a user can switch to another only clicking to the correspondent button. The user can choose to search new content, browsing the results, saving the interesting results to the *Intelligent Bookmarks* or viewing new results trough the simple use of a pen. All these interaction possibilities are given by the use of the buttons that give a quick

access to the other view. Storing data in categories and context is possible by selecting a document and pressing the “category button”. This implements the drag and drop functionality of the desktop interface (see also Figure 5b).

We implemented the user interface using the Macromedia Standalone-Flash Player for PocketPCs. Therefore, the user interface is portable to any device that has installed the player. If a Macromedia plug-in for the Pocket Internet Explorer is installed, the user interface can also be accessed as macromedia flash-film. However, since the interface is then re-scaled this has negative effects on picture quality, performance and usability. The PDA connects to the web services provided on our web server using an internet connection, e.g., via WLAN, Bluetooth or USB-connection (ActiveSync).

5 Conclusions

In this paper we have presented the concept of a meta search engine that can be used by desktop as well as mobile information retrieval interfaces. The system supports two types of annotations: Categorization based on information from user bookmarks (*Intelligent Bookmarks*) and semantic disambiguation of query terms using an ontology that can be selected by the user (*Sense Folders*). We have briefly discussed these approaches and how to implement them for building adaptive user interfaces for different devices (mobile and desktop) in order to accelerate the user’s search process. Furthermore, we discussed the advantages of dividing query results set processing (the information or content to be presented) from the interface design (information presentation) using web services in order to simplify the development of retrieval systems for standard desktop workstations as well as mobile devices.

References

1. Agosti, M., Crestani, F., and Pasi, G., Lectures on Information Retrieval, Lecture Notes in Computer Science. Vol. 1980, Springer Verlag, 2001.
2. Bade, K., De Luca, Ernesto W., and Nürnberger, A. Multimedia Retrieval: Fundamental Techniques and Principles of Adaptivity, In: Künstliche Intelligenz, 04, 2004.
3. Borgelt, C., Nürnberger, A., Fast Fuzzy Clustering of Web Page Collections, In: Proc. of PKDD Workshop on Statistical Approaches for Web Mining (SAWM), Pisa, Italy, 2004.
4. Brin, S., and Page, L. , The Anatomy of a Large-Scale Hypertextual Web Search Engine, In: Proc. of the 7th Int. World Wide Web Conf., pp. 107-117, Brisbane, Australia, 1998.
5. Brusilovsky, P., Methods and techniques of adaptive hypermedia, in User Modeling and User Adapted Interaction, 6:2-3, 1996.
6. Brusilovsky, P., Kobsa, A., Vassileva, J., (eds.), Adaptive Hypertext and Hypermedia, Kluwer Academic Publishers, 1998.
7. Brusilovsky, P., Stock, O., Strapparava C., (eds.), Adaptive Hypermedia and Adaptive Web-Based Systems, Proc. of the International Conference AH 2000, Trento, Italy, 2000.
8. De Luca, E. W., Nürnberger, A., Ontology-Based Semantic Online Classification of Documents: Supporting Users in Searching the Web, In: Proc. of the European Symposium on Intelligent Technologies (EUNITE 2004), Aachen, 2004.

9. De Luca, E. W., Nürnberger, A., Improving Ontology-Based Sense Folder Classification of Document Collections with Clustering Methods, In: Proc. of the 2nd Int. Workshop on Adaptive Multimedia Retrieval (AMR 2004), pp 72-86, Valencia, 2004.
10. Digital Equipment Corporation, AltaVista, http: //www.altavista.com/, 1995.
11. Dumais, S. T. Cutrell, E., and Chen, H., Bringing order to the web: Optimizing search by showing results in context. In Proceedings of CHI'01, Human Factors in Computing Systems, pp. 277-283, 2001.
12. Eisenstein, J., Vanderdonckt, J. and Puerta A., Adapting to Mobile Contexts with User-Interface Modeling, In: Proc. of the Third IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'00), pp. 83-94, 2000.
13. Grabski, K., Nürnberger A., A Search Interface for Context Based Categorization of Result Sets, In: Proc. of the European Symp. on Intel. Techn. (EUNITE 2004), Aachen, 2004.
14. Kartoo Homepage, http://www.kartoo.com/.
15. Labrou, Y. and Finin, T., Yahoo! as an ontology: using Yahoo! categories to describe documents. Proceedings of the eighth international conference on Information and Knowledge Management, Kansas City, Missouri, 1999.
16. Lam, H. and Baudisch, P., Summary Thumbnails: Readable Overviews for Small Screen Web Browsers, in Proc. of CHI 2005, Portland, OR, pp. 681-690, 2005.
17. Rocchio, J.J., 1971, "Relevance Feedback in Information Retrieval", IN: Salton, G., "The SMART Retrieval System", pp. 313-323, Prentice Hall, Englewood Cliffs, NJ.
18. Ruvini, J.-D., Adapting to the User's Internet Search Strategy. In Proceedings of the 9th International Conference on User Modeling (UM2003), Pittsburgh, 2003.
19. Stefani A., Strapparava. C., Exploiting NLP techniques to build user model for Web sites: the use of WordNet in SiteIF Project. In P. Brusilovsky, P. De Bra & A. Kobsa (Chairs), Proc. of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW, 1999.
20. Steinbach, M., Karypis, G., and Kumara, V.. A Comparison of Document Clustering Techniques, In: KDD Workshop on Text Mining, 2000.
21. Strapparava, C., Magnini, B., and Stefani A., "Word Sense Based User Models for Web Sites". 9th Int. World Wide Web, The Web: The Next Generation. Amsterdam, 2000.
22. Vivísimo Homepage, http://vivisimo.com/.
23. WordNet homepage, http://www.cogsci.princeton.edu/~wn/, („Five Papers on WordNet"), 2004.
24. Yang, J., and Filo, D. (1994). Yahoo Home Page, URL: http://www.yahoo.com.
25. Zhu, T.,Greiner, R.and Haubl, G.. Learning a model of a web user's interests. In The 9th International Conference on User Modeling (UM2003), Pittsburgh, 2003.