The MyView Project: a Data Warehousing Approach to Personalized Digital Libraries

Jens E. Wolff and Armin B. Cremers

Institut für Informatik III, Universität Bonn,
Römerstr. 164, 53117 Bonn, Germany

{jw,abc}@informatik.uni-bonn.de

Abstract. The MyView project aims at the integration of both structured and unstructured bibliographic information from a diversity of heterogeneous Internet repositories like electronic journals and traditional libraries. Based on the user's individual information need MyView maintains a personalized warehouse for bibliographic data in a unified scheme, which is locally available for browsing, ad hoc queries and analysis. This paper gives an overview of the project, emphasizes research issues and describes the current state of the implementation.

1 Introduction

The recent development in multimedia technology and the growth of the World Wide Web will have profound influence on libraries of the future. Besides traditional libraries offering their bibliographic data on the Web, many research projects in the USA (Digital Library Initiative[1]), UK (eLib Project[2]), Germany (Global Info[3]) and other countries (see [18]) have invested in digital library development. Nevertheless, however libraries will look like and whatever information they will provide in the end, the general problem for the user remains the same: how to query distributed repositories of knowledge efficiently and effectively with regard to her personal information need.

The vision behind MyView is that of a personalized information space, tailored to its user's information need offering efficient query evaluation and customized result presentation, with browsing facilities (eg authorship or citation networks), ad hoc analysis and sophisticated ranking techniques (eg weighted search terms, best-match retrieval) and with the integration of all kinds of "libraries".

In the following we will discuss the concepts of the MyView system which supports the maintenance of a personalized collection of bibliographic data[4]

[1] http://www.dli2.nsf.gov
[2] http://www.ukoln.ac.uk/services/elib/
[3] http://www.global-info.org
[4] Bibliographic data are metadata consisting of title, author, publisher and year, for instance, and possibly a link to the electronic version of the corresponding document.

about "documents". It locates resources and gathers information from multiple heterogeneous distributed information sources containing bibliographic data as there are digital libraries, traditional library catalogues, pure text archives (eg FTP Server for Technical Reports) and semi-structured WWW pages (eg catalogues of publishing houses or electronic journals).

The rest of this paper is organized as follows. Section 2 provides an overview of the goals and concepts of the MyView project. A detailed description of the system is given in Sec. 3. Implementation aspects are outlined in Sec. 4. In Sec. 5 we comment on other work that is closely related to MyView. Finally, Sec. 6 concludes and points out directions for future work.

2 Goals and Concepts

The MyView project aims at supporting the user-friendly definition, generation and maintenance of collections of bibliographic data records which are relevant to a user's individual information need.

The system gathers catalogue information from a multitude of heterogeneous information servers. It presents them in a unified view and supports direct on-line reorganization, browsing and selection as specified by the user. MyView's goal is the shift from data-centered to user-centered information access, as observed by Watters and Shepherd [38].

To support the above mentioned new functionalities, MyView transforms the gathered bibliographic data records into a uniform scheme and stores them in a personal database. In the database community this approach has recently become popular as *data warehousing* (see [14, 40]). Efficient data retrieval and query post processing on the local warehouse can thus be realized.

To justify the use of the term warehouse in our scenario consider the issues discussed in [40]:

> "The topic of data warehousing encompasses architectures, algorithms, and tools for bringing together selected data from multiple databases or other information servers into a single repository, called a *data warehouse*, suitable for direct querying or analysis."

MyView retrieves potential relevant information from different sources in advance, based on the specification of the user's information need. Data are stored in a personal database and queries are exclusively evaluated against this single repository without accessing the original sources. The advantages of this redundant storage of bibliographic data are obvious: efficient and rapid query processing, lower net load in the long run, uniform scheme, customizable searching and ranking, annotating, and managing historical information. Of course, the drawbacks of redundancy and missing up-to-dateness have to be considered. But in our application the amount of necessary storage is reasonably modest and the topicality can be achieved by periodical updates of small portions in spare hours.

Even if the primary use of data warehouses is in the commercial segment for decision support, the term warehouse for MyView's local database is used on account of the affinity to the above mentioned characteristics.

The principle architecture of the MyView system is sketched in Fig. 1.

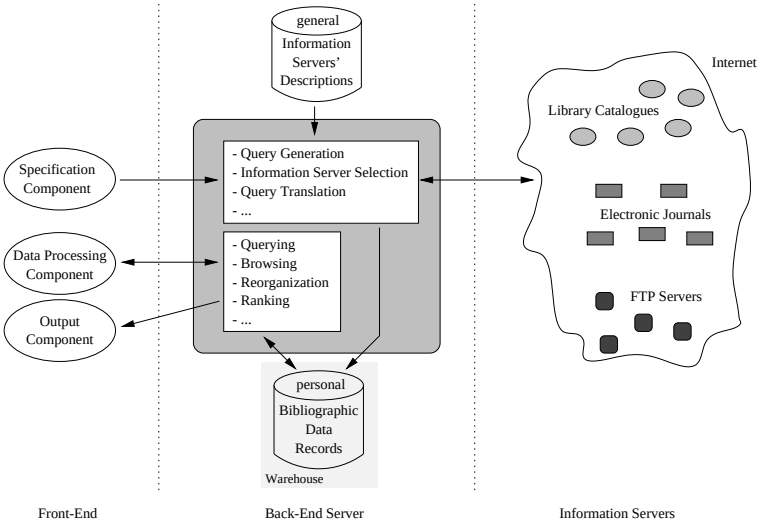

Fig. 1. MyView architecture

2.1 Front-End

The customizable front-end components (see Fig. 1 left-hand side) embody the interface to the user.

