Handelshögskolan
VID GÖTEBORGS UNIVERSITET

Department of Informatics
2004-11-02

Federated search

Searching information across the AstraZeneca organisation

Abstract

Finding information that is stored among many different databases has become a serious problem because of the increasing number of searchable databases on local area networks and on the Internet. Many large organisations in change, suffer from this problem due to a large number of databases and many different uncooperative search tools. By using federated search a single search interface provides access to all searchable text databases in one moment. The question for the problem is: in what ways can federated search improve searching and help satisfy employees' information needs? A theoretical investigating of the resent research issues for federated search have been made in addition to a study of the search habits among employees' within AstraZeneca R&D organisation. The empirical study was made by semi-structured interviews. The discovered search problems are discussed in relation to the latest research issues for federated search. Some of the facilities sought for by the employees' where a user-friendly search interface, a structured result list in response to a search-query and a possibility to easily find interesting documents within the result list. All these facilities are solved with federated search.

Keywords: federated search, metasearch, distributed search, DIR

Author: Peter Mattsson
Supervisor: Dick Stenmark
Examensarbete II, 10 poäng

1 Introduction

Finding information that is stored among many different databases has become a serious problem because of the increasing number of searchable databases on local area networks and on the Internet (Si and Callan, 2003). Large organisations in change have today normally several different information systems. These systems are often not integrated with each other. There also may be different search tools for different environments with no possibility to search information across the organisation. The users of information systems in these environments must because of that, search information in several different sources to get adequate information. Large organisations in change may therefore be among the first to concerning this as a serious problem. The intranet in a large organisation in change becomes larger and larger, the number of text databases increasing as well and besides that, external information becomes more and more important. Sooner or later the organisation becomes aware of the amount of time the employees' within the organisation have to dedicate to search appropriate information. By then, such an organisation starts to think of solutions of the problem. Two ideas that might come up is either another new database in replace for all the existing ones, or turning all documents into cross-linked HTML-documents that makes the documents searchable by a single search engine (Bawa et al, 2003). However, both these solutions are impossible to carry out because they would take too much valuable time in demand to put into effect.

The problem mentioned above exists in the R&D (research and development) organisation part of the company AstraZeneca. They have therefore made efforts to accomplish an improvement in the information systems within the organisation. Such an improvement should enable search across the organisation. According to Si and Callan (2003), a preferable solution would of course be a single, uniform search interface that provides access to all of the searchable text databases and HTML-documents available, and this is possible with federated search. Federated search involves building resource descriptions for each database, choosing which databases to search for a particular information need, translating the information need into a form suitable for each selected database, and merging of retrieval results into a single result list. This could be a way to overcome today problems and a way to improve all the systems. Federated search gives the possibility to search across the organisation and the possibility to reach a broader set of information sources with one single search. An investigation of the possibilities to introduce federated search capabilities to the information system is therefore carried out within the AstraZeneca R&D organisation. Besides that investigation a more objective study of employees' search habits is desirable together with an academic view on federated search. One empirical study and one theoretical study together show similarities and differences between problems in real-life conditions and theoretical problems found in the latest research about federated search. Federated search is a promising solution for a growing problem and worth learning more about.

1.1 Purpose and question at issue

One purpose is to give an overview regarding federated search and in particular the resent research issues. For this reason a compilation of the latest research on Federated search has been carried out. Another purpose is to illuminate search problems experienced by the information system users within AstraZeneca. A qualitative study of today's situation has been carried out for that reason. The study focuses on search habits among a selected group of (13) employees. The

discovered search problems are discussed in relation to the latest research on Federated search. This leads to the question at issue:

In what ways can federated search improve searching and help satisfy employees' information needs?

1.2 Outline

Chap 2: 2 Method

In this section the population in, and the procedure for, the empirical study is described. The source used for the theoretical study and the procedure for the theoretical study is also described. In the end of the section the method is discussed.

Chap 3: 3 Federated search theory

In this section a general description of Federated search is presented. Components and different shapes of a metasearch engine is described as well. The metasearch engine is compared with traditional search engines, and problems related with metasearch is mentioned.

Chap 4: 4 Current research issues in Federated search

Current research issues is presented in different categorise in order to give a clearer understanding of metasearch capabilities.

Chap 5: 5 Empirical results

In this section a compilation of the interviews is presented.

Chap 6: 6 Discussion

In this section the relation between 'the current research issues for federated search' and 'the interview result' is discussed.

Chap 7: 7 Conclusion and further work

In this section the discussion from the previous part is concluded. A suggestion of further work is made and a good reference for the further work is given as well. The further work gives an example of another solution that may help take control over a complex network.

2 Method

2.1 The empirical study

2.1.1 Population in the empirical study

The population consists of employees at AstraZeneca, using computers for search and/or retrieval purposes. The population is finite to R&D (research end development) organisations in Mölndal and Lund in Sweden.

A systematic selection is made from the population for two reasons. Partly to get as much information as possible, which can be accomplished by using respondents that are assumed to have a great knowledge about search/retrieval. Partly to get as big a variation as possible which can be accomplished by using respondents from different departments (environments).

Respondents in the empirical study belong to the following departments:

Discovery IS	1 person
Discovery Biological Sciences	1 person
Development IS	2 persons
Experimental Medicine	1 person
Outcomes research	2 persons
Clinical science	2 persons
Clinical Chemistry	1 person
Epidemiology	1 person
Medicinal Chemistry	1 person
Informatics	1 person
Total	13 persons

2.1.2 Procedure for the empirical study

Semi-structured interviews were made with all respondents within the population. A form of questions (see enclosure 1) was used during the interviews to ensure that some specific questions were answered. The purpose of this form of questions was to get a more homogeneous picture of the today situation.

The respondents were chosen in what Lundahl and Skärvad (1999) describes as a snowball selection ("snöbollsurval" in Swedish). In a snowball selection the present respondent propose a new respondent and so forth. In this study it was not always the respondents that proposed new respondents, different informants did as well. The key objective was to find respondents that had great need of search and/or retrieval in their every day work. By using the snowball method for the selection of respondents it became easier to get in touch with the right respondents for two reasons. Firstly, people in the field of work had a good idea how to find the "right" respondent.

Secondly, when a potential respondent was recommended by a colleague he/she was more likely to take part in the study.

Each meeting with a respondent was arranged and took place in the respondent's office room. The interviews were between twenty minutes and one hour long. The respondent was informed about the objective of the interview which was said to be a mapping of today's search and or retrieval habits in the respondents every day work situation. Every interview followed the same pattern. The questions were most of the time asked in the same order in every interview. The first interviews were recorded but notes were also taken. After a technical hitch with the recorder only notes were taken during the remaining interviews. To ensure that the respondent had been correctly understood during the interviews, impressions and notes from each interview were summarised and mailed to each respondent respectively. The respondent then had a chance to correct any misunderstandings from the interview, before a compilation of the results from all interviews was made. Holme & Solvang (1997) recommend this line of action when analysing interview results.

2.2 The theoretical study

2.2.1 Source for the theoretical study

Several databases (see table 1) were searched for documents, about "federated search". One of the checked databases Association for Computing Machinery (ACM) was selected to be used for a more extensive search for, and study on, federated search. Search input were the words/phrases: "federated search" OR metasearch OR "distributed search". The Boolean expression 'OR' means that all search hits contained documents that had any of the words/phrases from the search input. The abstract of the documents in the ACM-database where searched when searching for interesting documents. The recalls from a search with the mentioned search input were used in the theoretical study. The recalls were in the study ordered by publication-date.

Table 1: Number of search hits in different databases from search made in October, 2004. Search input where "federated search" OR metasearch OR "distributed search".

Year	**Academic Search Elite** - Ebsco	***Emerald Library***	**Science Direct - Elsevier**	***ACM* (Association for Computing Machinery)**
2004	**41**	***3***	***4***	***15***
2003	**7**	***1***	***4***	***25***
2002	**6**	***1***	***3***	***24***
2001	**7**	***0***	***0***	***24***
2000	**7**	***0***	***1***	***11***

1999	**15**	***0***	***1***	***6***
1998	**7**	***0***	***1***	***0***
1997	**7**	***0***	***0***	***5***
1996	**6**	***0***	***0***	***2***
1995	0	***0***	***1***	***1***
1994	0	***0***	*0*	***1***
1993	0	***0***	*0*	***1***
1992	0	***0***	*0*	*0*
1991	0	*0*	*0*	*0*
1990	0	*0*	*0*	*0*
Total	**103**	**5**	**15**	**115**
<1990	**0**	**0**	**1** (1989)	**0**

2.2.2 Procedure for the theoretical study

The ACM database seemed to have many articles. with a general approach on federated search. Besides that the ACM database contained the largest number of hits in total. The 'Academic Search Elite' seamed to be a good choice but had too many documents that not where research articles. As a starting point for the search, the "abstract" was used as target for the search because it is not unreasonable to say that an article of interest for this study at least should have words associated with federated search in the abstract. As mentioned in '2.2.1 Source for the theoretical study' the search input when searching the ACM database were: "federated search" OR metasearch OR "distributed search" (see figure 1). If any of these words/phrases where found in an abstract it resulted in a hit. The applied combination of words/phrases where found to generate most documents in return and where therefore chosen to be used in the study of resent research issues for federated search. Other combinations of search input where tested as well. When more than three words/phrases were used the number of search hits decreased. The same thing happened when only one or two words/phrases were used. The word "metasearch" generated more hits than the phrase "meta search" or the word "meta-search". The word "search" was replaced by the word "retrieval" which resulted in fewer search hits. There may be groups of words/phrases than these used here, that generate more search hits, but that is of less importance. It was desirable to get as many search hits as possible but that was not the sole target of the study.

