Werk

Titel: Integrated Access to Hybrid Information Resources

Autor: Sadeh, Tamar

Ort: Munich

Jahr: 2003

PURL: https://resolver.sub.uni-goettingen.de/purl?514854618_0013|log45

Kontakt/Contact

Digizeitschriften e.V.
SUB Göttingen
Platz der Göttinger Sieben 1
37073 Göttingen

info@digizeitschriften.de

LIBER QUARTERLY, ISSN 1435-5205
© LIBER 2003, All rights reserved
K.G. Saur, Munich, printed in Germany

Integrated Access to Hybrid Information Resources

by TAMAR SADEH

This paper is entitled "Integrated access to hybrid information resources", but integrated access is just a means to reach an end. Our challenge is to create an integrated environment in a heterogeneous world. Users are not aware, and do not want to be aware, of the differences between the various information resources that their institution provides. All they want is to be able to attain the information they are looking for, in the simplest and most straightforward way possible.

SIMULTANEOUS SEARCHING

Simultaneous searching refers to a process in which a user submits a query to numerous information resources. The resources can be heterogeneous in many aspects: they can reside in various places, offer information in various formats, draw on various technologies, hold various types of materials, and more. The user's query is broadcast to each resource, and results are returned to the user. The development of software products that offer such simultaneous searching relies on the fact that each information resource has its own search engine. The simultaneous searching product transmits the user's query to that search engine and directs it to perform the actual search. When the simultaneous searching software receives the results of the search, it displays them to the user.

Simultaneous searching is also known as *integrated searching*, m*etasearching*, *cross-database searching*, *parallel searching*, *broadcast searching*, and *federated searching* **[1]**. MetaLib, the library portal from Ex Libris, provides such simultaneous searching with its Universal Gateway component. In this paper, we shall refer to these systems as metasearch systems.

Let's take a look at an example of a metasearch process that a user carries out via MetaLib or a similar product **[2]**.

A student is interested in the works of Henrik Ibsen. Since the student knows that Ibsen is Norwegian, she submits a search query in several Norwegian resources that she knows about, such as the catalog of the National Library of Norway, the catalog of the University of Oslo, and several archives maintained by the National Library of Norway—the television, radio, and newspaper archives. The student submits the query ***author*** = ***Henrik Ibsen*** to all these information resources. She then receives the results.

If they are displayed by resource, she can easily pick out the results that seem most relevant. Let's say that one result from the television archive is a program about the play *Peer Gynt*, written by Ibsen. Looking at this record, the student decides that she can focus solely on the work *Peer Gynt* rather than all of Ibsen's works. She then uses additional functions of the system to submit a second query, ***title*** = ***Peer Gynt***, to the same information resources. This time she receives different results, including the *Peer Gynt Suite*, composed by Edvard Grieg - a result from the radio archive that she did not obtain earlier. However, the Ibsen play *A Doll's House,* from the catalog of the University of Oslo, did not come back this time, although it was on the previous result list.

Let's take the process one step farther and ask another question: How did the student know of the resources relevant to her research? Of course, she could have been knowledgeable in this area and aware of pertinent resources. If she was just starting out, however, perhaps she concentrated on the default resources that her library has set, on the basis of her group affiliation, as a component of its gateway. Alternatively, she might have requested resources relevant to her subject, a specific geographic region, a certain type of material, and so on, thus creating a personal searching scope maintained by the system and available for reuse. In MetaLib, such functionality is provided through the Information Gateway component.

ONE-STOP SHOPPING

Most researchers today deal with content residing in a wide range of materials. For example, our student might want to access materials such as the script of the play in book form or PDF file, literary analyses of the play, various recordings of the suite, the score of the suite, or a video or poster of a specific performance. The immediate search result is typically a bibliographic record or other form of metadata describing the actual material. From the end user's perspective, the bibliographic records serve only as a means of obtaining the material itself. Users do not want to be bothered with technical issues such as the format of the material they seek and the software that they need to access it - the library OPAC, Adobe® Acrobat® Reader®, Microsoft® Word or PowerPoint®, MP3, the MrSid viewer, or any other software that handles specific types of files.