- The *specification component* allows the user to specify her individual information need.[5] Identifying the personal information need as the central motivation behind MyView, supporting its gratification efficiently is the logical consequence. In general retrieval tools of traditional libraries support only restricted functionalities and simple exact-match queries (Boolean retrieval model). Sophisticated query languages and best-match evaluation (vector space retrieval model) overcoming this lack are rarely supported by their interfaces.
  The standardized Z39.50 protocol [32], for instance, is widely used by traditional libraries and catalogue centers for supporting information retrieval services. Although many functionalities are defined in the standard, only

[5] Beside content related specifications other selection criteria concerning the information servers themselves have to be taken into account. We are currently working on automated resource selection integrating additional criteria like costs, language, and electronic availability.

a few are actually provided by every Z39.50 server, typically linking query terms by Boolean operators and counting the result size of a query.
It is generally accepted in the information retrieval community that Boolean retrieval is insufficient [7] and yields the worst retrieval quality in comparison with other retrieval models. Therefore MyView has to support functionality going beyond the existing services. So, bridging the gap between individual user-specific information needs and simple queries for information servers is a basic but heavy duty that must be mastered.
The description of the possibly complex information need is of essential significance. As the formulation of the specification itself should be user-friendly and simple MyView allows the user to define her information need by just entering a set of keywords and keyword phrases. This approach naturally raises the problem of mapping the intended meaning of a term set on the restricted query capabilities of existing repository interfaces. A detailed discussion of this aspect can be found in [42, 43].

- The *data processing component* provides all the functionalities to explore the personal local warehouse data. Our goal is to include, for instance, querying, browsing and ad hoc reorganization like sorting or formatting. Furthermore analyzing procedures (eg statistical analysis), annotations and the definition of views (ie subsets of the warehouse) should be supported. A view may, for example, be defined by all the literature from a certain database research group together with a list of title keywords. An ad hoc query may select documents about some topic which are accessible electronically (ie which have an URL).
  Functionalities for the management of the warehouse data are also provided like adding, deleting and updating bibliographic data. The underlying storage management system is interchangeable and not restricted to a special type. For example SGML/XML[6] files, the Lore DBMS [30] or other suitable systems can be used to manage the semi-structured information.
- The *output component* for instance, presents the query results to the user, displays browsing hierarchies and exports bibliographic data for further usage in different settings.

2.2 Back-End

The tasks of the back-end server are to gather bibliographic data from a multitude of heterogeneous information servers on the Internet and fill the local warehouse in accordance with the user specification. That means selecting suitable information servers, translating queries, loading results, transforming and storing them in the warehouse. This process – following Wiederhold's idea of a mediator architecture [41] – is done automatically and does not require any modifications on server side (information provider).

To let MyView select the appropriate bibliographic data repositories and interact properly with them some general knowledge is required about these servers

[6] `http://www.oasis-open.org/cover/sgml-xml.html`

like access methods (ie whether a search engine is provided or the Z39.50 protocol supported), data formats (eg a certain HTML layout, BibTeX, MARC[7] (Machine-Readable Cataloguing)) and so on. It is also essential to know which kind of information each server offers. All these general descriptive information are stored in a database.

2.3 Information Servers

The information servers are distributed over the whole Internet acting as bibliographic data repositories. Among these heterogeneous servers different types can be identified which reveal the complexity of the whole information gathering process.

Traditional libraries maintain large catalogues which are generated in a very disciplined way according to sophisticated rules like AACR2 (Anglo-American Cataloguing Rules) and represented in standardized formats like MARC and its derivatives. These libraries are increasingly becoming accessible via the WWW.

Bibliographic data as provided by libraries are at the one extreme. At the other extreme we have an FTP server without any additional information. Here the only information provided by the server are filenames and nothing else.

There may be reasonable forms of metadata in between these extremes. The Dublin Core for instance (see [39]) consists of a restricted set of 15 attributes (much smaller than in complex systems like USMARC) which should encourage the authors to describe their documents by themselves, but their use is optional.

Semi-structured WWW pages offered by electronic journals, for example, can also be seen as bibliographic data repositories, but this is a completely different case once more: In general, they do not provide the data in accordance with some generally accepted predefined scheme. When using XML in the future the information exchange will hopefully become much easier.

Nevertheless, all the above mentioned repositories should be accessible via MyView.

3 MyView System

The two main tasks of the system are building the warehouse and exploring it. In the following we will discuss both jobs and end with some remarks on customization.

3.1 Building the Warehouse

The process of building the warehouse according to user's defined criteria incorporates all the steps from specifying the information need, selecting appropriate resources, translating queries, querying internet repositories, transforming query results and storing the retrieved bibliographic data records. For reasons of space, we discuss only some of these aspects in this paper.

[7] `http://lcweb.loc.gov/marc/`

Specifying Information Needs. The specification of the individual information need is the initial task of constructing the warehouse. In the current implementation, MyView allows only the use of simple nested Boolean queries with operators AND and OR, such as `database AND (relational OR deductive)` to initiate the gathering process. Since the result set is not presented to the user directly but stored in the local warehouse for further explorations, the original query may be "generous" in some way.

