Using Hypertext Composites in Structured Query and Search

Zhanzi Qiu, Matthias Hemmje, Erich J. Neuhold

GMD-IPSI, Darmstadt, Germany
{qiu, hemmje, neuhold}@darmstadt.gmd.de

Abstract. This paper describes a part of our effort to answer the open question how to use the structural and semantic information that is representable with the new Web standards efficiently for searching the Internet and filtering and retrieving relevant information. Link-based hypertext composites are applied in formulating structured queries and deriving structured search results. By enabling users to query different levels of the composite structures with same or different keywords and getting search hits that are not separate nodes but sets of inter-linked nodes, the precision of the search results can be improved. In addition, users may get more contextual information about the search results. This paper presents this idea and describes a prototype system that implements the idea.

1 Introduction

For several years the World Wide Web Consortium (W3C) [30] has been working hard to create and promote base technologies for enabling the "semantic Web" [2]. Among their efforts, the most important twos are XML (Extensible Markup Language) [28] and RDF (Resource Description Framework) [23]. XML is the universal format for structured documents and data on the Web, while RDF is a "metadata" framework that allows semantic relationships to be expressed in structures that can be read and processed by computer programs. These two standards improve the ability of expressing structures and semantics on the Web.

Accompanying W3C's effort, much research work is being done with respects to the models, architectures, and management aspects of the future semantic Web. Especially, it is an open question how to use the structural and semantic information that is representable with the new Web standards efficiently for searching the Internet and filtering and retrieving relevant information. To answer this question, we have proposed a schema for searching in the Web space by using hypertext contexts as a mechanism to specify the scope of information space to be examined [20; 21]. We also have presented an idea of making use of link-based domain models, which are hyperstructures that have domain specific semantics, in formulating structured queries so that Web users can get more specific results relevant to their information needs [22].

This paper describes another part of our effort to answer the open question. In this work, we focus on making use of hypertext composites, especially link-based hypertext composites, to formulate structured queries and derive structured search results. We argue that by enabling users to query different levels of the structures with same or different keywords and getting search hits that are not separate nodes but sets of inter-linked nodes, the precision of the search results can be improved. Furthermore, users can get more contextual information about the search results.

The remainder of the paper is organized as follows. Section 2 describes the concept of hypertext composites. Section 3 answers the question how hypertext composites can be represented with new Web standards. Section 4 proposes a schema for using hypertext composites in Web searching. Section 5 presents a prototype system designed for implementing the schema. Section 6 mentions related work. Finally, Section 7 summarizes this work and outlines our future activity.

2 Hypertext Composites

2.1 Overview

The logical structures in hypertext are usually supported by means of composites (a group of nodes and links) [Halasz and Schwartz 1994]. Hypertext composites have been used in a few hypermedia models and systems, such as Dexter Model [14; 15], HyperBase System [25], SEPIA System [27], HDM [9].

There are two kinds of hypertext composites: composite nodes that are composed by non-linking mechanisms (such as hierarchical structures in the structured documents defined by a DTD); and link-based composites that are composed by computation based on link types that represent containment (or part-of) relations.

Although in most cases, especially in a single system, non-linking composition mechanisms are more efficient than link-based composition mechanisms [4], there are also many cases where link-based constructs are desired [13]. Compared to non-linking mechanisms, link-based composites have many distinct properties. They can be constructed on the fly, and so they are more dynamic. They offer users to see and trace not only outgoing links from a node, but also incoming links to a node. They provide descriptive link types and names to help users in their navigation. They lend themselves to reuse existing links, nodes, and contents to create multiple overlapping views (and paths) upon the same set of information objects. They allow users to create links from documents to which they have no write-permission or from documents that have no public link embedding interface. These properties are valuable in augmenting the embedded link (jump address) based hypertext in the current WWW.

This paper is to explore the value of link-based hypertext composites in Web searching. Thus unless clearly specified, composites later in this paper refer to link-based composites. A more formal description about such composites is first given below.

2.2 Formal Description of Link-based Hypertext Composites

In a hypermedia system, a *node* provides the hypermedia “wrapping” for a document or piece of information. A *link* represents a relation between nodes. A *link-based hypertext composite* is a special kind of node that is constructed out of other nodes and composites [14; 15]. These nodes and composites are *components* of the composite. They are linked from the composite or the other components in the composite with containment (or part-of) relation. They may also link to each other with other types of relation. No link-based hypertext composite may contain itself either directly or indirectly.