To provide users with convenient access to materials contained in a range of resources, multiple software products need to be integrated, and they should offer a seamless interface to users. The first type of information typically presented to users as a search result is a description of the material - the metadata - such as a bibliographic record representing a video. Ideally, the user should see the material on her screen - in this case, a video - without having to concern herself about how to find the actual material and how to view it **[3]**.

The link from the bibliographic record to the actual material can be direct, an explicit URL embedded in the metadata, as in the MARC 856 field of a bibliographic record in library catalogs. However, in many instances, the system must perform calculations to create the link - for example, when the bibliographic record resides in one information repository, such as an abstracting and indexing database, but the actual material resides elsewhere, such as in an e-Journal repository or the library's printed collection. The user expects to reach the actual material nevertheless. A library can make this possible by configuring a context-sensitive linking server, such as the Ex Libris SFX server (Van de Sompel & Beit-Arie, 2001), that links the user to the actual material as a part of a set of extended services and onward navigation options. Such links include the appropriate copy of an article, the holdings in the user's library OPAC or any other relevant OPAC, the institution's document delivery service, citation information, a periodical directory, Internet searches, and information about the book in Internet bookstores or content-based services such as those offered by Syndetics. The software determines the list of links on the basis of the information in the specific bibliographic record and the institution's subscriptions and policies as predefined by the librarians.

RESOURCE DISCOVERY AND INFORMATION DISCOVERY

The process of finding relevant materials for research falls, therefore, into two stages. First is the resource discovery phase, when the user locates the resources most relevant to the specific search. Next comes the information discovery phase, when the search is executed in the various information resources and the results are retrieved. Institutions strive to provide their members - students, staff, and researchers - with high quality resources that offer information of real value. It is up to the librarians to determine what constitutes the institution's collections, both physical and virtual, and set the collections' boundaries. Every member of the institution should be able to define a personal scope that derives from the institution's scope.

Once the user sets the scope of the search and submits a query, the information discovery phase begins. The metasearch system delivers the query to the selected information resources and returns the results to the user. The process requires that the system 'understands' the expectations of the resources regarding the form of the query, on the one hand, and the nature of the results, on the other. It is up to the system to convert the unified query and adapt it to the requirements of each searched resource, deliver the query in the form appropriate to each resource, receive the results, and manipulate them so that they comply with the system's unified format.

RESOURCE METADATA

The first question, therefore, is which resources are available and which of those are appropriate for the institution. No software can replace librarians when it comes to an understanding of the scholarly information arena; only they can select the resources that are appropriate and affordable for their institution. However, the selection of a resource is just the first step. Information about the resource, *resource metadata*, is necessary as well. The metasearch software needs to obtain descriptive metadata about the resource, such as its coverage and the types of materials that it offers, and makes it available to end users so that they can make a knowledgeable decision about the relevance of the resource to their needs. Furthermore, the system needs technical metadata regarding its impending interaction with the resource.

Resource metadata can be made available in several ways:

1. Resources can offer their metadata to any metasearch system that attempts to access them for the purpose of information retrieval.
2. A central repository can offer resource metadata to any metasearch system.
3. Metasearch systems can maintain their own repository of resource metadata.

The first method - that a resource describes itself when relevant - seems the best. Resources provide the most accurate information about themselves, information that other repositories need not replicate. As a matter of fact, the Z39.50 Explain function was based on this premise. The idea was that when external software needed to access an information resource, the software would extract the details of the impending interaction from the resource on the fly and use the information to formulate the exact steps of the interaction. Apparently, few vendors implemented the Z39.50 Explain function, and those who did implemented it in a variety of forms. The Semantic Web approach takes the idea one step farther. With this approach, a typical metasearch process involves an interaction between agents that exchange requests and information to construct the final product, which is the information requested by the end user. This is the vision, but today's Web does not allow for such interaction between agents, and, therefore, an automated interaction between the metasearch system and a resource's own search engine cannot be achieved at the present time (Sadeh & Walker, 2003).