However, the uncertainty in formulating a vague information need, particularly at the beginning of the work, should not be underestimated. Therefore we investigated in [43] a different approach. Users should be able to specify their interests in a simple and comfortable way. Building on experiences gained by the information retrieval community, we propose sets of weighted terms and best-match retrieval for this purpose. However, many on-line library catalogues and WWW gateways provide only a Boolean interface (exact-match retrieval). We have therefore to tackle the problem of mapping a set of weighted terms to an appropriate collection of Boolean queries, considering the restrictions of local warehouse resources and the generated net load.

In [43] we define the mapping problem and optimal solutions in exact terms. We develop two heuristic algorithms for the weighted and unweighted case and discuss some important implementation aspects.

Querying Internet Repositories. MyView gathers bibliographic data from heterogeneous information servers. Since the properties of the servers are sometimes quite similar, we assign different classes to the query components (wrappers) to reuse subcomponents. Currently MyView supports the following three classes of information servers:

- Z39.50: This class encapsulates servers offering database access via the standardized Z39.50 protocol. The only individually needed information are the host name, the port address and a few other parameter. These data are dynamically loaded from external files. New Z39.50 servers can thus be added very easily by extending these files.
- WWW gateways: This general class incorporates servers with WWW access, ie all those offering a fill-out form and returning results as HTML pages. Since the provided search engines and the layout of the result pages are extremely varying every such server is handled by a separate program. Adding a new server results in writing a new piece of code, compiling it and linking it to the system.
- Semi-structured documents: This class encapsulates servers presenting their information in semi-structured HTML pages, such as electronic journals or technical report collections without search engines. In difference to the above classes no explicit querying is needed. The bibliographic data are extracted directly from the HTML pages by using a rule-based layout description language. The rule sets for the different servers are stored in external files and are interpreted during the extraction process. Once again, new servers can easily be added to MyView.

There are, of course, other approaches for extracting and transforming semi-structured data (see [4, 5, 9, 25]). Our language was inspired by the pattern matching approach of Hammer et al. [24] and the extension of attribute grammars proposed by Abiteboul et al. [2]. The requirements of being simple, flexible, robust and that the layout structure should not influence the final storage structure motivated us to develop our own method. It is not an all purpose language and not as powerful as other approaches, but it is suitable for our application and easier to use. For a detailed description we refer to [26].

The query components realize the functionalities for collecting and transforming bibliographic data from different information servers. Since the individual steps are independent (uniform interfaces provided) the query components can be divided into subcomponents, which can be reused and combined. The correct combination of the subcomponents for each class is the task of the back-end server.

To query the heterogeneous information servers their individual characteristics have to be considered. Therefore a description of their features is indispensable. The necessary information are stored in an external file, the so-called *General Information Servers' Descriptions*. The metadata about the information servers can be divided into three categories:

- Information, that are absolutely necessary for identifying and contacting the server (eg name, query component class).
- Information, that describe the properties of the server (eg query language, result format).
- Information, that support the automatic resource selection of relevant servers (eg content description, word distribution, language).

A detailed consideration of the maintained attributes can be found in [36].

The task of describing query capabilities or general features of data sources has already been considered in different efforts. The STARTS protocol [21] (see Sec. 5) defines two formats for resource description and content characterization. It is partly integrated in our scheme. Other approaches like [33] consider rewriting techniques based on capability descriptions to take advantage of all the query power of the different sources. As a first step, we have focused on Boolean queries, because they are at least supported by most information providers. Using not all possible query capabilities is compensated by the retrieval facilities of the warehouse.

Storing Bibliographic Data. The retrieved query results have to be transformed from their heterogeneous formats into a uniform scheme to enable efficient data retrieval and processing. There are a diversity of formats for describing, storing and exchanging bibliographic information. When analyzing the demands of MyView we considered many of them, for instance MARC (library exchange format) [15], BibTeX (LaTeX bibliography format) [20], SOIF (Harvest)

[10], RFC 1807 (NCSTRL) [27], Dublin Core [39], RDF (Resource Description Framework) [31], MCF (Meta Content Framework) [12], Semantic Header [16] and TEI (Text Encoding Initiative) [6].

Basically all formats are more or less suitable for our warehouse scheme. Only the expense for modifications varies in order to meet our requirements. Every format was developed with a special application domain in mind (eg MARC for cataloguing in libraries, BibTeX for maintaining bibliographies in LaTeX) resulting in a special attribute set. But extending these sets lead to incompatibilities (eg RFC1807 has a fixed scheme) or makes it much more difficult to process them (eg repetition of the same attribute in SOIF or attribute hierarchies in "flat" formats like BibTeX). The SGML based formats like TEI, RDF, and MCF are very flexible, but too complex or not finally released.