Precisely, if C is a *link-based hypertext composite*, then its contents must contain a pair (N, L), where N is a set of nodes in a hypertext graph and L is a set of semantic links whose endpoints belong to N. For any $n_1 \in$ N, there exists *link* (C-> n_1) $\in$ L and *link* (C-> n_1)*.type* represents containment relations, or, there exists $n_2 \in$ N so that

link (n_1-> n_2) $\in$ L *AND* *link* (n_1-> n_2)*.type* represents containment relations,

OR

link (n_1-> n_2) $\in$ L AND *link* (n_2-> n_1)*.type* represents containment relations.

We say that C contains a node *M* if *M* is in N and that C contains a link *l* if *l* is in L. *M* is a *node component* of C, while *l* is a *link component* of C.

Usually the *components* of a *hypertext composite* refer to its *node components*, as the meaning of the link components is mostly reflect in the building-up process of the composite. Unless clearly specified, this usage of *components* is adopted later in this paper.

2.3 Hypertext Composites and Hypertext Contexts

Based on the above definition of hypertext composites, a hypertext composite can be seen as a special kind of hypertext contexts. A *hypertext context* is a generic high-level hypermedia structure that groups together a set of nodes and links into a logical whole [20]. Furthermore, The nodes in a hypertext composite may be collected in various hypertext contexts. That is, hypertext contexts may be used to describe different views of a hypertext composite.

2.4 A Simple Example

One main use of a link-based hypertext composite is to organize a document in a

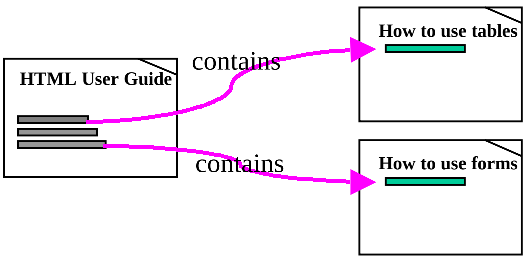

Figure 1 A simple composite "HTML User Guide"

hierarchical structure. A typical example is an on-line user manual, which has hierarchical structure as its backbone, but also has other hypertext links within or across the document boundary.

Figure 1 shows a simple hypertext composite “HTML User Guide” which contains the nodes “How to use tables”, “How to use forms” and so on as its components. These nodes can be grouped into various hypertext contexts that represent the different views of the document that is organized hierarchically with the composite.

3 Representation of Hypertext Composites

From the first beginning the WWW has only a single node type called the page. Something like the effect of composites can be obtained using pages full of URLs. However, true structuring composites are not supported. That is, pages can not be nested. All pages are in fact equally accessible in a “flat” pool [13].

This situation has been changed because of the development of new Web standards. W3C's recommendation for the latest version of HTML, HTML 4.01 [16], has defined a kind of LINK element which may only appear in the HEAD section of a document to describe relations between documents and a set of link types permitted in the documents. Moreover, it allows users define additional link types, by using a meta data profile to cite the conventions used to define the link types (The *Profile* is an attribute of the HEAD element). This makes it possible for an application system to compute link-based composites based on the link types.

The LINK element defined in HTML 4.01 contains a *rel* attribute to specify the relationship of the linked document with the current document and a *rev* to describe a reverse link from the linked document to the current document. The value of the both attributes is a space-separated list of link types. The predefined link types that relate to the containment relation between Web documents and can be used to compute hypertext composites are:

- *Contents* - refers to a document serving as a table of contents
- *Chapter* - refers to a document serving as a chapter in a collection of documents
- *Section* - refers to a document serving as a section in a collection of documents
- *Subsection* - refers to a document serving as a subsection in a collection of documents

A collection of documents can be organized in a composite. For instance, in the Figure 1, suppose the document for “How to use tables” is “table.html” and the document for “How to use forms” is “form.html”. They both are chapters of a document collection for “HTML User Guide”, whose table of contents is in “HTML.html”. Then ”HTML.html” may contain the following exemplary encoding for describing the containment relation:

```
<HEAD>
 ...other head information...
 <TITLE>HTML User Guide</TITLE>
 <LINK rel="contents" href="table.html">
```

```
<LINK rel="contents" href="form.html">
</HEAD>
```

Or, “table.html” and “form.html” may contains:

```
<HEAD>
 ...other head information...
<LINK rel="chapter" href="HTML.html">
</HEAD>
```

It is apparent that based on this kind of link information within HTML documents, an application system may compute and build up link-based composites.