The second method - building and maintaining a central repository - is under discussion by the new NISO metasearch committee, MetaSearch Initiative, which was formed in early 2003. Maintaining a central repository would assure the availability of resource metadata but would pose new challenges. First, a decision would need to be made about which kinds of resources such a repository would store. Then a format for the resource metadata would need to be specified, as well as protocols dictating the manner in which resource metadata find their way to and from the repository. Finally, a decision would have to be made about who is responsible for storing information in the repository and keeping it updated - the repository, by means of harvesting programs, or the resource

itself. Another undertaking similar to that of the NISO committee is the Information Environment (IE) Service Registry pilot project, driven by MIMAS, in the UK, in collaboration with UKOLN and the University of Liverpool. The purpose of the project is to provide a registry of IE collections and services and examine the feasibility of such a registry in terms of discovery, access, maintenance, sustainability, ownership, and scalability. The information science community is watching these initiatives with interest to see whether such repositories become comprehensive and robust enough to provide services as necessary.

The third method is one that various current metasearch products have already implemented. Each such product holds the metadata, both descriptive and technical, of all the resources that it can access. Products differ in the amount of descriptive metadata that they release to the end user and the way in which they display it. They also differ in the degree to which they implement the search interaction and hence vary in the amount of technical metadata that they store **[4]**. The method whereby each metasearch system maintains information about the resources has many drawbacks. The most obvious one is that every vendor of a metasearch system has to configure and maintain the resource metadata. Handling such a repository requires considerable effort and therefore depends on the capabilities of the individual vendor.

MetaLib, like other products, provides a repository that includes the metadata of all the resources that it can access. However, the metadata are not maintained as part of the software but stored in the MetaLib Knowledge Base, a repository of resource data and rules. The software itself does not include any information that relies on specific resources: it extracts the information from the Knowledge Base. This information enables the user to select the resources and the MetaLib Information Gateway to perform the actual search and retrieval. If, in the future, one of the first two options regarding the origin of the resource metadata materializes, MetaLib will only need to extract the required metadata from another repository.

THE METALIB KNOWLEDGE BASE

The MetaLib Knowledge Base is a proprietary repository provided to institutions along with the MetaLib software. The Knowledge Base holds two types of metadata about resources:

- Descriptive metadata, such as the resource's name, coverage, language, data types, and publisher. The user sees this information and, with it, can make a sensible selection of resources. It is the same information that enables the system to create resource lists based on the user's specifications and display them in a

comprehensive way. In short, this information serves the resource discovery phase described earlier.

- Technical metadata, such as the type of protocol that the resource supports, the cataloging format it uses, and the physical and logical structure of the records that it retrieves. We can describe this information as rules that define the flow, interface, and manner of searching and that the software uses for searching, retrieving the results, and manipulating them - that is, for the information discovery phase.

The resource metadata in the MetaLib Knowledge Base can be divided into global metadata and local metadata:

- Global metadata are that part of the resource metadata that is universal and does not depend on the implementation of MetaLib at a specific institution. These metadata include the name of the resource owner, the coverage, and the interfacing rules.
- Local metadata are institution-specific; they relate to the way in which the resource is used in the institution's environment and presented to the institution's members. Such metadata include elements of authentication vis-à-vis the provider of the resource, the authorization rules that apply to it within the institution, and the categorization information that the institution uses to enable the software to offer the resource in specific contexts. For instance, one institution might categorize a certain resource under *Medicine,* whereas an institution with a different orientation might categorize it under *Social Studies*.

Ex Libris maintains a master Knowledge Base, which is copied to every MetaLib installation. Automated routines ensure that the Knowledge Base at each installation is updated as necessary. Institutions localize the relevant metadata and add configurations to local resources.

SEARCHING AND RETRIEVING

The process of searching and retrieving in a heterogeneous environment is far from trivial. Each resource has its own expectations regarding the form and manner in which it receives queries; even if the resource supports a standard interface, such as the Z39.50 protocol, the metasearch system needs to make further adjustments so that the resource's engine will interpret the query correctly.