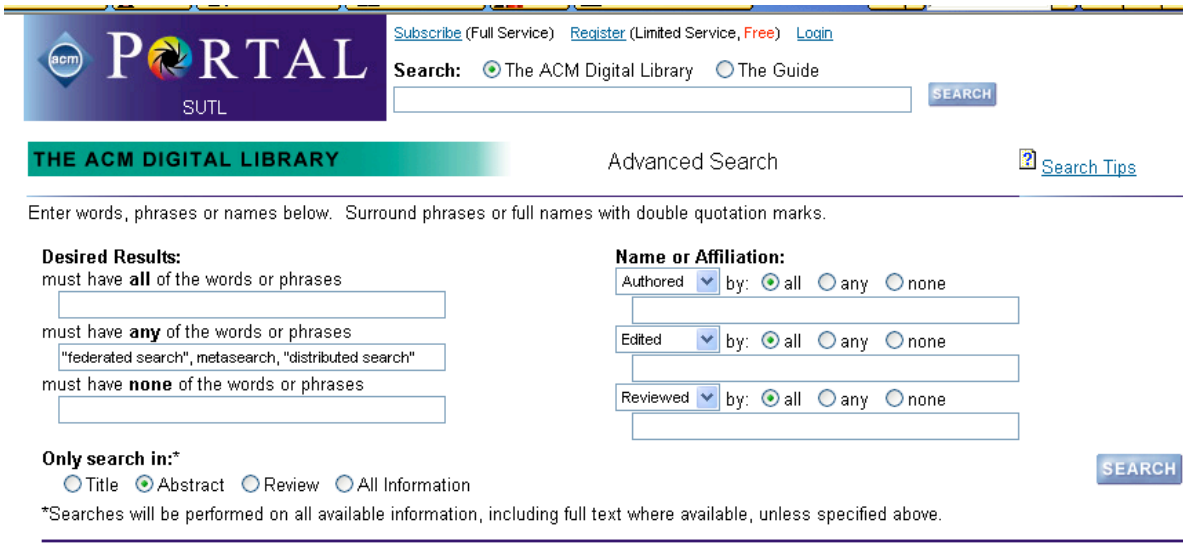

Figure 1: The interface for Advanced Search at the ACM digital library.

The search hits were sorted by "publication date". Fifty documents from the years 2004 and 2003 were then categorised regarding the subject of each article. Articles with similar subject were placed in the same category. If several, but not all articles in a category, in some way could be separated from the rest, then a sub category was made for this group.

2.3 Discussion the method

The choice of using semi-structured interviews instead of a questionnaire depends on difficulties in creating good answer alternatives when using a questionnaire. Too many answer alternatives had been needed and it had also been difficult to know what alternatives to use.

Letting the respondents correct the summarise from the interview brings the risk that the respondent gets a second thought about what was truthfully said during the interview and a chance to put a more favourable stamp on the interview result. On the other hand a summarisation of an interview does not always correctly reproduce the interview in the way the respondent wants it to do. This is especially the case when the summarisation is written in a foreign language, which was the fact in this case. In this study some of the respondents corrected their interview summarisation.

3 Federated search theory

3.1 What is federated search?

Federated search, **metasearch** or **distributed** search are all words used to describe the same phenomenon, and there may be more words than these. Baeza-Yates & Ribeiro-Neto (1999) give their definition of the concept.

> **"*Federated search:***
> *support for finding items that are scattered among a distributed collection of information sources or services, typically involving sending queries to a number of servers and then merging the results to present in an integrated, consistent, coordinated format."*
> (Baeza-Yates & Ribeiro-Neto, 1999, p. 442)

Korfhage (1997) is discussing **Distributed Document System** and means with this a system's ability to use distributed document sets and distributed processing. According to him, the typical user is interested in locating and obtaining documents regardless of where they reside. Further Korfhage says that the user would prefer to view the system as accessing a single logical database in response to a query, even when the system must consult multiple physical databases.

Baeza-Yates & Ribeiro-Neto (1999) is speaking about **Distributed Information Retrieval** (**DIR**) in a more technical matter. DIR is built on a Distributed Information System which is a set of server processes, each running on a separate processing node. The different server processes have the responsibility for different parts of the information managing where one designated broker process is responsible for accepting client requests, distributing the request to the servers, collecting intermediate results from the servers and combining the intermediate results into a final result for the client. Baeza-Yates & Ribeiro-Neto (1999) distinguish between engineering and algorithmic issues, where the latter is specific to information retrieval and the first to distributed systems in general. An algorithmic issue can be to deliver a particular search request to the appropriate server or combining the results from different servers. The engineering issue involves, among other things, defining a search protocol that specifies the syntax and the semantics of requests and results transmitted between clients and servers. Further the search protocol establishes a connection between the different parts of the distributed system, and specifies the underlying transport mechanism for communication (TCP/IP).
(Baeza-Yates & Ribeiro-Neto, 1999)

According to Baeza-Yates & Ribeiro-Neto (1999) and their description of a system for federated search, a custom made search protocol is required for a closed system consisting of homogeneous search servers and particularly if the customer requires special functionality such as encryption of requests and results. Otherwise a standard search protocol may be used with the benefit of a more easily interoperating with other search servers for the system.

Denzinger (2000) is trying to establish a concept for distributed search and gives as a matter of fact two of these, but first he gives his definition of the concept.

"defining transitions as undividable units and letting the computers do (different) transitions or transition chains in parallel
-> this we call distributed search"
(Denzinger, 2000)

This definition implies that distributed search is when a single and the same search query is being processed by different databases at the same time.

The two concepts Denzinger (2000) is trying to establish are both relying on agents, which are parts of a bigger system that have one common goal to accomplish. The first concept for distributed search is called the TEAMWORK method and means that all agents have the same abilities but differ in their strategies. This method is developed for distributed search processes that represent their states as sets of objects (for example, genetic algorithms that use sets of individuals). A system using this method is built with one Supervisor and several Experts with belonging Referees. In the beginning of a search each expert gets the whole problem to solve and they can work on this problem any way they wish. During the search process there are regular "team meetings" which means that the Referee sends an evaluation of its Expert together with a partial solution from the same, the Supervisor then determines the most successful Expert and sends the state of the solving process to all Experts. In the end of every meeting the Supervisor can replace the worst Experts. (Denzinger, 2000)

The second concept is called the TECHS approach, which stands for (TEams for Cooperative Heterogeneous Search) and means that the agents can have different abilities and therefore different search techniques and even different ways to express the same information. The concept applies only agents in the process which means no supervisor, instead the agents approving the search by assuming both the Expert role and the role of an Referee, here the Referee-role is separated in the two types Send –and Receive-Referee. A search starts with all agents working as searchers, then they stop for an evaluation of the result, the result is measured both in comparison with the particular piece of information used in the search and in comparison with other agents needs of this information. The result is then shared with the other agents. Since different agents may have different demands the agent in its role as a Send-Referee can give different information pieces to different agents. The agent then changes from Send-Referee to the role as a Receive-Referee. As a receive-referee the agent tries to predict and evaluate the impact the information pieces from the other agents might have on its own search. It selects the pieces that have positive impact and discard pieces that might only hinder its search. Then it translates the selected pieces into a form its searcher role can understand and integrates them into its own search. Then the cycle starts over again. Both the TEAMWORK concept and the TECHS concept result in a synergetic effect that leads to either a faster solution or a better solution within a given time limit. (Denzinger, 2000)

Meng et al (2001) also describe federated search but they use the word **metasearch**. They mean that a **metasearch engine** is a system that supports unified access to multiple local search engines. A metasearch engine for the web does not maintain its own index on Web pages, but a sophisticated one does often maintain characteristic information about each underlying local search engine in order to provide better service. When a metasearch engine receives a user query, it first reformats the query in many different shapes in order to fit all search engines and then

passes the query to the appropriate local search engines. In the last step the metasearch engine collects the results from its local search engines and sometimes it even reorganize them. Most existing metasearch engines employ a small number of general-purpose search engines as their underlying local search engines. Metasearch engines for the web as MetaCrawler and SavvySearch can cover a larger portion of the Web than any individual search engine. A good metasearch engine should have the retrieval effectiveness close to that as if all documents were in a single database, while minimizing the access cost. (Wu et al, 2001; Meng et al, 2002; Meng et al, 2001)

3.2 Components and different shapes of the metasearch engine

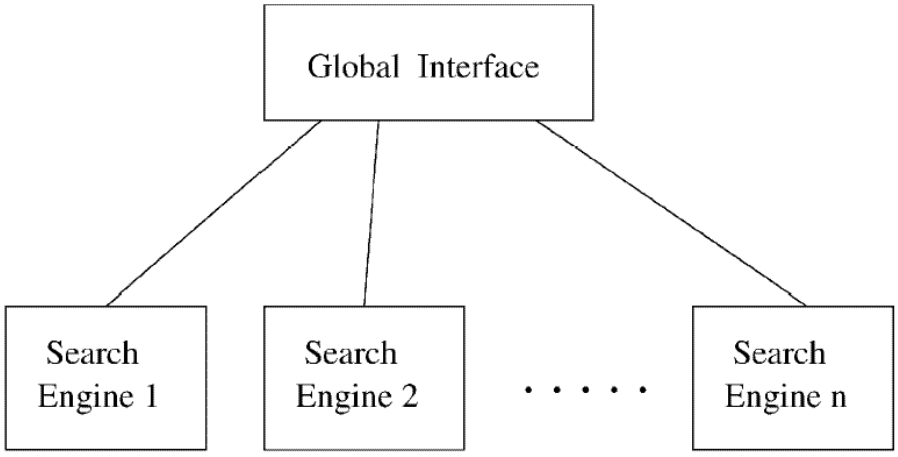

Figure 2. A simple metasearch architecture (Meng et al, 2002, p. 50)

According to Meng et al (2002), Baumgarten (1997), Gravano & Gracia-Molina (1995), Sheldon et al. (1994), and Yu et al. (1999) say that the two level architecture (see figure 2) can be generalized to a hierarchy of more than two levels when the number of underlying search engines becomes large.

Figure 3. below shows a two level architecture for a typical metasearch engine.