With respect to XML documents, XLink [Xlink] provides an efficient mechanism to represent typed links. In XLink, links are encoded in linking elements. The types of links can be encoded via the *role* attribute of linking elements. The values of this attribute are of a kind of CDATA. They may be predefined in DTDs (fixed) or specified in documents (no default value is provided in DTDs). For instance, suppose in the Figure 1 the XML document for “HTML User Guide” is “HTML.xml”, the document for “How to use tables” is “table.xml” and the document for “How to uses forms” is “form.xml”. The following out-of-link extended xlink can be contained in “HTML.xml” to describe the simple composite shown in the figure:

```
<content xml:link=”extended” inline=”false”>
 <locator href=”table.xml” role=”contains”>
 <locator href=”form.xml” role=”contains”>
</content>
```

Still, RDF [23] provides a more systematic way to describe the composites, whose components may be either HTML documents or XML documents or any other kinds of Web resources. For instance, the following sample RDF encoding uses Dublin Core [8] vocabularies to describe the simple composite shown in Figure 1 (suppose the documents are in HTML format):

```
<rdf:RDF xmlns:rdf="http://www.w3.org/TR/WD-rdf-syntax#"
        xmlns:dc="http://purl.org/metadata/dublin_core#"
        xmlns:dcq="http://purl.org/metadata/dublin_core_qualifiers#">
   <rdf:Description about="HTML.html">
        <dc:Relation>
            <dcq:RelationType
 rdf:resource="http://purl.org/metadata/dublin_core_qualifiers#hasPart"/>
            <rdf:value resource="table.html”/>
            <rdf:value resource="form.html”/>
        </dc:Relation>
   </rdf:Description>
</rdf:RDF>
```

As the encoding shows, the link type “hasPart” is represented by the tag <dc:Relation> and the rdf:resource attribute in <dcq:RelationType>. In fact, Dublin

Core has defined two qualifiers for containment relations: "has Part" and "is Part of". They provide an ideal way to represent hypertext composites.

4 Using Hypertext Composites in Structured Query and Search

With the possibility of representing hypertext composites or the link types necessary for building up the composites with the new Web standards, it is time for us to explore new search methods making use of the composites for the Web. As shown in Figure 2, based on composite structures, hyperstructure-based query and search facilities can be implemented. These facilities will enable users to query different levels of the structures with the same or different keywords (see the left-hand side part of Figure 2) and get search hits that are not separate nodes but sets of inter-linked nodes (see the right-hand side parts of Figure 2). Compared to the search results that are single nodes, the structured search results may be more precise and relevant to users' some specific information needs. Furthermore, users are provided with more contextual information about the nodes contained in the results.

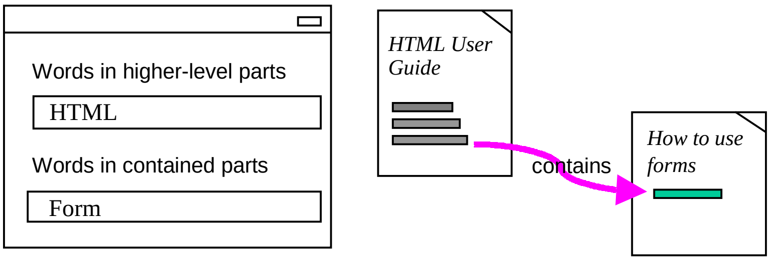

Figure 2 Structured query and search using link-based composite structure

4.1 Structured Queries Based on Hypertext Composites

Formally, the general structured queries based on hypertext composites can be described as follows:

Definition 1 *Structured Query Levels*

Structured query levels SQLs is a value that indicates how many structural levels are to be contained in a structured query.

Definition 2 *Query Terms*

A query term is the content descriptor used to search for nodes indexed in a document collection. It may be a keyword, a phrase, or a Boolean expression.

$QT = \{QT_i\}$, where $1 \le i \le N_{qt}$, N_{qt} is the number of query terms

Example: QT = {< HTML user guides>, <digital library>, <information retrieval>}

Definition 3 *Query Fields*

A query field corresponds to a structural level that is greater than 1 but less than the structured query levels (SQLs). It limits the scope of a query term, i.e. it requires that the provided term has to be contained in the indexing result of nodes at the level with probably some special constraints.

$QF = \{QF_i\}$, where $1 \le i \le SQLs$

Example: QF={<QF_1>, < QF_2>, < QF_3>)

Definition 4 *Qualifiers*

A qualifier is used to describe the quality and form of the query terms that users input.

$Q = \{Q_i\}$, where $1 \le i \le N_q$, N_q is the number of qualifiers

Example: Q={<=>, <*?>}, here "=" means exactly like, "*?" means using stemming expressions

Definition 5 *Field Query Expressions*

A field query expression $\overline{FQ_i}$ is constructed by a query field QF_j with a query term QT_k combined by a qualifier Q_l. That is:

$\overline{FQ_i} = (QF_j \; Q_l \; \overline{QT_k})$, where $QF_j \in QF$, $QT_k \in QT$, $Q_l \in Q$

Example: $\overline{FQ_1}$ **=** (QF_1= "informational retrieval")

$Field(\overline{FQ_i})$ is used to denote the query field used in the query expression $\overline{FQ_i}$, i.e., $Field(\overline{FQ_i}) = QF_i$.