In the end, we decided to define a new format for the MyView warehouse to meet all our demands. This approach should not be mistaken for a proposal of a new format. It is just for internal use. It comprises the common attributes of the previously mentioned formats, especially BibTeX, RFC1807 and Dublin Core, and partly represents the complexity of MARC by using a fine-grained structure of title and keyword attributes. The format is based on SGML to take advantage of its internationally standardization, flexibility and widespread use (many tools and applications). It uses only basic features to achieve compatibility to XML. For reasons of space we skip the detailed discussion of the maintained attributes (see DTD in [36]). Instead, to convey an impression of the stored information in the warehouse format and the underlying tree structure we present an example:

```
<!DOCTYPE metarec SYSTEM "metarec.dtd">
<metarec>
  <record>
    <sys>
      <source>test.bib</source>
      <srcid>rijsbergen79:inf</srcid>
      <add-date>
        <year>1998</year>
        <month>January</month>
        <day>13</day>
      </add-date>
    </sys>
    <names>
      <author>
        <name>van Rijsbergen, C.J.</name>
      </author>
      <publisher>
        <name>Butterworth</name>
        <address>London</address>
      </publisher>
    </names>
    <phys>
      <ident>
        <isbn>0-408-70929-4</isbn>
      </ident>
      <pub-date>
        <year>1979</year>
      </pub-date>
      <edition>2nd</edition>
      <type>book</type>
    </phys>
    <desc>
      <titles>
        <title>Information Retrieval</title>
      </titles>
    </desc>
  </record>
</metarec>
```

The information of a bibliographic data record is divided into five sections:

<sys> – information about the data provider
<names> – information about persons and organizations
<phys> – information describing the formal/technical properties
<desc> – information describing the content
<unknown> – information that can not be mapped or transformed, but should be available (not present in the example)

There is, of course, a great need for standardization to simplify the information exchange on the Web, as the many discussions about metadata formats show. But there will be still a great discrepancy between different objectives tied up with different demands (eg cataloguing information in libraries in extensive formats like MARC following sophisticated rules and a minimalistic set of 15 attributes in Dublin Core for describing networked documents). Therefore the explicit collection of metadata in non-uniform schemes will go on in the future. The only thing to pay attention to is a common basis and the chance for a simple transformation like in MyView.

3.2 Exploring the Warehouse

After the local warehouse is filled with potential relevant bibliographic data the user can explore the gathered information.[8] By now, we have implemented a Boolean query engine and an interface to a Lore DBMS. Browsing facilities are in preparation.

Boolean Queries. One possibility of querying the warehouse is the use of a WWW based interface for submitting Boolean queries (see Fig. 2). On the left-hand side attribute names or path expressions can be specified. The corresponding search strings are entered into the fields on the right-hand side. The search terms can be connected using the Boolean operators AND and OR. Furthermore the user can select case sensitive or insensitive processing.

The query depicted in Fig. 2 searches for all documents about logic which are accessible electronically. This is achieved by forcing the string "logic" to appear in an attribute "below" the node DESC (eg TITLE or ABSTRACT) and by checking whether the attribute URL exists. The query language is inspired by the subtree model proposed by Lowe et al. [28]. A discussion of this language and its features is beyond the scope of this article (see [26]).

Lorel Queries. As a proof-of-concept for the interchangeability of the underlying storage management system, we have implemented an interface to the Lore DBMS [30]. In addition the user can take advantage of the supported query capabilities of the Lorel query language (see [3]), assuming she is familiar with Lorel or OQL.

The following query, for instance, searches for all document titles containing the string "logic" in the attribute TITLE when knowing just the root node and the attribute name:

```
select T
from METAREC.#.TITLE T
where T grep "logic";
```

[8] The process of retrieving data from different sources may take some time. The user should not expect the system to establish the warehouse within a few minutes. Ideally, the process should be carried out over night.

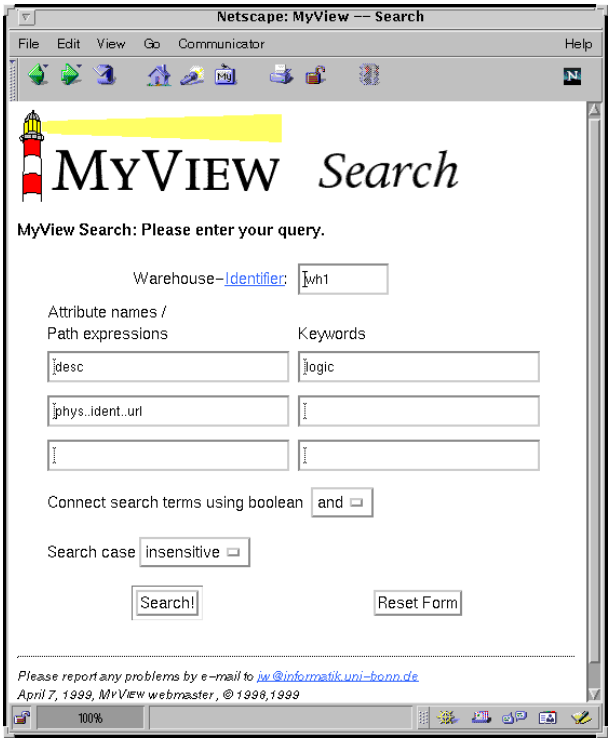

Fig. 2. Query form for searching

The above mentioned query (Sec. 3.2) for selecting all data records containing the string "logic" below the node DESC and having an URL entry looks like this:

```
select T,U
from METAREC.RECORD.DESC.# T,
     METAREC.RECORD.PHYS.IDENT.URL U
where T grep "logic"
and exists(METAREC.RECORD.PHYS.IDENT.URL);
```

To "discover" the structure of the underlying data, one can ask for all paths from the root node METAREC to the leaves named NAME:

```
select distinct path-of(P)
from METAREC.#@P.NAME;
```

One would obtain:

```
RECORD.NAMES.AUTHOR
RECORD.NAMES.CORPAUTHOR
RECORD.NAMES.EDITOR
RECORD.NAMES.PUBLISHER
RECORD.NAMES.CONTRIBUTOR
```

These are only some simple examples for the use of Lorel. Other more complex queries can be constructed.