The types of information that the Knowledge Base maintains to enable the system to search include the following examples:

- Access mode: What kind of interfacing protocol does the resource employ? Is it a structured, documented interface, such as Z39.50, the PubMed Entrez protocol, or a

proprietary XML gateway? Or is it an unstructured HTTP protocol that dictates the use of HTML parsing techniques to access the resource?

- Password control: How does the user access a specific, licensed resource? Are a user ID and password required, which the metasearch system delivers when the connection is established? Should the software redirect the query via a proxy to grant the user access?
- URL creation: If a URL needs to be formulated to hold the specific query, what should the structure of the URL be?
- Character conversion: What character set does the system use at the resource end? Does the character set comply with that of the end user?
- Query optimization: How should the query be structured?
    - What is the exact syntax that the resource's system expects?
    - How should fields be mapped to the fields of the resource; for example, to which field should the system map the "author" field selected by the user for a specific query?
    - How does the system expect to receive an author's name? Should it be
      <last name><,><first name>;
      <last name>< ><first initial>; or in some other format?

Normalization: What should the system do when the search engine at the resource end does not support a specific type of search? For instance, what rules should be applied if the user looks for a specific subject but a certain resource does not support a search by subject?

Once the information is there, the metasearch system can indeed adapt a single, unified query to the requirements of the specific resource, as in the following example.

The user submits a query for ***title*** = ***dreams*** *and* ***author*** = ***Schredl, Michael*** in the following resources:

- Library of Congress (Z39.50 access to Endeavor's Voyager ILS)
- NLM PubMed (the Entrez HTTP protocol)
- HighWire Press® (HTML parsing)
- Ovid MEDLINE® (Z39.50 access via the SilverPlatter ERL platform)
- University of East Anglia (XML access to the Ex Libris ALEPH ILS)

Even when looking at one brick of the process structure - the query syntax - we can clearly see the differences between the resources:

- The Library of Congress expects this query string:
  *1=**Schredl, Michael** AND 4=**dreams***
- PubMed expects this query string:

*term=**dreams**+AND+**Schredl**+**M***

- HighWire expects to see the encoded form of the following URL:
  *author1=**Schredl**,+**Michael**&author2=&title=**dreams***
- Ovid's MEDLINE via ERL, although accessed by the same protocol (Z39.50), expects this query string:
  *1003=**Schredl-M*** AND 4=**dreams***
  (Note the phrasing of the author's name.)
- The ALEPH system at UEA expects the following encoded request:
  *wau=(**Schredl**, **Michael**) AND wti=(**dreams**)*

PRESENTATION OF SEARCH RESULTS

Up to now we have discussed only the flow from the user to the resource. However, now that the query has been processed, the metasearch system needs to get back to the user with search results. Typically the interaction between the metasearch system and the resource consists of two phases. The first occurs after the search has been invoked: the resource returns the number of hits and some kind of reference to the result set. This phase is important because it gives the user some information about the search and enables the user to refine the query before browsing through the results. For instance, if a user sees that there are thousands of hits, she can modify the query to be more specific and thus reduce the number of results. The second phase consists of retrieval: The metasearch system retrieves the number of hits along with the first few records for each resource. This information is shown to the user instantly, even though the query might result in hundreds or thousands of hits. Some systems, including MetaLib, allow for further retrieval upon request.

Why do the systems provide such limited retrieval initially? First, retrieval depends on the use of networks, which are still not as rapid as one would like. Retrieving hundreds or thousands of records over a network is an extremely time-consuming process, and users are not likely to wait until it is completed. Second, people have difficulty handling immense result sets; after seeing the number of hits for each resource, users are likely to refine their query to obtain fewer hits. Once retrieved from the resource, each result is converted to a unified format before the user sees it. The rules that define the manipulation of the retrieved data are part of the resource metadata, which, in MetaLib, is stored in the Knowledge Base. These rules include information about the logical format, the cataloging format, the script, and the structure of certain fields, such as the citation field. For further processing to take place, the metasearch system must be able to apply these rules and convert all retrieved records, regardless of their origin.