Definition 6 Structured Query Expressions

A structured query expression is a conjunction of some field query expressions.

$\overline{SQ} = \wedge(\overline{FQ_1}, \overline{FQ_2}, \overline{FQ_3}, \ldots, \overline{FQ_n})$,

where n = SQLs, i.e. the structured query levels, "$\wedge$" is logical operator AND

Example: $\overline{SQ} = \wedge$(QF_1="User guide", QF_2="HTML", QF_3="form"), which searches for a three-layer hyperdocuments with "User guide" in the first level, "HTML" in the second level, "form" in the third level.

4.2 Structured Search Results Based on Hypertext Composites

As mentioned, the structured search hits resulted from the structured queries are not separate nodes but sets of inter-linked nodes. That is, each search hit (as shown in the right-hand side part of Figure 2) is itself a hierarchical composite, in which the containment type of links exist between the component nodes.

Formally, the search results for a structured query $\overline{SQ}$ can be described with the following definitions:

Definition 7 *Field Query Results*

The field query results for a field query expression ($\overline{FQ}$) are a set of nodes that "contain" the query term (with the qualifier as quality control) specified in FQ.

$$\mathrm{FQR}(\overline{FQ}) = \mathrm{FQR}(QF\ Q\ QT) = \{N_i\}$$

where N_1 "contains" (with Q as qualifier) QT

Definition 8 *inter-containment-linked node chains*

An inter-containment-linked node chain is a series of nodes, each of which (except the first one) is linked from the node before it with a type that represent containment relations and (except the last one) also links to the node after it with a type that represent containment relations.

$$\mathrm{CILNC} = \{CILNC_j\}$$

$$CILNC_j = \{N_1(\text{contains})\text{->}N_2(\text{contains})\text{->}\ \dots\ \text{->}N_n\},$$

where n = the number of the nodes in the chain

Definition 9 *Structured Search Results*

The structured search results for a structured query expression ($\overline{SQ}$) are a set of inter-containment-linked node chains, which are derived by computing the containment links between the nodes contained in the field query results corresponding to the field query expressions in $\overline{SQ}$.

$$\mathrm{SSR}\ (\overline{SQ}) = \mathrm{SSR}(\wedge(\overline{\mathrm{FQ}_1}, \overline{\mathrm{FQ}_2}, \overline{\mathrm{FQ}_3}, \dots, \overline{\mathrm{FQ}_n})) = \{CILNC_n\}$$

$$CILNC_j = \{N_1(\text{contains})\text{->}N_2(\text{contains})\text{->}\ \dots\ \text{->}N_n\},$$

where $N_l \in \mathrm{FQR}(FQ_l)$, n = SQLs, i.e. the required structured query level

4.3 Issues for Supporting Structured Query and Search Based on Hypertext Composites

To enable the use of link-based hypertext composites in structured queries and searches, a few issues are to be addressed. The first is to represent hypertext composites in a standard way and make them sharable and reusable throughout the Web. This issue has been discussed in Section 3 above in this paper and can be seen as the prerequisite for addressing other issues.

With this prerequisite, a search system which intends to enable the use of hypertext composites in formulating structured queries and deriving structured search results should be able to

- gather from the Web available hypertext composite information by computing the links between the nodes (pages) and organize the information in the system efficiently,
- provide a friendly, adaptive interface to enable users to specify structured queries in a comfortable way,
- derive the structured search results with acceptable system performance, and
- present structured search results in a way that is good for users to understand and get more contextual information about the results.

In the next section we will see how our prototype system addresses these issues.

5 A Prototype System

The prototype system is designed for testing our idea of using hypertext composites in structured query and search and studying the feasibility of the structured query and search model proposed above. In the following we first give its high-level system architecture and then introduce several technologies in its implementation. Finally, we will discuss its application domains and evaluation issues.

5.1 Architecture Overview

As shown in Figure 3, the system contains the following components that represent different aspects and are mostly layered:

The **information gathering and hypertext composite detection component** is responsible for gathering Web resources (HTML, XML, RDF docs) based on users' server selection and detecting (extracting) hypertext composite information (node and link components of the composites) from the Web resources gathered and stores the information in databases. The **indexing component** does keyword indexing to the Web resources gathered. These two components together make up the **info agent** of the system.