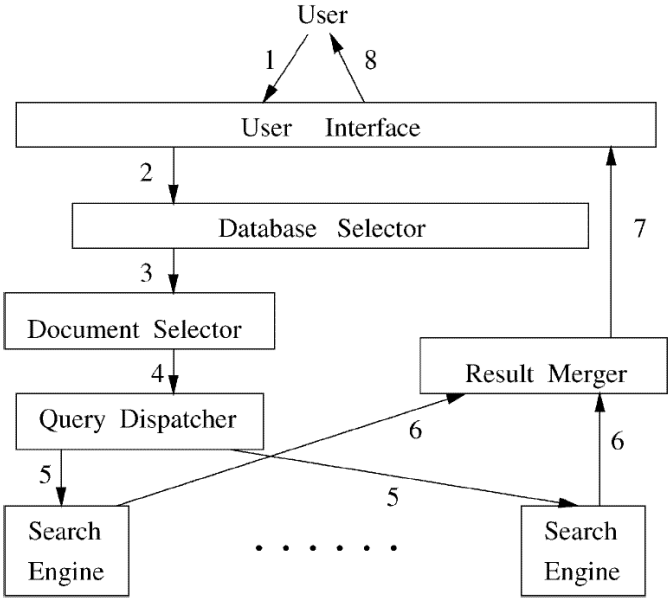

Figure. 3. Metasearch software component architecture. The numbers indicates a new step in the search process. (Meng et al, 2002, p. 55)

The **Database selector** shall correctly identify as many potentially useful databases as possible with as few useless databases (for a specific query) as possible, among the identified ones. (Meng et al, 2002)

The **Document selector** determines either the number of documents that should be returned from a database or a threshold which works as a measure for which documents that shall be allowed to be retrieved from the database. (Meng et al, 2002)

The **Query dispatcher** uses HTTP (HyperText Transfer Protocol) to establish a connection with the server of each selected search engine, and for data transfer. The GET –and POST methods like the query format may differ between the database engines, the query may for that reason be translated into a new query before being sent to a search engine. (Meng et al, 2002)

The **Result merger** combines the returned results into a single ranked list and a selected number of documents from the top of the list are then forwarded to the user interface to be displayed. A good result merger should rank all returned documents in descending order of their global similarities with the user query. (Meng et al, 2002)

Yu et al. (1999), propose a hierarchy of more than two levels if the number of databases is very large (thousands or tens of thousands). Further they provide an algorithm to search the hierarchy with the same effectiveness as the corresponding two-level hierarchy. Yu et al. (1999) have shown that the search of the hierarchy is efficient for single term queries which are submitted frequently in the Internet environment according to them. In addition to this Yu et al (1999) mean that the use of multi-term queries also is efficient in the "more than two levels hierarchy" under the circumstances that databases are clustered properly.

The more than two levels hierarchy involves superdatabases which each contains several databases. The next level of the hierarchy contains representatives of superdatabases (super-representatives) formed from local database representatives directly. The root node representative contains a representative for each local database and super-representatives. (Yu et al., 1999)

Yu et al. (1999) mention two reasons for using a hierarch of more than two levels: firstly they mean that this hierarchy solves the storage and efficiency problems, the amount of storage to contain all database representatives could be enormous, secondly they mean that the number of estimations needed to find proper databases for a search query significantly can be reduced.

According to Glover et al (1999), Barry (1993), and Schamber et al. (1990) says that studies have shown that users consider many factors, including some which are non-topical, when making relevance judgements. Further Glover et al. (1999) describes the next generation of Inquirus, the metasearch tool at NEC Research Institute. Inquirus architecture shown in Figure 4 makes certain user preferences explicit.

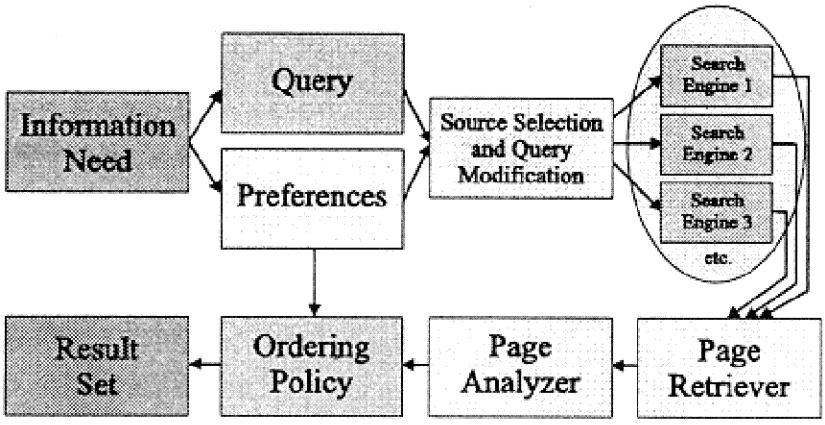

Figure 4. The architecture of the Inquirus 2 search engine. (Glover et al, 1999, p. 212)

The users **Preferences** used for improved "Sources Selection" is a reasonable model of user value. It is possible to determine how good any given source is for a given need on average (e.g. a user looking for news will usually prefer results from a site dedicated to news regardless of their query). (Glover et al, 1999)

The **Query Modification** allows the user to add query terms in addition to the provided query (e.g. the provided query 'information retrieval' is combined with the query term 'Research Papers' or 'General introductory about'). It can also mean the use of the search engines specific options like 'sort by date' or constrain to a language. The query modification is one method of causing the underlying search engines to provide more valuable results for a given information need. (Glover et al, 1999)

Inquirus 2 download the web pages and order them based on the full content. The **Page Analyzer** extracts the attributes (there are several different attributes) for every page. (Glover et al, 1999)

The **Ordering Policy** is "sort by value". Each user has selected different kinds of information needs where each category has an associated additive value (a value function that also involves the 'attributes') which means that different users, even with the same query and the same set of documents, will have results presented in an order meaningful to their individual need. (Glover et al, 1999)

Inquirus 2 has a dynamic interface that immediately shows results for the user as they are downloaded and scored, if the user is satisfied it's possible to stop the search process otherwise new results will be inserted as they are scored and if a result is "better" it will automatically be displayed on top. (Glover et al, 1999)

Han et al. (2003) say that top-ranked documents in search results frequently are irrelevant to what users are interested in and that this might be due to limited query capabilities (e.g., lack of Boolean query support), the poor ranking mechanism of search engines, a poor choice of keywords, and/or the problems of word synonymy and polysemy. According to Han et al. (2003), Zhao & Karypis (2002), and Cutting et al. (1992) give an approach to this problem that involves document clustering which provides intuitive navigation and browsing mechanisms by organizing large amounts of information into small number of meaningful clusters.

Further Han et al. (2003) say that Boley et al (1999), Chen & Sycara (1998), and Pazzani et al. (1996) give an approach to the problem based on personalization of information. These personalized information filtering systems typically try to find pertinent information based on the interest of the user as an individual or as a member of a group. (Han et al., 2003)

According to Han et al. (2003), iXmetafind shown in Figure 5 is the first product of its kind that has all the features: metasearch capabilities, personalization, and clustering methods.

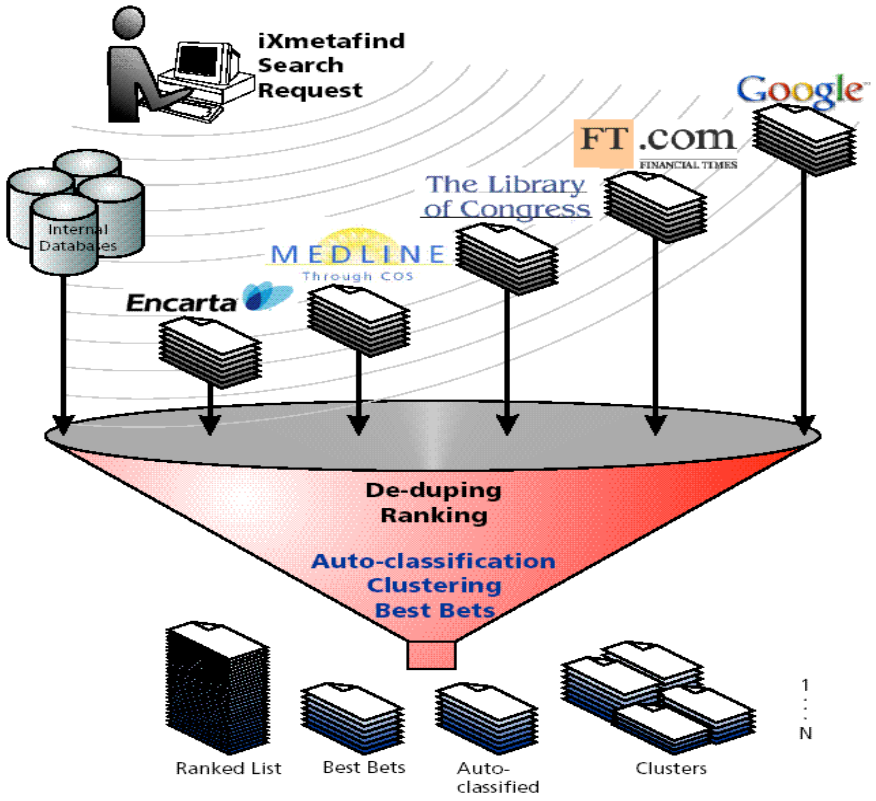

Figure 5. Main components and features of iXmetafind (Han et al., 2003, p. 493)

For each query, similar queries in a query history are searched and corresponding hits are returned which influences the final ranking of the hits for the current query. The **Ranked List** depends on the query history and may therefore differ for different query histories. (Han et al., 2003). This means that if the query history of 'recipes' is used for the query "salmon", recipes for salmon might come up at the top of the final hits. With the same query, when the query history of 'reproduce' is used, hits related to spawning might come up at the top of the hits. (Han et al, 2003)

The author of this essay interprets the **Best Bets** to be the assumed best (most interesting) hits from the query history combined with ditto from the current query. The set of hits from the query history forms a concept, and this concept is used to find the "**Best Bets**" from the past query history and from the search results of the current query. (Han et al., 2003)

Auto-Classification means that only documents closely related to a sample of documents will be identified and presented to the user. The samples are collected by the user and are assumed to belong to a topic or theme (e.g. business, sports, travel, books, and movies). (Han et al., 2003)

Clusters are made based on algorithms. The clustering mechanism finds groups of documents that are similar within the result set and provides sets of words that describe the clusters. Users

can easily identify clusters that best fit their needs and make a new better search with a new query that contains additional words in the clusters they found interesting. (Han et al., 2003)

3.3 Metasearch engine versus traditional search engine

Wu et al (2001) claim there are reasons to believe that many special-purpose search engines combined together can provide a better coverage of the Web than a few major search engines combined. Meng et al (2002) also advocate this idea. According to Glover et al. (1999), Lawrence & Giles (1999) say that no single search engine cover more than 16% of the estimated size of the publicly index able Web but that a metasearch engine that combines 11 major search engines could manage to cover about 42% of the same.