Such additional processing can include the unified display of the records to end users; the merging of result lists from heterogeneous resources into one list; the comparison of records to eliminate duplicates; the creation of an OpenURL to allow context-sensitive

reference linking; and the saving of records in whatever format is required. Consequently, functionality that might have been missing from the native interface of the resource, such as the provision of an OpenURL, is added to the same set of records by the metasearch system. However, the display of result lists is not as straightforward as might be expected. Users are well acquainted with Web search engines and therefore have solid expectations regarding the display. They would like their results ranked, merged into one list, and filtered for a selected resource. Furthermore, they would like to be able to sort results by various attributes, such as title, author, and date.

Given that only the first results are retrieved from the various resources, these expectations are not so easily satisfied. When the result sets are small, all records are in the system's cache memory and so the metasearch system can offer the expected functionality in a comprehensive manner. However, the larger the number of hits, the greater the value of merging, sorting, de-duplication, and ranking - and the more difficult these features are to provide. Consider, for instance, the merging of the lists. How should it be done? The number of hits may vary considerably from resource to resource. Would it be appropriate to merge the two hits received from one resource with the dozens or hundreds of hits received from another resource? And if so, in which order? Every resource returns results in a different sorting order - by date (ascending or descending), title, relevance, or another attribute of which the users are not necessarily aware. Because only the first records are retrieved, the issue of merging the results needs careful consideration.

Other issues are the sorting capability and relevance ranking that users expect to find when looking at results, even when the resource itself does not support such functionality. Does it make sense to rank and sort only those results that have been retrieved? Let's say that the metasearch system applies certain relevance-ranking algorithms to all retrieved records and sequences them accordingly in the display to the user. This display can be rather misleading, because the 'best' hits are not necessarily those that were retrieved first. It could well be that if the user asks for more hits better results will be retrieved. A similar problem applies to sorting: even though a system might enable the user to sort the records according to various parameters, this sorting would apply only to the set already retrieved.

MetaLib handles these issues by always allowing the user to see the results for each resource. If the resource supports sorting, the user can request that the result list be sorted. Then MetaLib submits the search to this resource again, asking that the entire set of results be arranged in the order requested by the user. Hence, the user indeed receives the first records of the whole set. MetaLib also enables users to explicitly request and obtain a merged set at any point. Such a set is already de-duplicated and sortable. Institutions are likely to limit the number of records that can be merged, to avoid lengthy waiting periods caused by the retrieval of large result sets.

LOCAL REPOSITORIES AND LOCAL INDEXES

End-users may wonder why other searching systems, primarily the Web search engines, are able to provide them with large sets that are merged and ranked. The reason is that these systems use a different type of technology to provide the users with search results. Metasearch systems are based on 'just-in-time' processing. The system does not maintain any indexes of its information landscape locally; only when the information is required does the system access the various resources to obtain the results. The approach of Web search engines is based on 'just-in-case' technology. Huge efforts are invested in preparing the information prior to users' requests so that when the information is needed, it is obtained immediately. Google™, for example, holds indexes for the entire World Wide Web, including not only pointers to sites but also information that enables the search engine to evaluate the relevance ranking of a site. When the user searches with Google, only the indexes are scanned - and the information that Google initially displays on the screen is not from the sites themselves but from this vast repository of indexes. The search engine provides the actual access to a certain Web location only when the user selects it from the list. Needless to say, huge computing power and disk space along with sophisticated technologies for harvesting, evaluating, and maintaining the information are necessary for such powerful tools.

The use of local repositories of indexes in the library environment started some time ago. As opposed to union catalogs, which actually replicate the information that is located in local catalogs, repositories such as MetaIndex from Ex Libris hold only the indexes to the bibliographic materials that are kept in the resources. An example is the MetaIndex implementation at the Cooperative Library Network Berlin-Brandenburg (KOBV), which preceded the metasearch systems a few years back: At KOBV, MetaIndex enables each of the consortium members to maintain its library system and cataloging conventions while the consortium provides a single search interface for end users. MetaIndex has now become a resource available to MetaLib at KOBV, along with other resources. No doubt that a local repository of indexes has many advantages. Information that is gathered and processed prior to queries can be organized, evaluated, and de-duplicated and therefore can be accessible to end-users in a rapid and comprehensive manner. However, maintaining such a repository has a major drawback: the repository is another system, with hardware and software, to create and maintain, and personnel must be available to take care of it.