The **query component** enables specifying structured queries, transfers the queries to the retrieval component, and presents search results derived by the retrieval component to users. This component will be implemented as Web browser clients with a form-based user interface.

The **retrieval component** is responsible for using data in the database to derive search results and sending search results to the query component.

The **DB manager** is a backbone of the entire system. It receives data from the info agent, and provides data to the retrieval component and the info agent. It can be an object relational DBMS, e.g. Informix Universal Server.

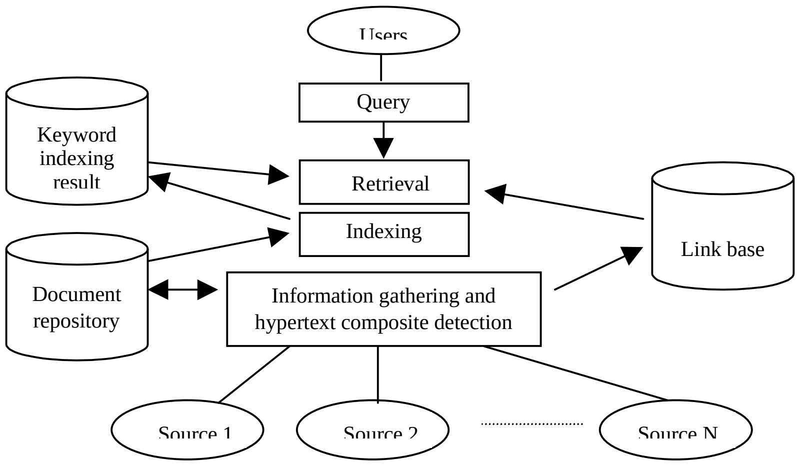

Figure 3 High-level system architecture

5.2 Information Gathering and Hypertext Composite Detection

To support the structured query and search based on hypertext composites in a collection, to gather the hypertext composite information in the collection is the basis. As described before in this paper, a link-based hypertext composite is constructed with links of the types that represent containment relations as its bones. Thus, the aim to gather hypertext composite information can be attained by gathering the link information, which contains where links from and to, as well as the types of the links.

Indicated by its name, the information gathering and hypertext composite detection component in the system performs mostly 2 functions: Web crawling and hypertext composite detection. Web crawling is to download the Web resources (whose URIs are given by a URI server) specified by users or parsed out from the resources during the processing and stores them in the document repository. Hypertext composite detection is to read the repository and parse the documents in it. Every web resource gets an associated ID number called a nodeID that is assigned whenever a new URI is parsed out of a web resource. All link information is extracted and stored (through a

Node Table	nodeID	URI		
Link Type	linktypeID	linktype		
Link Table	linkID	source_nodeID	linktypeID	target_nodeID

Figure 4 Database for hypertext composites – a link base

store server) in the link base of the system. The database schema for the link base is shown in Figure 4.

The node table is for storing the URIs of the nodes (Web resources) gathered in the collection. The link type table is for storing the link types existing in the collection. The link table contains the links between the nodes in the collection. With the link information in the link base, any composites in the collection can be computed dynamically when necessary and be used to derive structured search results that meet users' structured queries.

The crawler in the system is written in PERL and the extractor for hypertext composite detection is JEDI (Java Extraction and Dissemination of Information) [17] tool. The JEDI tool consists of a wrapper that can collect information by navigating through multiple documents and by explicating their implicit logical structure, and a mediator that maps the collected information to an integrated view. It meets the demand of extracting hypertext composite information from heterogeneous textual information sources that contain the composite information in the system. The heterogeneity is resulted from the different ways to represent hypertext composites, as described in Section 3.

5.3 Indexing and Retrieval

Indexing component in the system is to do keyword indexing to the Web resources (HTML and XML docs) gathered. The keyword indexing technology is so mature that we do not need to give much detail about it. As an example, our prototype system simply makes use of Glimpse [10] as its keyword indexing and search engine.

Then the structured search results are derived in this way: the retrieval component of the system first sends query term in each structured query level to Glimpse and gets distinct responding results, and then uses the link information stored in the link base as filters to get structured search results.

For instance, the process to derive the structured search results for the exemplary structured query with 3 levels

$$\overline{SQ} = \wedge(\overline{QF_1}=\text{"User guide"}, \overline{QF_2}=\text{"HTML"}, \overline{QF_3}=\text{"form"})$$

is as shown in Figure 5.