Web robots used by major search engines to gather data automatically may meet with opposition in cases when sites not allow their documents to be indexed but instead may allow the documents to be accessed through their search engines only (these sites are part of the so-called deep Web). (Meng et al, 2002). In the case traditional search engine versus metasearch engine, it is a question of finding search engines to enable search in the deep web. A metasearch engine provides a natural access to the deep web and the user do not have to find or/and use necessary search engines in order to reach the deep web.

In addition to the potential increased search coverage of the Web, the metasearch engine has the advantage over a large general-purpose search engine by the fact that the metasearch engine easier can keep index data up to date. The metasearch engine can use several local search engines that cover only a small portion of the web. (Wu et al, 2001). Meng et al (2002) describe the problem in related terms.

In view of using a metasearch engine for searching the Web, the metasearch engine requires much smaller investment in hardware in comparison to running a large general search engine such as Google which uses thousands of computers. (Wu et al, 2001). In addition to this, Meng et al (2002) claim that the metasearch engine approach is likely to be significantly more scalable than a centralized general-purpose search engine approach if the number of search targets is big (e.g. searching the entire Web).

A general-purpose search engine usually divides documents into different clusters where each cluster contains related documents. When evaluating a query, clusters related to the query first can be identified and then searched more thorough. The special purpose search engines can be used as natural clusters by a metasearch engine and in addition to this the special-purpose search engines may be better "clusters" than those created by the general purpose search engine. (Meng et al, 2002)

Meng et al (2002) give a good description of the reasons for using a metasearch engine:

> *"As an example, consider the case when a user wants to find the best 10 newspaper articles about a special event. It is likely that the desired articles are scattered across the databases of a number of newspapers. The user can send his/her query to every newspaper database and examine the retrieved articles from each database to*

identify the 10 best articles. This is a formidable task. First, the user will have to identify the sites of the newspapers. Second, the user will need to send the query to each of these databases. Since different databases may accept queries in different formats, the user will have to format the query correctly for each database. Third, there will be no overall quality ranking among the articles returned from these databases even though the retrieved articles from each individual database may be ranked. As a result, it will be difficult for the user, without reading the contents of the articles, to determine which articles are likely to be among the most useful ones. If there are a large number of databases, each returning some articles to the user, then the user will simply be overwhelmed. If a metasearch engine on top of these local search engines is built, then the user only needs to submit one query to invoke all local search engines via the metasearch engine. A good metasearch engine can rank the documents returned from different search engines properly. Clearly, such a metasearch engine makes the user's task much easier." (Meng et al., 2002, p. 50)

3.4 Problems related with metasearch

Some problems related to metasearch have been recognised and is presented below.

Problem 1:

The database selection problem is to identify, for a given user query, the local search engines that are likely to contain useful documents for the query. (Wu et al, 2001)

The objective of performing database selection is to improve efficiency as by sending each query to only potentially useful search engines, network traffic and the cost of searching useless databases can be reduced. (Wu et al, 2001)

Solutions of the given problem 1:

Most metasearch engines rank the databases for a given query based on certain usefulness measures. In order to perform database selection well, a representative for each database needs to be **stored** in the metasearch engine to indicate the contents of the database. In Wu et al (2001) an alternative to this method is described. This method (there is no name of the method) involves an integrated representative for all databases, with a size kept below a few Gigabytes regardless of the number of databases in the metasearch engine. This method has an advantage over other methods if the number of databases included in the metasearch engine is large (e.g. tens of thousands). Under all circumstances it has the advantage of only having to consider a small constant number of databases for each query during database selection. According to Wu et al (2001), the method is highly scalable in both computation and storage and for typical Internet quires (an environment with a great number of databases) the method retrieves close to 100% of the most similar documents. (Wu et al, 2001)

Problem 2:

To rank the documents received from several distributed sources as the user would prefer it to be ranked (the collection fusion problem) is a complex problem. One matching is to be made of the result sets given from the different sources in the search. Different subsystems may rank specific documents differently first of all because a document is compared with different documents in

different sources, besides this, there are more than one way to rank a single set of documents even from only one source. (Korfhage, 1997), and (Wu et al, 2001)

Solutions to the given problem 2:
The best way to overcome this problem is to let the main user system make its own ranking or rating of each document received. (Korfhage, 1997), and (Wu et al, 2001)

Problem 3:
Korfhage (1997) describes different problems that might appear when using one single system for reaching different databases allocated in other and different places. Different databases probably have different formats and therefore different processing requirements.

Solutions to the given problem 3:
To overcome this problem he suggests the use of subsystems, one for each different kind of database. The main systems role will here be to translate each search input into the appropriate form for each subsystem. (Korfhage, 1997)

Problem 4:
Documents may be copied and therefore stored in several different databases, this give rise to data redundancy in the search result when a search is made in several databases at the same time. To make a system able to identify and eliminate duplicates is not difficult but the problem becomes more difficult to solve when documents are changed without a significant difference to the new document's content. (Korfhage, 1997)

Solutions to the given problem 4:
Korfhage (1997) does not give a solution to this problem but says: -if a search input gives few answers in return, a substantial portion of the returned documents may effectively be copies of one single document. (Korfhage, 1997)

For metasearch engines some of the problems related to large general-purpose search engines remain, for example the inability to update index information quickly and the lack of the mechanism and effort to control the quality of indexed documents. (Meng et al, 2001)

4 Current research issues in Federated search

In an attempt to investigate current research issues in the area of "federated search" the 50 most recent published documents, from a search made in the ACM database, have been investigated. Categories have then been made for different research issues found in the documents. The most common considered issue among the investigated documents was database selection, and fusion of data sets collected from different databases. The latter issue includes ranking of the documents chosen to be presented for a user of a metasearch engine.

There is lack of detailed description for some of the methods and terms mentioned in this section. The over all purpose for this section is to demonstrate the current research issues and their respective performance.

4.1 Debate about federated search

Some of the documents found in the research discuss IR (Information Retrieval), including federated search, in more general terms and not necessarily the most recent research. Nevertheless, these documents gives a view of common research issues in the field of federated search that can help get a clear view of problems and terms related with the topic.

Rao (2004) gives his viewpoint of the terms 'metasearch' and 'federated search'. According to him search can be supported over multiple collections in a variety of ways, most notable by the following:

> **metasearch**: providing a search of models from each collection to find appropriate collections.
>
> **federated search**: brokering queries to multiple search services and combining the results.

Cruz et al (2002) from the University of Illinois at Chicago have made some work in the metasearch engine area with a focus on Web pages. They also give a simple explanation of how a metasearch engine works:

> *"When a query is received, the metasearch engine determines the best search engines (a very small number of search engines) to invoke for answering the query. After Web pages are retrieved by these selected search engines, they are ranked by the metasearch engine in descending order of desirability and then presented to the user."*
> (Cruz et al, 2002, p. 103)

Their previous work in the area consists of the following:

(a) Optimal selection of search engines for any given query.
(b) Selection of Web pages to retrieve from each chosen search engine.
(c) Merging of Web pages of selected search engines.
(d) Utilising linkage information among Web pages to perform search engine selection.

Their recent work in the area consists of the following:

(a) Automatic connection of a metasearch engine to numerous search engines.
(b) Personalised retrieval in which concepts are automatically associated with a user query, based on the users retrieval preferences.
(c) Automatic extraction of URLs retrieved by search engines.

(Cruz et al, 2002)

Allan et al (2002) reports from a workshop about challenges in information retrieval and language modelling, held at the University of Massachusetts. They bring up several statements, and questions without foregone answers, concerning metasearch. One of these statements hits the core soul of metasearch: to outperform the best underlying search engine on a per query basis. This can typically be achieved when combining systems of similar performance. However, this goal is often unachieved when combining search engines of widely varying levels of performance. This leads Allan et al (2002), to two questions: Can a metasearch technique be developed which consistently outperforms the best underlying search engine? Or can a technique be developed which is capable of distinguishing the good underlying systems from bad on a per query basis?

In the next part, Montague and Aslam (2002) presents a solution for internal metasearch. Their statement can be considered as an answer to the first question above.

4.2 Internal metasearch

Only one document considered this issue and is refereed below. Use of a metasearch engine to search one single database (internal metasearch). Under this circumstances the first question mentioned by Allan et al (2002) in previous part, can be considered as answered.

Montague and Aslam (2002) advocate the use of a metasearch engine in front of a single search engine in the case of searching one database only. They speak of this as internal metasearch and mention several advantages with this. In the case of searching one database with a metasearch engine, a specific document may be retrieved by several sub search engines. Internal metasearch therefore provides more consistent and reliable performance than individual search engines. Since metasearch aggregates the advice of several systems, the fusion tends to smooth out the errors of any one system, yielding a more reliable search system. A metasearch engine is modular and a highly specialised sub engine module can be developed for each information source about the documents in a collection (such as word frequencies, textual structure within a document, hyperlink structure between documents, etc.). By querying all the sub engines in parallel and combining their results using metasearch, performance is improved. In addition to this, metasearch leads to focused ranking algorithms that can take advantage of novel, specific information sources within documents.

4.3 Creation of metasearch engines

One document considered the issue of a late user interaction when building a metasearch engine and is refereed below. The document also gives a fine description of features in a general metasearch engine, except for item (3) which might be unique.

According to Wu et al (2003) previous building of customised metasearch engines are only capable of user interaction during the "building" process because the capability to connect to by the user chosen databases (search engines) needs to be established in advance. SE-LEGO, a web-based prototype system is on the other hand capable of creating metasearch engines on the fly. Only the URLs of the search engines to be used need to be provided to SE-LEGO. The system can be broken down into two major modules: Metasearch Engine Generator, and Metasearch Query Processor. Several components are needed to implement the two modules:

(1) Automatic Search Engine Connection: A component that analyses the source file of the interface of any given search engine and generates a program that can pass queries to the search engine.
(2) Automatic Search Result Extraction: For any given search engine, the component generates a program to extract the results (e.g., URLs) related to the retrieved documents from the result pages of the search engine.
(3) Search Engine Discovery: If a user submits a URL that does not have a search engine on the Web-page, the component will crawl Web-pages nearby to find search engine interfaces.
(4) Search Engine Representative Collection: For any given search engine, the component generates a representative for the search engine by collecting desired feature information from it.
(5) Search Engine Selection: For each user query submitted to the metasearch engine, the component selects, based on representative information of the underlying search engines, a small number of potentially useful search engines to invoke.
(6) Query Dispatcher and Result Merging: the component dispatches queries to selected search engines and merges the results extracted from returned pages into a single ranked list to present to the user.