3.3 Customization

The integration of information providers is done manually by an administrator (one day possibly customized by the user). It is intentionally not our goal to automate this process as the quality assurance should be up to an expert.

So far, we discussed the idea of MyView in the context of a single individual information system. But the same method can be applied in shared environments like project groups, departments or the like, where people with the same interests are working together. In such cases it pays off to have one central internal administrator. She maintains the information servers for common use that individual users wanted to be added. Sharing the same system enables the participants to search for information in the common warehouse and benefit from previously collected data. This results in faster response time and avoidance of redundant searches. The union of individual users with nearly the same focus of interest in fact is desirable to achieve a high synergetic effect in the long run: everyone knows different valuable information servers resulting in a highly relevant server collection for the common information need.

4 Implementation

The previous sections described the conceptual architecture of the MyView system. We now outline some implementation aspects.

4.1 Component Architecture

Let us begin with the back-end server. Figure 3 illustrates its structure and the connection of the separate components. The shaded areas represent the interfaces to the user (front-end) and to the Internet resources (information servers).

The WWW server (Apache[9]) establishes the contact to the user by presenting HTML pages (server data) for specifying the information need, managing and querying the warehouse and displaying query results. Additional server programs (CGI scripts) are necessary to add supplementary data to the HTML pages dynamically.

The construction of the warehouse is done as described in Sec. 3.1. The query components are divided into the three subcomponents query translation, server communication and query result transformation, which are supplied with metadata from the general information servers' descriptions.

The exploration of the warehouse is again realized through a WWW interface in combination with additional tools (sgrep, see Sec. 4.4).

[9] `http://www.apache.org/`

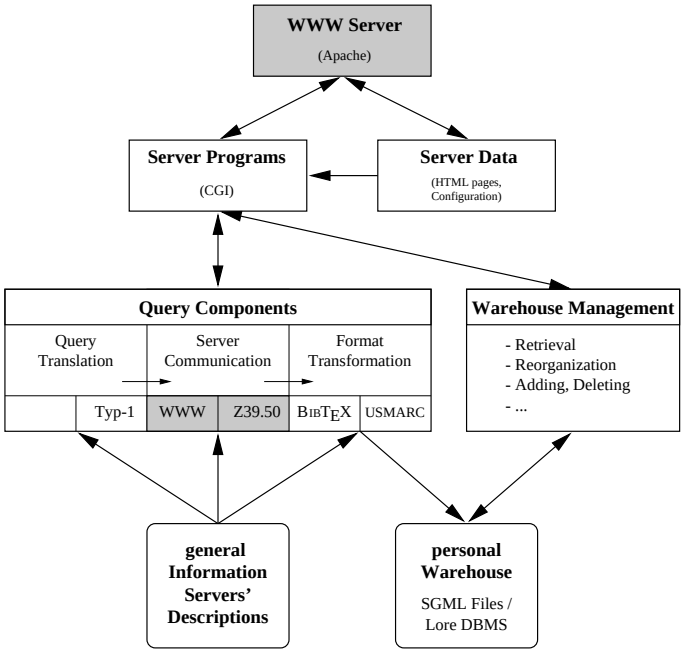

Fig. 3. Component architecture of the back-end server

4.2 Integrated Information Servers

At present the integrated information servers are[10]

- libraries offering a Z39.50 interface (Library of Congress Server, Bell Labs Server, On-line Computing Library Center (OCLC) Test Server),
- on-line catalogues with WWW gateway (The Collection of Computer Science Bibliographies[11]), and
- semi-structured WWW pages without search engine (Technical Report Server University Bonn[12], Journal of AI Research[13], Logic Journal of the IGPL[14], The Computer Journal[14], Journal of Logic and Computation[14]).

To implement the Z39.50 protocol we made use of an Application Programming Interface (API) used in a german library network project (DBV-OSI II). The queries are transformed in Typ-1 format with the Bib-1 attribute set. The query results are converted from USMARC into the internal warehouse format.

[10] This is an ongoing process, of course; we expect a fully-operational system to have many more information servers.

[11] `http://liinwww.ira.uka.de/bibliography/`

[12] `http://www.informatik.uni-bonn.de/III/forschung/publikationen/tr/`

[13] `http://www.cs.washington.edu/research/jair/`

[14] `http://www.oup.co.uk/igpl/` or `http://.../comjnl/` or `http://.../logcom/`

The communication with the WWW gateway is realized by constructing an appropriate HTTP-Request and by extracting the results from the corresponding HTTP pages. Due to lack of space we do not discuss these issues and the implementation of the query functionalities for the semi-structured WWW pages, but refer the reader to [26].

4.3 Layer Model

When bibliographic data are collected from internet repositories, they are transformed into the uniform MyView scheme. This scheme should not be mistaken for conventional database schemes. The structure of bibliographic data records is irregular [37]: Some may have an abstract, a reference to conference proceedings or a journal volume, other do not. The record structure is implicit: Single items must be identified in raw data as different as BibTeX and MARC. Bibliographic data records thus have features typical of *semi-structured data* as defined by Abiteboul [1].