In the figure, the *result set 1* is a set of documents that contain “User guide”. The *result set 2* is a set of documents that contain “HTML”. The *result set 3* is a set of documents that contain “form”. The *link_pairs 1* is set of link pairs. In each link pair the “from” node belongs to the result set 1, the “to” node belongs to the result set 2. Finally, the *final search results* are a set of link chains. In each chain, the first node belongs to the result set 1, i.e. contains the term “User guide”. The second belongs to the result set 2, i.e. contains the term “HTML”. The third belongs to the result set 3, i.e. contains the term “form”. The link pairs or link chains are constructed based on the links of the types that represent containment relations in the link base.

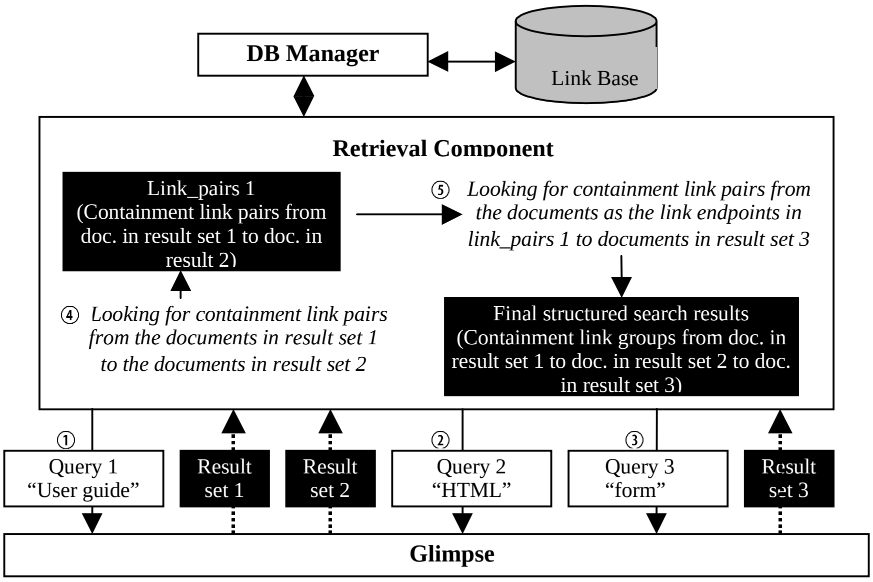

Figure 5 Deriving structured search results (an example)

5.4 Adaptive Form-Based Interface for Formulating Structured Queries

To enable users to formulate structured queries that reflect their specific information needs based on hypertext composites, the system provides an adaptive form-based interface. The adaptivity is mainly reflected in the adjustability of structured query levels. That is, if the user first select 2 levels but get unsatisfied results, he/she may ask the system to adjust the structured query levels to 3 or more. Every time when the value of the structured query levels is modified, the system will regenerate the form. An examplary query in such an interface is shown in Figure 6.

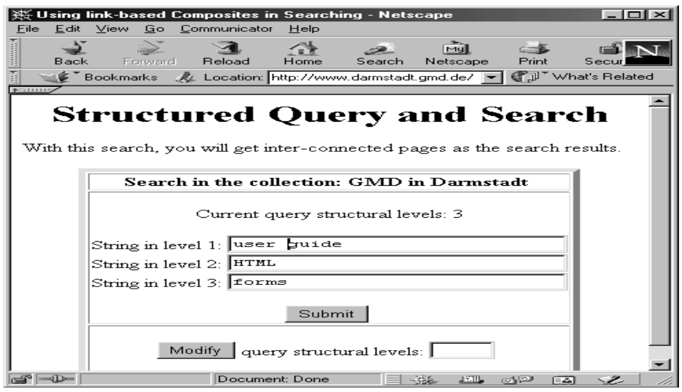

Figure 6 Form-based interface for formulating structured queries

5.5 Presenting Structured Search Results

As described, the structured search results derived from the structured queries based on hypertext composites are not separate nodes but sets of inter-linked nodes. How to present them to users is also crucial for the system's success. A good presentation may improve users' satisfaction to the results and enable users get more contextual information about the results and even the relevant resources.

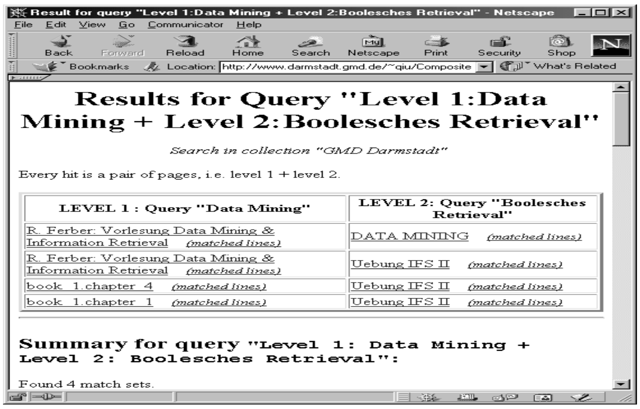

Figure 7 Presenting structured search results

In the moment the prototype system just presents the structured search results in a simple but integrated way. A table, which contains the designated structured query levels as the number of columns, is built to contain the results. Each row represents one result. A screenshot for an exemplary result presentation is given in Figure 7.