(Wu et al, 2003)

4.4 Database selection

The problem is that of selecting a set of document databases to which a search query should be sent in such a way that the selected set of databases contains as many documents relevant to the search query as possible.

One way to find the best databases for a specific question is to use an algorithm for finding them. The Bayesian Inference Network Model of information retrieval ranks each database by the belief that the query find documents of interest in the database. The best ranked databases will then be selected for search. For example is the CORI-algorithm mentioned below based on this, yet much more sophisticated.

Powell and French (2003) have evaluated different algorithms used for collection selection (database selection) by demonstrating a uniform methodology for the study of collection selection approaches and their relative performance. A uniform methodology for the study was necessarily to gain insight into both the collective and individual behaviour of these algorithms. Three collection selection techniques took part in the evaluation: CORI, gGlOSS and CVV. The result where that the CORI approach consistently outperformed the other approaches, suggesting that effective collection selection can be achieved using limited information about each collection. Further on the experiments showed simpler approaches to be more effective.

Sogrine and Patel (2003) have addressed the problem of automatic selection of Web document databases in a distributed search system by investigating the available methods of database selection (gGlOSS, CORI, and CVV) and evaluating their performance. The result was that CORI algorithm performed generally better than gGlOSS and CVV methods for both long and short queries. They also found that using total number of term occurrences in a database instead of document frequency increased performance of CORI network even more.

4.4.1 Automate classification of databases or representative extraction

When an algorithm in a metasearch engine calculates the beliefs of finding documents of interest for the underlying search engines (databases), the algorithm typically need representatives (characteristic information about the database of a search engine) provided by the underlying search engines. The representatives is not always provided by a interacted search engine and the metasearch engine meets with opposition. To overcome this problem some efforts have been made to estimate needed information from uncooperative search engines.

Liu et al (2002) mean that in the Internet environment, each search engine is usually autonomous and managed with its own interest in mind, contents may be viewed as proprietary and the local search engine may not be willing to provide all the information requested by the metasearch engine, or worse, provide information that leads to an incorrect/inaccurate representation of its contents. Further on Liu et al (2002) describe how the number of documents indexed by a search engine can be estimated and how the maximum weight of different terms in the vocabulary of a search engine (how common a term is in the database contents) can be calculated.

Gravano et al (2003) consider the same issue as Liu et al (2002). They introduce Qprober which automates classification process by using a small number of query probes, generated by document classifiers. It use a variety of types of classifiers to generate the probes. It exploits the number of matches that each query probe generates at the database in question. Experiments show that the system has low overhead and achieves high classification accuracy across a variety of databases.

4.4.2 Combining representatives from topically related databases

According to Ipeirotis and Gravano (2004) techniques for extracting a document sample from a large database subsequently used to drive representatives (automate classification of databases), suffer from a sparse-data problem. The representatives tend to include the most frequent words, but generally miss many other words that appear only in relatively few documents.

Ipeirotis and Gravano (2004) introduce a technique to overcome this problem. To apply their technique each database first have to be categorised into a topic hierarchy by using representatives, either provided by the database or by automate classification. Representatives from similar topic categories can then mutually complement each other which generates high-quality representatives for these databases. For the best result in the database selection this technique should only be applied on representatives (databases) which indicates a high variance of possible scores, in other cases existing representatives is god enough. A database selection algorithm then use the fresh set of representatives to calculate a ranking list where the best ranked databases is used for the search.

4.4.3 User participation in database selection

A metasearch engine may disregard databases that a user would have preferred to pass the query to. The metasearch engine can't guess the users intentions with the query, just pass the query on to the appropriate sub search engines (databases). Due to the same reason, if a user has discarded a database earlier, documents from that source may appear in the result list when his/her source selection task proceeds and a metasearch engine is used.

Conrad and Claussen (2003) have analysed thousands of real user queries and show that precision can be significantly increased when queries are categorised by the users themselves, then handled effectively by the system. They say that in order to retrieve the largest sets of relevant documents for general users optimisation only the top-ranked databases is adequate. But this statement should not be applied on environments of professionals who require precise documents in response to their queries. In fact, such compromises in recall may be unacceptable. Besides the benefits of a metasearch engine the user is guided by navigation, by using attenuated decision trees. The decision trees provides the metasearch engine with more than just the query input which leads to a more specific result extraction in the end.

4.5 Fusion and ranking of results from different databases

The problem is simply to merge sets of documents received after passing a query to multiple search engines simultaneously, which is the case when using a metasearch engine. The merging issue strive for a single result list with the documents presented in descending order of desirability with the more interesting documents in front of the less interesting ones. There are several methods for doing this of which some initially where invented for merging results from a single search engine where scores assigned to the documents easily could be used for ranking. In the context of metasearch scores are usually not available from the sub search engines why rank based methods are more interesting. The score is a universal measure of the quality of a document and without the scores it's difficult to conclude differences in quality between documents from different sources. A major distinction between these methods is given by Renda and Straccia (2003). They say that the methods can be classified based on whether: they rely on the rank; they rely on the score; and they require training data (e.g. Bayes-fuse method) or not. According to Renda and Straccia experimental results seem to indicate that score based methods outperform rank based methods, while methods based on training data perform better than those

without training data. Another method is to download and analyse the ranked documents in order to produce the final ranking.

In the work of Renda and Straccia (2003) rank and score based methods, without training data, are compared in the context of metasearch. They report of experimental results for the rank based methods based on Markov chains. The result is that the Markov chain based methods perform comparable to score based methods. Further Renda and Straccia say that the Markov chain based methods do not rely on hits, but do rely on rank comparisons only.

Wu and Crestani (2004) have evaluated methods to merge results from databases which partially overlap. According to them this is still an open question. Their method requires scores obtained from the databases for all documents. The first step in their method is to find duplicates of documents. This is accomplished by comparing such as titles, authors and publishers of documents from different databases, though in most cases the URL is enough (for Web search engines). In order to rank the documents in a final list the scores are used as a measure. Documents that are duplicates gets a new score which is the sum of the scores of the same document in two different databases. One document may have different scores in different databases. To be able to compare the new score of a duplicate with scores of documents only occurring in one database the single documents gets a calculated score which is the sum of the real score and an assumed score for the single document if it had exist in a different database (a shadow document). This method Wu and Crestani call the SDM (Shadow Document Merging) method which relies on a coefficient which has been empirically determined by experiments.

The SDM method have been compared with linear regression methods like CombMNZ and CombSum, and methods that use min score (Min), max score (Max) or average score (Avg) for duplicates, when making a final list by merging result lists. The SDM method performed much better than CombMNZ and CombSum for the circumstance of heavy database overlapping and slightly better than Min, Max, and Avg in the same situation. (Wu and Crestani, 2004)

4.5.1 Learning systems

Si and Callan (2003) gives a new approach to result merging based on semisupervised learning (SSL). Their approach is especially adaptive for distributed IR in environments where there is little or no overlap in the contents of the selected databases, and where it is unusual for two independent search engines to return the same document.

By using query-based sampling, mentioned in the part ‘Automate classification of databases or representative extraction’ and normally used for database selection, a resource description for each database is created and stored in a new database (centralised sample database). This can give a good approximation of the scores documents would have received if they had been retrieved from a single global database. Usually the only input to a result-merging algorithm is document ids and scores returned from each of the selected databases. The SSL approach broadcast the query not only to the selected databases but also to the centralised sample database. The ranked list of document ids and scores returned by the centralised sample database are provided together with ditto from the selected databases to the result-merging algorithm. The database independent scores from the centralised sample database and the database-specific

scores from the selected databases can be used to teach a machine learning algorithm how to transform database-specific scores into database-independent scores. Given pairs of database-specific and database-independent scores it is possible to learn a function (e.g. linear regression function) that accurately map all database-specific scores into theirs corresponding database-independent scores. At least three, preferable ten, pairs must be found, else documents are separately downloaded and integrated into the centralised sample database. These separately downloads is rarely necessary. (Si and Callan, 2003)

4.5.2 User participation when merging document lists

Han et al (2003) refereed below is also referred in the part 'Components and different shapes of the metasearch engine' in the section 'Federated search'. Their product iXmetafind is a learning system but it also has personalisation facilities.

Han et al (2003) have created a metasearch engine with four alternatives for merging results. The merging can be chosen to depend on clusters, auto classification, best bets or else a normal ranked list can be obtained. The clustering algorithm group the results into topics which gives the user a possibility to easily navigate through search results by selecting relevant clusters (topics). Auto classification needs a document sample at the beginning. If a user has collected some articles they can be used as samples for different topics of interest for the user. Once a classification model is learned from documents representatives of a topic, all the search results become classified according to this model. Only the documents closely related to the sample documents of the classification model will be identified and presented to the user. The best bets is based on a query history containing the latest queries and corresponding hits (information content that users selected/liked), and is a concept used to merge the best results from the query history with the search results of a current query. A normal final ranking in iXmetafind is depending on the query history and by that depending on all users corresponding to it. Different query histories can be used though.

4.6 Automatic integration of Web search interfaces

Providing a unified access to multiple e-commerce search engines selling similar products allow users to search and compare products and prises with ease. Such search interfaces is carried out either manually or semiautomatically which is inefficient and difficult to maintain.

He et al (2004) introduce WISE-Integrator (Web Interface of Search Engines) that performs automatic integration of such interfaces. The tool explores a rich set of special metainformation that exists in Web search interfaces and uses the information to identify matching attributes from ditto for integration. This necessary metainformation is automatically extracted. A key problem for the issue is to identify semantically matching attributes across multiple interfaces. A two-step clustering approach based on positive –and predictive matches is shown to be a highly effective way to tackle this problem.

5 Empirical results

5.1 Compilation of interviews carried out in Mölndal and Lund

The situation concerning the internal network in Mölndal is much different at the time for publication of this study than during the study. New search functions have been added to the network and problems with the old search functions have been adjusted. This together has improved the functionality of the internal network in Mölndal to a great extent.

Num1(M), Num2(M), …, signifies the respondents where Num1 stands for number one etc, (M) stands for Mölndal and (L) stands for Lund. Num1(M) signifies the respondent from the first interview in Mölndal.

The English language use the word search which means search and the word retrieval which is trying to find (get in touch with) a specific, document for an instance.