Graphs and trees have proved to be suitable for representing semi-structured data (see [13]). A tree representation of data records is at the heart of the MyView system. Each data record is represented by a tree with labeled internal and external nodes. Labels of internal nodes denote record components (author, title, ...), whereas labels of external nodes carry the values (eg an author's name). In our (as yet) simplified model all values are of type string.

This conceptual data model links the physical data storage with the interface layer. The interface layer is responsible for the transmission of data from internet repositories. Data are transformed into their internal tree representation which is then passed to the physical layer for persistent storage. This layered architecture enables us to experiment with different storage mechanisms as SGML/XML files and the Lore DBMS.

Furthermore, the conceptual data model forms the basis for user queries. The user may search for documents which contain some given keywords. But he can also require these keywords to occur only in certain leaves of the tree which are specified by their ancestor nodes. We picked up this issue in Sec. 3.2.

4.4 Warehouse Management

So far, we have implemented only a few retrieval functionalities (see Sec. 3.2: Boolean queries, Lorel queries). The Boolean queries are translated into sgrep[15] commands operating on an SGML file representation of the warehouse data.

sgrep (structured grep) is a tool for searching text documents. The search is based on text regions, which can be defined by constant strings or start and end tags (like in SGML). An sgrep query consists of region expressions and can for example check, whether one region includes another region.

An sgrep command for a query which searches for documents containing the string "sgml" in the title, looks like this

[15] `http://www.cs.helsinki.fi/~jjaakkol/sgrep.html`

```
sgrep -i 'NAMED_ELEMS(record) containing (NAMED_ELEMS(title) containing
                                ((``sgml'')))'.
```

The option `-i` is for switching to case insensitive search and `NAMED_ELEMS(X)` is a macro for defining a region enclosed with start and end tags for `X`.

The Lorel queries are evaluated directly through Lore (the Lore API could be used instead). The underlying Database is constructed by transforming the warehouse format into the Object Exchange Model (OEM) used in Lore.

5 Related Work

In recent years many efforts have been made in digital library projects and Internet information retrieval tools to provide functionalities like search, storage, access, and organization. In the following we describe some of the proposed approaches and compare them with MyView.

Existing search engines (AltaVista[16], InfoSeek[17]) and resource discovery tools (see [11]) are impressively powerful what concerns the keyword-driven discovery of Internet resources. But they do not integrate the millions of document descriptions of traditional library catalogues. Web-based interfaces to libraries[18] on the other hand will in most cases support only simple queries and each offers a different user interface. In-between these two extremes meta search engines (MetaCrawler [35], SavvySearch[19]) and networked literature collections (NCSTRL[20]) overcome the latter interface diversity, but problems remain: predefined search space not configurable by the user, restricted retrieval capabilities.

The Harvest system [10] is an integrated set of customizable tools for gathering information from diverse Internet repositories and their subsequent effective use. The architecture enables the construction of topic-specific content indexes (broker), but the definition of a personalized view is not supported directly. As stated in [29], the original goal of having enough Harvest brokers for most purposes and leading the users by querying a central registry (Harvest Server Registry) to the right broker has never been reached. Furthermore some types of information repositories cannot be handled, such as traditional library catalogues.

The Search Broker[21] [29] is a search tool combining two search phases into one regular search. In a first phase the search is after the right database, whereas in a second phase the selected database is queried for relevant information. This idea grew out of the Harvest project. User input is a list of keywords with the first being a subject identifier followed by the actual query. When asking for just

[16] `http://www.altavista.com/`
[17] `http://www.infoseek.go.com/`
[18] `http://www.lights.com/webcats/`
[19] `http://www.savvysearch.com/`
[20] `http://www.ncstrl.org/`
[21] You can find the homepage at `http://sb.cs.arizona.edu/sb/`, but the system is no longer maintained.

one keyword the internal subject list is queried and information about it and all its related subjects is given for further usage. Usually, the response of a regular search is not modified and just appended to an introduction describing and referencing the source search engine. So, no further processing of the result sets is done leaving the work to the user. This is one of the differences to MyView. We are in the line with the Search Broker and many other meta search engines what concerns the usage of available search engines on the web. But our approach also integrates different information providers and explicitly stores and maintains the gathered information.

The TSIMMIS system [19] integrates data from multiple heterogeneous sources and provides users with seamless integrated views of the data. It translates a user query on an integrated view into a set of source queries and postprocessing steps that compute the answer to the user query from the results of the source queries. The explicit view definitions and the view expansions by the mediators are the precondition for query evaluation and as such the central key to the underlying information. An automated resource selection has not to be done in TSIMMIS at the expense of predefining the views. The MyView system pursues a different path. Instead of describing the properties of the information repositories extensively, it only needs some general metadata to connect to the information server and query their repositories. Naturally, the querying on the local warehouse has still to be done. The heterogeneity of the data sources in TSIMMIS is handled by using the semi-structured data model OEM (Object Exchange Model).

A system which aims at integrating distributed Internet resources and uses word-frequency information for their selection is *GlOSS* [23]. It focuses on the identification of relevant text databases for a given query and uses the word-frequencies to estimate the result sizes of the query. The hard problem of modeling a user's information need is not tackled in *GlOSS*. The generalized version *gGlOSS* [22] also deals with vector-space databases and queries, but at the expense of additionally required statistical information about the databases.