5.6 Application Domains and Evaluation Issues

Apparently the great value of the structured query and search method proposed in this paper for making use of hypertext composites will mostly be reflected in the application domains that own rich hypertext composite information and the information is represented in the standard ways. Any information service providers, either digital libraries or E-commerce systems or any small or large sites that provide well-organized hyperdocumens may benefit from supporting this search method. For instance, in our primary experiments with the system, we aim to provide users more specific results when they query about the technical documents, online user manuals and teaching materials and other kinds of well-organized hyperdocuments in the GMD Darmstadt site (*http://www.darmstadt.gmd.de/*).

A thorough evaluation about the system should not only measure the quality of its search results but also cover its storage requirements, its performance in extracting composite information, indexing and deriving structured search results. We have not done such a thorough evaluation. However, it is quite certain that the system will provide users more precise search results that meet users' some specific information needs and more contextual information about the results in the whole collection than normal search systems that provide users single pages as search results. There is also some significance that the system can scale well to the size of the Web, as it chooses a scalable DBMS and stores all link information gathered for computing the hypertext composites in databases.

6 Related Work

There is a trend of making use of additional structural information to improve Web searching. Structural information (mainly links) has so far been used for enhancing relevance judgements [1; 11; 19; 31; 24; etc.], ranking Web pages [5; 18; 3; etc.] or other purposes [e.g. 26]. Among the work in this area the achievements of Google [3; 12] and Clever [7; 18] are most attractive. Both systems use weighted link popularity as a primary criteria in their ranking mechanism. As far as we know, few systems have taken into account the Web's new abilities in expressing structural and semantic information in their search algorithms yet.

XML [28] and RDF [23] are two basic technologies from W3C [30] to improve Web's ability in expressing structure and semantics. It is still an open question how to use the structural and semantic information that is representable with these new Web standards efficiently for searching the Internet and filtering and retrieving relevant information. To answer this question, we have proposed a schema for searching in the Web space by using hypertext contexts as a mechanism to specify the scope of information space to be examined [20; 21]. This schema is also applied to hypertext composites, as hypertext composites can be seen as a special kind of hypertext contexts. We also have presented an idea of making use of link-based domain models, which are hyperstructures that have domain specific semantics, in formulating structured queries [22].

This work is a part of our most recent effort to answer the question. The structured query and search model that we propose is for making use of hypertext composites, i.e. the composites composed by computation based on link types that represent containment (or part-of) relations. In comparison, the XML Query [29] working group in W3C is devoting to provide flexible query facilities to extract data from XML documents, in which the composite nodes are composed by non-linking mechanisms.

Traditionally, structured search tools fall into the domain of database technology. A very good introduction to database technologies is [6]. The prototype system in this work also takes advantage of database technologies to store the link information in hypertext composites so that the structured search based on the composites can be implemented. However, with respect to the structures used in search and the form of the search result, the structured search we mean in this paper is different from the general structured search in the database community.

7 Summary and Future Work

This work proposes a structured query and search model for applying link-based hypertext composites that can be represented with new Web standards in Web searching and describes a prototype system that implements the model. We argue that by enabling users to query different levels of the composite structures with same or different keywords and getting search hits that are not separate nodes but sets of inter-linked nodes, the precision of the search results can be improved. In addition, users may get more contextual information about the search results.

The prototype system can gather hypertext composite information from the Web based on users' server selection, store the information in databases and then enable users formulate structured queries with an adaptive form-based interface, derive structured search results and finally present the results to users in a simple but integrated way. Being still quite primitive, it is just for demonstrating and testing our idea.

The follow-up work will be to perform a thorough evaluation about the method and the system as more hypertext composite information is provided on the Web and make further improvements to the method and the system. A more visualized interface for formulating structured queries and presenting structured search results will be provided in the system.

It will also be done to explore the value of the composite information in page ranking, filtering and the other activities related to Web searching. For instance, the appearance of a page at a higher level in the structures would be ranked higher when providing single pages as search results in a system.