5.1.1 Search and retrieval frequency

Most of the respondents search or retrieve documents and information, on the computer, once or a couple of times every day. Two of the respondents search and/or retrieve less frequently, and three respondents search and/or retrieve more often. Respondent Num1(M) does not search much on his own but instead he coordinates tasks to other persons to fill information needs of bigger importance, (search information in greater extent is not important for this respondent). The other respondent that searches and retrieves less is Num4(L), who approximately searches/retrieves twice a week. More extensive searching/retrieving is made by Num5(M), Num5(L), and Num2(L). Respondent Num5(M) and respondent Num5(L) make approximately one new search or retrieval every hour depending on time required for each search/retrieval. Respondent Num2(L) searches and/or retrieves information continuously in his every day work. The three more extensive "searchers" shall be interpreted as respondents searching/retrieving more frequently than the other respondents. It does not necessarily mean that these respondents search/retrieve less complex information.

5.1.2 Sources (most used)

The most commonly used search engine is overall Google. Google is used by all respondents once in a while. This search engine is frequently used even when it is not necessary. It assists as an extra resource in the hunt for information, to find out new ideas of interest for science or as a starting point for a search. As a result of this, some web sites/pages (External Network) become the most frequently used sources for information. Example of visited web sites/pages worth mentioning is MAPI, Authority for Regulation, different institutions of research, parts of the FDA, Autonomy, Vignette, Plumtree, and different universities. A variant of Google is 'Google Microsoft' which is used when searching information on sites related to Microsoft.

The most commonly used source for search/retrieval is Medline with its search engine Ovid. The second most used source is EMbase followed by Pubmed. The reason for this is that all

respondents have a need of basic information and this information is found in periodicals (web based). Some of the respondents are satisfied with this kind of information while other groups of users in addition to this have their special needs. Statisticians use SAS, Planet, EMbase and the internal library for their needs while Chemists use pdb and relibase for protein and protein-ligand complex structures, and isis and ISAC databases for small molecule searches, which both are examples of information needed by the chemists. At the Epidemiology they use GEL and a purveyor (partner) supplies with forms for questions which are available in an external database. Biological Sciences in Lund is to a great extent using E-lab, an AstraZeneca web-portal containing different searchable bioinformatics databases. The staff at Discovery uses 'Our discovery' for management of documents, an internal system only available for the staff at Discovery. At Discovery they also use AZsearch as a base for information retrieval, and searching in a greater extent. The AZsearch is internally reachable (not only for the staff at Discovery). In forum, the internal network within AstraZeneca in Lund, is like AZsearch available for all personnel and used in a greater extent by the respondents Num1(L), Num3(L), and Num5(L). In forum includes gateways as well, e.g. Planet.

There were two exceptions among the different respondents regarding the choice of sources. Num5(L) and Num8(M) used the sources: Outlook, shared files, and My documents in Windows file system, in a greater extent. The reason for this was frequent cooperation with other persons in there every day work. Specifically 'shared files' were used by another respondent as well, namely Num5(M) but he did not use that source much at all.

5.1.3 Sources (information of interest)

During the interviews some information related to specific sources appeared. According to Num3(M), the 'internal library' is used if the source of an article is familiar, otherwise Medline is used.

According to Num5(M), 'shared files' are sometimes used but it takes a lot of effort to retrieve information this way and he gives the following description: when he recalls that something of interest is written and stored among the 'shared files', the fastest way to find this information is to find the person who wrote this information. That person may know where among the heaps of 'shared files' the article or white paper is stored. The coffee break is used to ask around trying to find out who the author is. Windows search system is an alternative way to find information among the 'shared files'. This method can be used if at least a part of the title is familiar, but this alternative takes to much valuable time in demand. As said in "Sources (most used)" respondent Num5(L) and Num8(M) also used shared files, without complaints but not without problems.

According to all the respondents the 'internal network' is never or rarely used for search since this method has been very unsuccessful. One of the respondents (Num6(M)) shows how he tries to retrieve "something" he knows he should be able to find. The recall is zero with the headline as an input, the result is the same with related words used as input. Num8(M) becomes an exception here because he uses this source more frequently. He pointed out though, that the source was unreliable and that he needs to use other sources as well when using the internal network at Mölndal for search.

The Medline is possible to use both internally and externally within AZ. Respondent Num6(M) says he prefers using Pubmed (which is one external way to reach Medine) when using Medline (or the databases included in Medline) because he prefers the user interface and the facilities of Pubmed. Respondent Num2(L) is of the same opinion and says that the most common way for him to reach Medline is by Pubmed, because of the user-friendly interface.

5.1.4 Sources (number used)

The chemists need to use at least three different sources in their regular work. Information from one search/retrieval is used as input in the next, and so on. Half of the remaining respondents use more than one source when searching or retrieving information, and the other half only need one source. The respondents mention a number of reasons why using only one source. One respondent (Num3(M)) says he normally finds what he is looking for in Medline, another respondent (Num6(M)) believes he does not miss much when using Medline because periodicals presented at Medline are the most reviewed ones according to him. When Num1(L) searches information he normally uses only one source to find adequate information. Normally he knows where different kinds of information are stored, the source selected can differ from time to time. Num3(L) has similar experiences, he often knows where to find necessary information and only one source is therefore most of the time needed. The information is often found on a specific web-site and Num3(L) can easily find it. Num4(L) normally selects only one set of well known databases in Ovid for his searches. The facilities of Ovid give him the favour of only having to do one search instead of one search per database. Ovid is the search engine in Medline.

Other facts of interest regarding the respondents searching methods are that one respondent (Num5(M)) uses Google as a controlling tool. If an input generates few or no answers there may be better search words to use. In the case of no search hits the word or phrase might be wrongly spelled. Respondent Num3(M) starts the search with title as the target in a try to minimize the number of hits in recall. If the number of returned hits is too small he continues the search with the abstract as target. As input for search or retrieval there are some typical categories used. Besides “author” and “title”, forms for questions, names of substances, names of statistical methods, date, failure-messages, or by the chemists even pictures of molecule structures, and physical-chemical property criteria for small molecules, are used as search input. In addition to these categories, one or a group of single words, or phrases of different kinds are of course used by all respondents once in a while. One respondent, Num5(L), says that when scientific databases, reachable from In forum, is used for search, Boolean expressions like ‘or’ and ‘and’ is used mixed with single words. Phrases are never suitable where he searches or retrieves information.

5.1.5 Search and retrieval challenges

All respondents in Mölndal pointed out one common challenge related to search and retrieval; the possibility to search and retrieve information in the ‘internal network’. This possibility is by all respondents considered as non-existing. One respondent (Num6(M)) says it is because he does not know what is searchable there. Another respondent (Num5(M)) says it is because of the difficulties in finding adequate information there. Num8(M) does not consider this to be a challenge.

Another challenge is to find "the raisins in the cake" when the recall is big. One respondent (Num3(M)) says he finds it difficult to verify the most important result among a large number of hits. One way to get rid of all unwanted hits is to change and/or specify the search input, the same respondent considers this as a challenge as well. Respondent Num1(L) considers specifying a second search input after a first search with too many search hits as the major search challenge. Respondent Num7(M) considers the major search related challenge to be accomplishing a search without getting too much noise in the search result. A good example is when he searches the ISAC database using a list of identifiers and then has to find and extract the appropriate small molecule structure. The major search challenge for Num8(M) is finding a concept for search input that generates few and the right few answers in recall. One major search-challenge for Num2(L) is too put together or to get a research summary without getting lost in too many details. This problem appears when the number of search-hits is to big, which happens frequently according to Num2(L). In addition to this Num4(L) says that it is sometimes difficult to get a consistency from too much information, but he also says that it is sometimes difficult to find enough information and sometimes difficult to limit the search results.

Num3(L) considers the major search-challenge to be finding relevant and up to date information. The taxonomy for date can differ depending on, for example an articles origin, and it can therefore not be taken for granted that the search result is one hundred percent correct.

The biggest challenge for Num5(L) is to find out where a specific and known report or document is stored.

Respondent Num6(M) comes up with an other challenge; odd titles. The title 'A23/2:5' for an instance is impossible to use as a search input because all signs are not accepted by the search engine.

5.1.6 Ideas of improvement

The most wanted improvement, sought after by the respondents, is a more structured internal network in Mölndal. Other improvements or facilities the respondents had ideas about were for example:

An advanced help facility, or a user-friendly search interface that is easy to use even for inexperienced searchers. This could simplify searching and make it easier for Num1(M) to do more extensive and advaced searching on his own.

To improve search and retrieval Num6(M) would like to have a sort function for the recalls, a possibility to sort the results within a recall by date for example. Num1(L) is of the same opinion, one way to make search and retrieval easier for him could be to make it possible to pick out and search in just the interesting part of a source which he believes could be possible with a facility that allows the user to mix search and navigation. Num8(M) would like to have better presentation of the search results with a possibility to categorise the results by source or issue perhaps with graphics. Num4(L) is of the same opinion but he also takes the idea one step further, he saying that search and retrieval would be easier for him if he had the ability to split up subject

areas in smaller areas which then could be searched separately. Num2(L) says that a better structure of the search results would make work easier.

Num6(M) also has a viewpoint on the search engines that do not allows Boolean operators and/or truncation. These searching facilities both improve searching and retrieval.

According to Num3(L)'s major search challenge a uniform taxonomy for date would solve the problems with finding up to date information without missing a lot of information.

One user interface for several databases would make every day work easier for Num7(M). Especially with a possibility to do one search like in a relation database instead of as today, first search information in one database and then use the search result as input in a different database and so on. Num5(L) says in addition to this that a single user interface with access to several sources could make the every day work easier for him.

Another improvement could be to increase the access to different sources for some of the personnel. Several of the respondents had a feeling there was useful information they did not have access to. Some of the health economists for an instance, that use forms for questions in there work, felt they would like to be able to see clinical statistics from the source SAS which they today only are able to get if a person that works with the statistics is asked to give it to them. With this follows two problems. They may have to ask several persons who have been working with parts of the wanted statistics to get the whole part, and there is a problem for the asked persons to get access to the statistics if it is to old. The latter part is experienced as a problem even for the person who works with the clinical statistics. The statistics they once worked with they today have no access to.

Related to the latter case is the Num8(M)'s idea of having links and/or search possibilities for external sites/web pages and periodicals to make it more easily to reach these.