The *Stanford Proposal for Internet Metasearching* STARTS[22] tries to facilitate the three main tasks a metasearcher has to perform: the selection of the best source, the evaluation of the queries at these sources, and the result merging. The group effort of more than ten companies and organizations, coordinated by Stanford's Digital Libraries Project leads to a protocol definition for Internet retrieval and search [21]. Unfortunately, as far as we know, STARTS is only used in Stanford's own InfoBus [34] - a prototype infrastructure to extend the current Internet protocols. We would really appreciate the realization of the STARTS proposal, but we believe that the active support of information providers like libraries or publishing houses is the exception in real life applications.[23] That is the reason why we do not wait for the providers to do something, but describe the resources on our own in MyView.

[22] `http://www-db.stanford.edu/~gravano/starts_home.html`

[23] The involvement in such projects may be motivated by the hope of taking some advantage (image cultivation or in financial ways), when the taken approach reveals to be widely accepted.

Beside the Lore DBMS mentioned above other approaches have been made for combining structured documents with database technology. Böhm et al. [8] describe declarative and navigational access mechanisms in HyperStorM, building on a configurable database-internal representation of documents. Avoiding the parsing of a DTD to speed up operations may be worthwhile to consider in the future.

We strongly believe that our approach is an improvement and has the potential to be a significant step forward. A user can use the default settings and participate in the benefits of the system. Under the prerequisite of investing some time at the beginning for customization (adding servers that has a relevance to the personal information need or, as a first guess, selecting some of those already known to the system) the query results will be of even higher quality. This may not be very surprising, but a lot of systems do not even give the end users a chance in controlling its behaviour.

6 Conclusions and Future Work

The MyView project comprises a diversity of research issues in the area of digital libraries, networked information retrieval and internet information systems. For instance, the resource discovery problem, the collection fusion problem and the metadata discussion have to be considered. We know, that one system or even one model is not capable of solving all the problems, but we have shown how such an approach may look like and how we believe to realize some aspects of it. Our proposal combines fully automatic parts (query generation and submission) and manual parts (adding information providers, defining the information need) to support the user in time-consuming and monotonous tasks, but leave the responsibility to him in mission critical details.

In this paper we presented the present state of the MyView system, a warehouse for bibliographic data which is locally available for browsing, ad hoc queries, re-arrangements and analysis. The global architecture was sketched and the current implementation described. At the moment we are working mainly on the automated resource selection of the information repositories and investigate in suitable query languages and user interfaces for the warehouse exploration. Furthermore, we have to examine how we can take advantage of the recent XML developments.

We have discussed the MyView concept in the domain of searching for literature, but the principle design decisions and the architecture are of general interest for a number of other application domains. We strongly believe that the MyView approach is a worthwhile step in the right direction.

7 Acknowledgements

We are grateful to Jürgen Kalinski, Annette Langer, Jan Stohner and all the others involved in the MyView project for many fruitful discussions and the contributions they made.

References

1. Abiteboul, S. Querying semi-structured data. In *Proc. of the 6th Int. Conf. on Database Theory (ICDT)*, LNCS 1186, 1–18. Springer, 1997.
2. Abiteboul, S., S. Cluet, V. Christophides, T. Milo, G. Moerkotte, and J. Siméon. Querying documents in object databases. *Int. J. on Digit. Libr.*, 1(1):5–19, 1997.
3. Abiteboul, S., D. Quass, J. McHugh, J. Widom, and J. Wiener. The Lorel Query Language for Semistructured Data. *Int. J. on Digit. Libr.*, 1(1):68–88, 1997.
4. Atzeni, P., G. Mecca, and P. Merialdo. Semistructured and structured data in the web: Going back and forth. *SIGMOD Record*, 26(4):16–23, 1997.
5. Atzeni, P., G. Mecca, and P. Merialdo. To weave the web. In *Proc. of the 23th Int. Conference on Very Large Data Bases (VLDB)*, 206–215. 1997.
6. Barnard, D. and N. Ide. The Text Encoding Initiative: Flexible and Extensible Document Encoding. *J. of the Am. Soc. for Information Sci.*, 48(7):622–628, 1997.
7. Belkin, N. J. and B. W. Croft. Retrieval techniques. *Annual Review of Information Science and Technology*, 22:109–145, 1987.
8. Böhm, K., K. Aberer, E. Neuhold, and X. Yang. Structured Document Storage and Refined Declarative and Navigational Access Mechanisms in HyperStorM. *VLDB Journal*, 6(4):296–311, 1997.
9. Bonhomme, S. and C. Roisin. Interactively Restructuring HTML Documents. In *Proc. of the 5th Int. WWW Conf.*, 1996.
10. Bowman, M., P. Danzig, D. Hardy, U. Manber, and M. Schwartz. The Harvest Information Discovery and Access System. In *Proc. of the 2nd Int. WWW Conf.*, 763–771, 1994.
11. Bowman, M., P. Danzig, U. Manber, and M. Schwartz. Scalable Internet Resource Discovery: Research Problems and Approaches. *CACM*, 37(8):98–107, 1994.
12. Bray, T. and R. V. Guha. A MCF Tutorial, 1997.
13. Buneman, P. Semistructured Data. In *Proc. of the 16th ACM Symp. on Principles of Database Systems (PODS)*, 117–121. 1997.
14. Chaudhuri, S. and U. Dayal. An Overview of Data Warehousing and OLAP Technology. *SIGMOD Record*, 26(1):65–74, 1997.
15. Delsey, T. The Evolution of MARC Formats. In *The Future of Communication Formats*, International Conference, Canada, 1996.
16. Desai, B. C. Supporting Discovery in Virtual Libraries. *Journal of the American Society for Information Science*, 48(3):190–204, 1997.
17. Dogac, A., M. T. Ozsu, and O. Ulusoy, eds. *Current Trends in Data Management Technology*. Idea Group Publishing, Hershey, USA, 1999.
18. Fox, E. A. and G. Marchionini. Toward a worldwide Digital Library. *CACM*, 41(4):29–32, 1998.
19. García-Molina, H., Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, V. Vassalos, and J. Widom. The TSIMMIS approach to mediation: Data models and Languages. *J. of Intelligent Information Systems*, 8(2):117–132, 1997.
20. Goossens, M., F. Mittelbach, and A. Samarin. *The LaTeXCompanion*. 1994.
21. Gravano, L., C.-C. K. Chang, H. García-Molina, and A. Paepcke. STARTS: Stanford Proposal for Internet Meta-Searching. In *Proc. of the 1997 ACM SIGMOD Int. Conference on Management of Data*, 207–218. 1997.
22. Gravano, L. and H. García-Molina. Generalizing GlOSS to vector-space databases and broker hierarchies. In *Proc. of the 21th Int. Conf. on Very Large Data Bases (VLDB)*, 78–89. 1995.