Reference:

1. G. O. Arocena, A. O. Mendelzon, G. A. Mihaila, "Applications of a Web quary language," *Proc. 6th International World Wide Web Conference*, 1997.
2. Tim Berners-Lee, "What the semantic web can represent," available at: http://www.w3.org/DesignIssues/RDFnot.html.
3. S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web-search engine," *Proc. 7th International World Wide Web Conference*, 1998.
4. Carr, L., Hill, G., Roure, D. D., Hall, W., and Davis, H. "Open information services," *Proc. 5th International World Wide Web Conference*, 1996.
5. J. Carriere, R. Kazman, "WebQuery: Searching and visualizing the Web through connectivity," *Proc. 6th International World Wide Web Conference*, 1997.
6. Chris, J. Date. An Introduction to Database Systems. The Systems Programming Series. Addison-Wesley, Reading, Massachusetts, sixth edition, 1995.
7. Clever: http://www.almaden.ibm.com/cs/k53/clever.html
8. Dublin Core: http://purl.oclc.org/dc/
9. Garzotto, F., Paolini, P., "HDM – A model for the design of hypertext applications," *Proc. 3rd ACM Conference on Hypertext, December 1991*, pp. 313-327.
10. Glimpse: http://glimpse.cs.arizona.edu/
11. G. Golovchinsky, "What the query told the link: the integration of Hypertext and Information Retrieval," *Proc. 8th ACM Conference on Hypertext*, pp. 67-74, 1997.
12. Google: http://google.stanford.edu/
13. GRONBAK, K. and TRIGG, R., "Towards a Dexter-based model for open hypermedia: Unifying embedded references and link objects," *Proc. ACM Hypertext'96*, pp.16-20, 1996.
14. Halasz, F. and Schwartz, M., "The Dexter Hypertext Reference Model," *Proceedings of the Hypertext Standardization Workshop*, pp. 95-134, U.S. Government Printing Office, Washington, DC, 1990.

15. Halasz, F. and Schwartz, M., “The Dexter Hypertext Reference Model,” *Communications of the ACM*, 37(2), pp.30-39, February 1994.
16. HTML 4.01 Specification (W3C Recommendation 24 December 1999). Available at: http://www.w3.org/TR/html4/.
17. Huck, G., Fankhauser, P., Aberer, K., Neuhold, E., “Jedi: Extracting and synthesizing information from the Web,” *Proc. 3rd IFCIS International Conference on Cooperative Information Systems (CoopIS'98)*, New York City, August 1998, pp. 32-43.
18. J. Kleinberg, “Authoritative sources in a hyperlinked environment,” *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 1998.
19. M. Marchiori, “The quest for correct information on the Web: Hyper search engines,” *Proc. 6th International World Wide Web Conference, 1997*.
20. Qiu, Z., Hemmje, M., Neuhold, E. J., "ConSearch: Using hypertext contexts as Web search boundaries," in: Etzion, Opher & Scheurmann, Peter (Eds.), *International Conference on Cooperative Information Systems (CoopIS 2000)*, (pp. 42-53). Berlin [etc.]: Springer, 2000 (Lecture notes in computer science; 1901).
21. Qiu, Z., Hemmje, M., Neuhold, E. J., "Towards supporting user-defined hypertext contexts in Web searching," *Proc. IEEE Advances in Digital Libraries 2000* (pp. 48-57). Los Alamitos, CA, IEEE Computer Society.
22. Qiu, Z., Hemmje, M., Neuhold, E. J., “Using Link-Based Domain Models in Web Searching,” *Proc. 2000 KYOTO International Conference on Digital Libraries: Research and Practice*, November 13-16, 2000, Kyoto University, Kyoto, Japan.
23. RDF: http://www.w3.org/RDF/
24. E. Rivlin, R. Botafogo and B. Shneiderman, “Navigating in hyperspace: designing a structure-based toolbox,” *Communications of the ACM*, 37(2), pp. 87-96, 1994.
25. Schuett, H., Streitz, N., “HyperBase: A hypermedia engine based on relational database management system,” *Proc. 1st European Conference Based on a Relational Database Management System*, pp. 95-108, Versailles, France, Nov.28-30, 1990.
26. E. Spertus, “ParaSite: Mining structural information on the Web,” *Proc. 6th International World Wide Web Conference*, 1997.
27. Streitz, N., Haake, J., Hannemann, J., Lemke, A., Schler, W., Schuett, H. and Thuering, M., “SEPIA: A cooperative hypermedia authoring Environment,” *Proc. 4th ACM Conference on Hypertext*, pp. 11-22, 1992.
28. XML: http://www.w3.org/XML
29. XML Query: http://www.w3.org/XML/Query
30. W3C: http://www.w3.org/
31. R. Weiss, B. Veles, M. Sheldon, C. Nemprempre, P. Szilagyi, D. K. Gifford, “HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering,” *Proc. 7th ACM Conference of Hypertext*, 1996.