5.1.7 Importance

The importance of the possibility to search and retrieve information in the every day work is not to be neglected. Only one respondent (Num1(M)) considered information retrieval as a not very important or necessary part of the every day work.

Information is used as foundation for decision making, according to Num2(M) that part sometimes is very important. An other respondent Num4(M) says finding articles is necessary to be able to plan a new study. For Num3(L) it is very important to find the right information at the right time, since decisions are made depending on what information is known.

Num5(M) says that his entire work is depending on searching information and a good search tool is therefore of vital interest for him. Num5(L) is reviewing a lot in his work because what he produces is depending on what is exists today. Searching and retrieving information is an essential part of Num4(L)'s every day work. Both Num6(M), Num2(L) and Num7(M) say that information retrieval and searching is necessarily in their every day work.

Searching is very important for Num1(L) in his every day work. He gets help to solve problems behind cryptically failure-messages, he can be updated with the latest products, search for helping document and white papers etc. Searching makes every day work much easier for Num8(M), especially the problem solving issues.

Num3(M) says that he probably could manage his work without searching but that the search is making work much easier.

6 Discussion

In the part ‘Sources (most used)’ it is mentioned that Google is the most frequently used search engine, several reasons for this is mentioned as well. A result of that is that many different Web-pages are visited (by Google’s indexer/spider) and among them several different universities. Unfortunately a great portion of the Web (the deep Web) is probably not presented in the search result when searching with Google (see section 3.3 ‘Metasearch engine versus traditional search engine’). On this deep Web, online databases provide dynamic query-based data access through their query interfaces, instead of through static URL links. As a door to the deep Web, these query interfaces can be integrated in a metasearch engine. A metasearch engine is undoubtedly a possibility to get a much richer base of information, in the hunt for new ideas of interest for science.

In the part ‘Sources (information of interest)’ it is mentioned that two respondents prefer to use the interface of Pubmed because they think that is a user-friendly interface. In another part, ‘Ideas of improvement’ it is mentioned by another respondent that a “user-friendly interface” is believed to simplify searching. When query interfaces (user interfaces) for different databases normally differ in their usability the user-friendliness for different databases may differ as well. To accomplish user-friendliness for all query-interfaces, a new interface can be built on top of the other ones, which actually is the case when federated search is applied to the information systems. At least one single interface becomes more familiar than many different interfaces and the user get away from the feeling of not being capable of making a search like in the most preferred query-interface, this together gives a more uniform picture of the performance of the search. In addition to this it is not unreasonable to believe that one single query interface with a wider group of users is constructed with user friendliness in mind to satisfy all different kinds of users. In the next part more facilities for user-friendliness are mentioned.

When using Ovid, the search engine in Medline, it is mentioned in the part ’Sources (number used)’ that it is possible to select a number of well known databases and then do only one search instead of one search for each database. This resembles a metasearch engine but there are differences. As it is mentioned in the theory part ‘Debate about federated search’, a metasearch engine determines the best search engines to invoke for answering a query. The number of databases chosen by a metasearch engine does not necessarily have to be bigger than when using Ovid but a metasearch engine selects the most appropriate databases for an ad hoc query unlike Ovid. On the other hand some experts in the field of metasearch advocate user participation when choosing which databases a query should be addressed to. In the theory part ‘User participation in database selection’ it’s mentioned that professionals (e.g. chemists or legal experts) may require precise documents in response to their queries. This preciseness can be achieved with a browse functionality which guides the user to the right databases and away from superfluous databases. Regarding user-friendliness of query interfaces, attenuated decision trees integrated with metasearch may be considered as an user-friendly query interface for an experienced searcher who knows where to find appropriate information (documents). An inexperienced searcher on the other hand may not benefit from this, or worse, limit the number of good databases and in the end be lacking in information.

As mentioned in ‘ Search and retrieval challenges’ many of the respondents think it is difficult to receive good search results or at least problem of finding them among large amount of results. The general statement of this problem is that the respondents want few and good documents in their search result, which they often are not able to get. In previous part a solution with an attenuated decision tree integrated with metasearch was mentioned. This could be a good solution, especially for users searching information in wide topic areas (e.g. authority of regulation, legal area or chemistry). Another solution is mentioned in the theory section in the part ‘User participation when merging document lists’. iXmetafind can classify (auto classification) a sample of good documents received earlier and then in a new search only select the documents closely related to the sample documents to present to the user. Another facility of iXmetafind allows the user to navigate through the search results by different topic-clusters. A result list divided in different clusters can seem to be a good solution but is maybe only helpful when the user wants to separate recipes for apple pie from information about New York (The Big Apple) when the query for an instance is ‘apple’. Whether this cluster-mechanism can separate subtopics from a more general topic like economy is unclear. In this case iXmetafind’s auto classification, or its normal ranking solution based on a query history where documents more likely to be relevant is presented in front of the other documents, is more likely to be a good solution. In fact all research in the field of metasearch strives in the end for a good presentation of the search results with as many relevant documents as possible in the result list and preferable in front of the less interesting ones. But this is also the goal when only one database is involved in a search. Of course it is easier to get an overall picture of a short result list and determine what documents are interesting. But whether a short result list is a product of a quality search or the opposite is difficult to know. There is reason to say that a good result list in general should present as many relevant documents as possible for the user.

In the part ‘Search and retrieval related challenges’ it is mentioned that one respondent has a feeling of uncertainty when using date as search input in order to find up to date information. The feeling of uncertainty relies on experiences of missing interesting information due to different taxonomy for date. As mentioned in the theory part ‘Automate classification of databases or representative extraction’, a metasearch engine needs characteristic information about the database of an underlying search engine. This information is not always provided. In these cases a sample of the database can be extracted with several different queries. By doing this the metasearch engine gets knowledge about different terms in the vocabulary of a search engine and is then capable of choosing or not to choose that database for a current query made by a user. One term of interest should be publication date of a document. If the sample of documents extracted from an underlying search engine contains a lot of resent published documents (if this is interesting) or many documents from a specific year, the database is chosen for search. Because only a date rarely is interesting as search input the underlying search engine of course must allow some kind of Boolean expression, which most search engines do. The problem with a specific date as input still remains a problem because of taxonomy differences. If a search engine is unable to tackle this problem a metasearch engine should be. Metasearch engines have to communicate with underlying search engines in their “languages” and it is hard to believe that this does not include different taxonomies for date as well. There still might be a problem. Even if a metasearch engine picks a document sample from a database and identifies different taxonomies for date, which enables ranking of these documents by date, its uncertain if the metasearch engine is capable of producing new queries. Several queries (one for each taxonomy for date) is necessary to receive all date-specific documents if the underlying search engine is not

capable of recognise different taxonomies by itself. The metasearch engine is only capable of using the underlying search engines and is bounded by their performance. The user of the metasearch engine can of course make more than one query for date (one for each taxonomy) and then receive a properly date-ranked list. That should be the best way to solve the problem. Whether a metasearch engine is capable of producing queries on its own (not only adjust a query to fit a search engine) in order to fill the gap of lack in performance of an underlying search engine and its unstructured database, may be considered as an open question.

In the part 'Search and retrieval related challenges' it is mentioned that one respondent think it is difficult to know where a specific and known document is stored. A preferable way this question can be expressed is: How can I find a specific and known document? In this context the answer is given, namely metasearch. With headline as single target and with some words from the documents title as query input the wanted document should come up in top of the result list when a metasearch engine is used in the purpose. But the problem might be more complex than that. Maybe the document is not a Web-based document and is either a document stored in a database but for an instance among the shared files. Then a metasearch engine is not capable of finding the document. The only positive with not finding the document with the metasearch engine is in that case that the ongoing retrieval can be narrowed to non-database storage places and all the Web-based documents can be neglected as well. It's mentioned in the part 'Sources (information of interest)' that another respondent think it is difficult to find documents among the shared files much because the Windows search system takes to much valuable time in demand to find a document. It is unlikely that a metasearch engine would include such a slow search process in a global search. For that reason, if a document is not a Web-based document neither stored in a database, it is not able for a metasearch engine to use any search engine to find the document. In addition to this it is worth mention that the iXmetafind mentioned earlier in the 'Discussion' section, includes a facility called the best bets. If the iXmetasearch search engine has been used earlier and some interesting documents have been selected these documents have been stored in a query history. If this specific and known document mentioned above is among the documents selected earlier when using iXmetafind the best bets facility easily can present this document in the result list together with documents from a new search.

In the part 'Ideas of improvement' it is mentioned that several users sought after a better presentation of search results. One desirable improvement mentioned where a sort capability by date. Many single search engines have such capabilities and it is likely that most metasearch engines allow reranking of a result set by date or title for an instance. Another mentioned desirable improvement where a possibility to categories a result set by source or issue. Concerning categories of issues the iXmetafind mentioned earlier in this section has the closest solution for that with its clustering algorithm, able to group the results into topics which gives the user a possibility to easily navigate through search results by selecting relevant clusters (topics). As mentioned in a previous part of this section it is unclear how fine topic areas this clustering algorithm is able to produce. But it is still a solution worth considering. Concerning categories of sources, an attenuated decision trees integrated with metasearch (also mentioned earlier in this section) is a preferable solution that allows the user to refine the search to just the wanted sources.

Some of the improvements sought after are likely open questions. One respondent sought for the possibility to split up subject areas in smaller areas, which then could be searched separately,

another respondent were of the same opinion and wanted to be able to mix search and navigation. The closest solution for this is the iXmetafind but with that follows no possibilities to make a new search within a cluster. The clusters (topics) are a result of one search and the clusters has to be manually searched by the user. One respondent sought for an even more complex possibility. A search function like in a relation database where the answer from one search is used as input for a new search in another source and so on, which is what he has to do on his own today. With precise query input, clear structured databases and a fine search tool able to pick out just the right information this could probably be solved for some kinds of stored information. This is probably an open question and definitely not a research issue in the field of metasearch today. In the part 'Search and retrieval related challenges' one respondent thinks odd titles is a challenge and gives the example A23/2:5. A title like this is not possible to use as query input because some of the characters are not accepted by the search engine. A metasearch engine is using underlying search engines for the retrieval purpose and are bounded by their capabilities. A metasearch engine is therefore not able to solve such a problem. This also applies to the fact that all search engines do not allow Boolean operands (e.g. or, and), which is sought for by one respondent mentioned in the part 'Ideas of improvement'. Whether a metasearch engine in such a case pass two or more separate queries from a Boolean expressed query which means several result sets, just passing the first term in an Boolean expression, the term believed to be most important in an Boolean expression, or the whole expression as a single query, to a search engine that do not allow Boolean operands, is not clear. Hopefully this problem is taken care of in the search interface of the metasearch engine. None of the resent research considered this issue. The problem of search engines that do not accept Boolean operands may therefore either been solved easily or else such search engines may be rare and disregarded in the database selection and therefore not considered as a problem. If the latter case is a fact the problem is still an open question but that is unlikely. Many of the research issues mentioned in the section 'Current research issues' indicate that a metasearch engine often is aware of the database contents, and that many metasearch engines are capable of rank documents based on comparable measures. This means that a database is selected if it's considered as interesting, and interesting documents are collected irrespective of capabilities of an underlying search engine.