23. Gravano, L., H. García-Molina, and A. Tomasic. The efficacy of GlOSS for the text database discovery problem. Tech. Rep. STAN-CS-TN-93-2, Stanford University, 1993.
24. Hammer, J., H. García-Molina, J. Cho, R. Aranha, and A. Crespo. Extracting semistructured information from the web. In *Workshop on the Management of Semistructured Data*, 1997.
25. Document Style Semantics and Specification Language (DSSSL), 1996.
26. Langer, A. *Extraktion von halbstrukturierten Daten im personalisierten Literaturkatalog* MyView. Master's thesis, Inst. f. Inf. III, Univ. Bonn, 1998. in German.
27. Lasher, R. and D. Cohen. A Format for Bibliographic Records. Request for Comment (RFC) 1807, 1995.
28. Lowe, B., J. Zobel, and R. Sacks-Davis. A formal model for databases of structured text. In *Proc. of the 4th Int. Conf. on Database Systems for Advanced Applications*, vol. 5, 449–456. 1995.
29. Manber, U. and P. A. Bigot. The Search Broker. In *First Usenix Symp. on Internet Technologies and Systems, Monterey, CA*. 1997.
30. McHugh, J., S. Abiteboul, R. Goldman, D. Quass, and J. Widom. Lore: A database management system for semistructured data. *SIGMOD Record*, 26(3):54–66, 1997.
31. Miller, E. An introduction to the resource description framework. *d-lib Mag*, 1998.
32. National Information Standards Organization. Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. NISO Press, 1995.
33. Papakonstantinou, Y., A. Gupta, and L. Haas. Capabilities-based query rewriting in mediator systems. In *Proc. of the Int. Conf. on Parallel and Distributed Information Systems*. 1996.
34. Röscheisen, M., M. Baldonado, C. Chang, L. Gravano, S. Ketchpel, and A. Paepcke. The Stanford InfoBus and Its Service Layers: Augmenting the Internet with Higher-Level Information Management Protocols. In *Digital Libraries in Computer Science: The MeDoc Approach*, LNCS 1392, 213–230. Springer, 1998.
35. Selberg, E. and O. Etzioni. Multi-Engine Search and Comparison using the MetaCrawler. In *Proc. of the 4th Int. WWW Conf.*, 195–208. 1995.
36. Stohner, J. *Sammlung von Metainformationen im personalisierten Literaturkatalog* MyView. Master's thesis, Inst. f. Inf. III, Univ. Bonn, 1998. in German.
37. Suciu, D. Semistructured Data and XML. In *Proc. of the Int. Conf. on Foundations of Data Organization*. 1998.
38. Watters, C. and M. A. Shepherd. Shifting the information paradigm from data-centered to user-centered. *Information Processing & Management*, 30(4):455–471, 1994.
39. Weibel, S. and J. Hakala. DC-5: The Helsinki Metadata Workshop. *D-Lib Magazine*, 1998.
40. Widom, J. Research Problems in Data Warehousing. In *Proc. of the 4th Int. Conf. on Information and Knowledge Management*, 25–30. 1995.
41. Wiederhold, G. Mediators in the architecture of future information systems. *IEEE Computer*, 38–49, 1992.
42. Wolff, J. and J. Kalinski. The MyView System: Tackling the Interface Problem. Tech. Rep. IAI-TR-97-5, Institut für Informatik III, Universität Bonn, 1997.
43. Wolff, J. and J. Kalinski. Mining Library Catalogues: Best-Match Retrieval based on Exact-Match Interfaces. In *Proc. of the Int. Workshop on Issues and Applications of Database Technology (IADT'98)*. 1998. Also appeared in [17].