Considering libraries' budget constraints and limitations in the technical expertise available to them, a combination of just-in-case and just-in-time approaches would be optimal for metasearch systems. Local repositories would be useful in the following cases:

- When no searching mechanism exists at the resource end. This situation is typical of various types of local repositories, such as those that hold research papers written by institution members or spreadsheets relevant to institutional activities; but it could

obviously apply to any other data that have not yet been made available to the public.

- When the information is scattered. A local repository may be worthwhile if several resources that are mutually compliant form a single resource of value to the institution. For example, a worldwide organization that has dozens of branches, each of which holds regionally relevant information, wants to provide a simultaneous search capability that will cover all the local information. Creating an index such as MetaIndex would be preferable to requiring users to search all the repositories simultaneously.
- When the interface is not reliable. Some institutions want to provide access to resources that are not always online or do not offer reliable networking for accessing them. In such cases, an institution might be better off harvesting the information and keeping it as a local repository.
- When preprocessing is important. Preprocessing tasks such as relevance ranking and the elimination of duplicate records can be of value for some institutions. However, a component like MetaIndex can provide a solution only if the search scope is defined and limited. For instance, at KOBV, the consortium catalogs represent a limited search scope; as a result, the mathematics department of the consortium was able to develop a sophisticated de-duplication algorithm that permitted the construction of a comprehensive MetaIndex component.

MetaIndex from Ex Libris is created through the harvesting of information from other repositories. One of the harvesting mechanisms is the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). The use of such harvesting protocols can facilitate the gathering of data and is applicable to a wide range of resources that are now becoming OAI compliant. Furthermore, MetaIndex itself can become OAI compliant, thus serving as both a resource for MetaLib and an OAI-compliant resource that enables other systems to harvest the data from it.

SUMMARY

The promise of a truly integrated environment in a heterogeneous world may not yet be a reality, but with the active involvement of all the stakeholders, significant progress has been made. Just a few years back, metasearch systems seemed like a dream; today they are already a building block in the information resource environment serving the academic and research community.

NOTES

1. The term *federated searching* is used by some to describe a process in which indexes are 'pregenerated'. We refer to this concept as 'just-in-case' processing, as explained later in this paper.
2. We provide this example only to illustrate the process; references to specific resources are not necessarily accurate.
3. The issue of copyrights is not discussed in this paper. In this context, we assume that the system that offers the material handles the copyright issues.
4. For instance, some products offer unified searching, but once the user requests the result record, the software links the user to the record in the resource's native interface. Such products do not need to maintain all the technical metadata that is required for manipulating the retrieved record and converting it to a unified format.

REFERENCES

1. Sadeh, T. & J. Walker.: "Library portals: toward the semantic Web". *New Library World* 104(2003)1/2, 11-19.
2. Van de Sompel, H. & O. Beit-Arie.: "Open Linking in the Scholarly Information Environment Using the OpenURL Framework". *D-Lib Magazine*, 7(2001) 3.
3. http://www.dlib.org/dlib/march01/vandesompel/03vandesompel.html

WEB SITES REFERRED TO IN THE TEXT

Ex Libris. http://www.aleph.co.il/

Information Environment (IE) Service Registry. http://www.mimas.ac.uk/iesr/

MIMAS - Manchester Information & Associated Services. http://www.mimas.ac.uk/

NISO MetaSearch Initiative. http://www.niso.org/committees/metasearch-info.html

The Open Archives Initiative Protocol for Metadata Harvesting.

http://www.openarchives.org/OAI/openarchivesprotocol.html

SemanticWeb.org - The Semantic Web Community Portal. http://www.semanticweb.org/

UKOLN. http://www.ukoln.ac.uk/

University of Liverpool. http://www.liv.ac.uk/

Z39.50. http://lcweb.loc.gov/z3950/agency/