7 Conclusion and further work

In what ways federated search, and especially resent research, can improve searching and satisfy employees' information needs have been discussed. To enable this, an investigation of the resent research issues concerning federated search has been made. A number of employees within the company AstraZeneca have been interviewed about their search behaviour -and needs. The respondents sought for a single user-friendly search interface and a structured result list with mouldable capabilities in order to minimize the noise (the number of unwanted documents). Due to the fact that many of the respondents make extensive searching in their every day work, they want to make their searching more efficient. A "modern" metasearch engine (federated search) with some of the latest metasearch capabilities should most likely also include a user-friendly search interface. Federated search is able to provide a natural access to the so called deep Web, which generates a richer result set. A "modern" metasearch engine is capable of producing a single structured result list with the most relevant documents at the top of the list. When searching in wide topic areas like authority of regulation, legal area, or chemistry, an attenuated decision tree integrated with the metasearch engine can be used to guide a user to appropriate

databases if the user is aware of specific information within the wide topic area, and preferable where to find this information. Some metasearch engines use document samples as a template when retrieving documents. This can be a helpful way of collecting only relevant documents (documents that match the template) especially if the search query is badly formed. Automating clustering is another existing helpful tool that combines documents of the same topic in a cluster. The user can with the cluster mechanism chose to look at only documents related to a single topic. iXmetafind is a metasearch product that has both the cluster facility and the possibility to use document samples as a template when retrieving documents. Always when using a search engine and in particular when using a metasearch engine, there will be "noise" in the result list. No resent research in the area of federated search indicates of a possibility to make a second search within a result list received from a first search. Most "modern" metasearch engines improve searching and satisfy users information needs in an efficient way by producing good result lists in response to a query.

7.1 Further work

In further work a P2P file-sharing system could be investigated. P2P can make it easier to find non-HTML documents, which is typically not cross-linked like HTML, but often poorly arranged, in an intranet. YouSearch is a product on the market based on P2P and is architected to be extremely simple to use. YouSearch was released within the IBM intranet in mid-September 2002 and is said to be fast and efficient and satisfying users' need for search on personal webservers. An interesting referee to start with and refereed above is (Bawa et al, 2003).

References

Allan, J., Harper, D. J., Hiemstra, D., Hofmann, T., Kraaij, W., Lafferty, J., & et al. (2003). *Challenges in information retrieval and language modelling: report of a workshop held at the center for intelligent information retrieval, University of Massachusetts Amherst, September 2002*. ACM SIGIR Forum, Volume 37, Issue 1 (pp. 31 – 47). ACM Press, New York, NY, USA.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. ACM Press/Addison Wesley, New York, NY.

Bawa, M., Bayardo Jr., R. J., Rajagopalan, S. & Shekita, E. (2003). Make it fresh, make it quick: searching a network of personal webservers. In *Proceedings of the twelfth international conference on World Wide Web* (pp. 577 – 586). ACM Press, New York, NY, USA.

Conrad, J. G. & Claussen, J. R. (2003). *Early user system interaction for database selection in massive domain-specific online environments*, Transaction on Information Systems (TOIS), Volume 21, Issue 1 (pp. 94 – 131). ACM Press, New York, NY, USA.

Cruz, I., Khokhar, A., Liu, B., Sistla, P., Wolfson, O. & Yu, C. (2002). *Research activities in database management and information retrieval at University of Illinois at Chicago*. ACM SIGMOD Record, Volume 31, Issue 3 (pp. 103 – 108). ACM Press, New York, NY, USA.

Denzinger, J. (2000). *Distributed Search* [WWW document]. URL http://pages.cpsc.ucalgary.ca/~denzinge/projects/distr-search.html [Mars 2004]

Glover, E. J., Lawrence, S., Birmingham, W. P., & Lee Giles, C. (1999). Architecture of a metasearch engine that supports user information needs. In *Proceedings of the eighth international conference on Information and knowledge management* (pp. 210–216). Kansas City, Missouri, United States.

Gravano, L., Ipeirotis, P. G. & Sahami, M. (2003). *Qprober: A system for automatic classifications of hidden-Web databases*, Transaction on Information Systems (TOIS), Volume 21, Issue 1 (pp. 1 – 41). ACM Press, New York, NY, USA.

Han, E.-H., Karypis, G., Mewhort, D., & Hatchard, K. (2003). Intelligent metasearch engine for knowledge management. In *Proceedings of the twelfth international conference on Information and knowledge management* (pp. 492–495). New York, NY, USA.

He, H., Meng, W., Yu, C. & Wu, Z. (2004). *Automatic integration of Web search interfaces with WISE-Integrator*. The VLDB Journal – The International Jouornal on Very Large Data Bases, Volume 13, Issue 3 (pp. 256 – 273). Springer-Verlag New York, Inc. Secaucus, NJ, USA.

Holme**,** I**.** M**.** & Solvang**,** B**.** K**.** (1997)**.** *Forskningsmetodik, Om kvalitativa och kvantitativa metoder.* Lund**:** Studentlitteratur**.**

Ipeirotis, P. G. & Gravano, L. (2004). When one sample is not enough: improving text database selection using shrinkage. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data* (pp. 767 – 778). ACM Press, New York, NY, USA.

Korfhage, R. R. (1997). *Information storage and retrieval*. New York: Wiely, cop.

Liu, K.-L., Yu, C. & Meng, W. (2002). Discovering the representative of a search engine. In *Proceedings of the eleventh international conference on Information and knowledge management* (pp. 652 – 654). ACM Press, New York, NY, USA.

Lundahl**,** U**.** & Skärvad**,** P**.**-H**.** (1999)**.** *Utredningsmetodik för samhällsvetare och ekonomer***.** Lund**:** Studentlitteratur.

Meng, W., Wu, Z., Yu, C., Li, Z.: *Highly Scalable and Effective Method for Metasearch,* ACM Transactions on Information Systems, Vol. 19, No. 3, July 2001, Pages 310–335.

Meng, W., Yu, C., & Liu, K.-L. (2002). *Building Efficient and Effective Metasearch Engines*. ACM Computing Surveys (CSUR), Volume 34 Issue 1 (pp. 48–89). ACM Press, New York, NY, USA.

Montague, M. & Aslam, J. A. (2002). Condorcet fusion for improved retrieval. In *Proceedings of the eleventh international conference on Information and knowledge management* (pp. 538 – 548). ACM Press, New York, NY, USA.

Powell, A. L. & French, J. C. (2003). *Comparing the pereformance of collection selection algorithms*. ACM Transaction on Information Systems (TOIS), Volume 21, Issue 4 (pp. 412 – 456). ACM Press, New York, NY, USA.

Rao, R. (2004). *From IR to Search, and Beyond*, Queue archive, Vol. 2, Issue 3 (pp. 66 – 73). ACM Press, New York, NY, USA

Renda, M. E. & Straccia, U. (2003). rank vs. score based rank aggregation methods, In *Proceedings of the 2003 ACM symposium on Applied computing* (pp. 841 – 846). ACM Press, New York, NY, USA.

Si, L. & Callan, J. (2003). *A semisupervised learning method to merge search engine results*, ACM Transaction on Information Systems (TOIS), Volume 21, Issue 4 (pp. 457 – 491). ACM Press, New York, NY, USA.

Sogrine, M. & Patel, A. (2003). Evaluating database selection algorithms for distributed search, In *Proceedings of the 2003 ACM symposium on Applied computing* (pp. 817 – 822). ACM Press, New York, NY, USA.

Wu, S. & Crestani, F. (2004). Shadow document methods of results merging. In *Proceedings of the 2004 ACM symposium on Applied computing* (pp. 1067 – 1072). ACM Press, New York, NY, USA.

Wu, Z., Meng, W., Yu, C., & Li, Z. (2001). TOWARDS a Highly-Scalable and Effective Metasearch Engine. In *Proceedings of the tenth international conference on World Wide Web* (pp. 386–395). Hong Kong, Hong Kong.

Wu, Z., Raghavan, V., Du, C., Sai C., K., Meng, W., He, H. & Yu, C. (2003). SE-LEGO: creating metasearch engines on demand, In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 464 – 464). ACM Press, New York, NY, USA.

Yu, C., Meng, W., Liu, K.-L., Wu, W., & Rishe, N. (1999). Efficient and effective metasearch for a large number of text databases. In *Proceedings of the eighth international conference on Information and knowledge management* (pp. 217 – 224). New York, NY, USA.

Enclosure 1

Intervjuformulär

Metod

Öppen semistrukturerad intervju

Frågor

1. Visa / berätta hur du söker information i arbetet.

Används ett eller flera sökverktyg / sker sökning efter info på mer än ett ställe
Vad kan anses vara det mest vanliga / det mest ovanliga?

Hur?

>
>
>
>
>
>
>
>
>
>
>
>

Var?

Det grafiska gränssnittet: exempelvis i intranätet (här …), i google, osv.

>
>
>
>
>
>

Hur är informationssystemet uppbyggt?

Sker sökning på ett eller flera ord / fraser, både och. Hittas ”träffarna” i text eller i rubrik etc.

>
>
>
>
>
>

Vilka datakällor söker du i?

I vilka databaser söker du?

>
>
>
>
>
>
>

Vilken är den största utmaningen du brukar utsättas för vid sökning inom arbetet?

>
>
>
>
>
>
>
>

Vad skulle kunna underlätta/förbättra sökningen för dig rent praktiskt?

>
>
>
>
>
>
>
>
>

Hur ofta söker du information med hjälp av datorn, i arbetet

>
>
>
>
>
>

Hur viktig är sökningen för dig?

>
>
>
>
>