ConSearch: Using Hypertext Contexts as Web Search Boundaries

Zhanzi Qiu, Matthias Hemmje, Erich J. Neuhold

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

Abstract. This paper describes a schema for searching on the Web by making use of hypertext contexts that can be represented with new Web standards as search boundaries and argues that this would help to provide users more accurate results in their searches concerning specific topics or subject domains. It also proposes several issues to be addressed in making the schema applicable and presents a prototype system that addresses these issues and supports hypertext context-based search.

1 Introduction

The ability to browse is generally regarded as one of the most important reasons for using hypertext, while searching facilities should also be supported in modern hypertext environments [9]. The World Wide Web is such a hypertext environment. Because of its huge scale and arbitrary structure, it creates many challenges for the development of its searching capabilities.

In the current Web, most links are not typed, and there is no link-based composition mechanism. Thus the Web lacks explicit structural meta information [18], and the search engines on it are typically keyword-based. With such engines, people usually get a large amount of pages that they can not process, or even more, many of the pages are totally irrelevant to their information needs, especially when they search for information on specific topics.

To improve the ability of expressing structures and semantics on the Web, several new standards, mainly XML (Extensible Markup Language) [21] and RDF (Resource Description Framework) [15], are developed or under development. These standards open new opportunities to improve the information access on the Web. However, it is an open question how to make use of the structural and semantic information that can be represented with the standards efficiently for search purposes. This paper describes a part of our effort to answer this question.

In this work, we focus on making use of hypertext contexts, one of the main high-level hypermedia structures that can be represented with the new Web standards for searching on the Web. We view a hypertext context as a mechanism to specify the scope of the information space to be examined in a search and argue that this would help to improve the search response time and the quality of the results of searches concerning a specific topic or subject domain.

The rest of the paper is organized as follows. Section 2 describes the concept of hypertext contexts and gives simple examples. Section 3 answers the question how hypertext contexts can be represented with new Web standards. Section 4 proposes a schema for searching on the Web by making use of hypertext contexts and discusses issues to be addressed to make the schema applicable. Section 5 presents a prototype system that supports hypertext context-based search. Section 6 mentions related work. Finally, Section 7 summarizes this work and outlines our future activities.

2 Hypertext Contexts

2.1 Concepts

A *hypertext context* is a generic high-level hypermedia structure that groups together a set of nodes and links into a logical whole. The idea of *hypertext contexts* was first introduced by Schwartz and Delisle [16]. *Contexts* partition the data within a hypertext graph. A hypertext graph contains one or more *contexts*; each *context* has one *parent context* and zero or more *child contexts* [3].

Precisely, if C is a *hypertext context*, then its contents must define a pair (N, L), where N is a set of nodes in a hypertext graph and L is a set of links whose end nodes belong to N. 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.

If C1, C2 are two *hypertext contexts*, we say that C1 contains C2 only when all nodes and links in C2 are contained in C1. That is, if C1 defines a pair (N1, L1), C2 defines a pair (N2, L2), and C2 $\subseteq$ C1, then N2$\subseteq$ N1 and L2$\subseteq$ L1. Contrary, if N2$\subseteq$ N1 and L2$\subseteq$ L1, then C2 $\subseteq$ C1. We say C1 is the *parent context* of C2, and C2 is one of the *child contexts* of C1.

Usually, a *hypertext context* is said to be a container for a group of nodes, while the links between the nodes are thought to be included implicitly in the context. We adopt this meaning of hypertext contexts later in this paper. That is to say that unless clearly specified, the *components* of a *hypertext context* refer to the nodes contained in it.

In practice, *hypertext contexts* can be used to support configuration, private workspaces, and version history trees [16]. They can be used as a mechanism to describe different context views of the same hyperdocuments, tuned to different applications or classes of users of the documents [4]. In this sense, a (group of) hyperdocument(s) may contain any number of *hypertext contexts*. Such *hypertext contexts* can exist statically in hypertext document collections or can be created dynamically by hypertext based information systems.

2.2 Examples

Typical examples of hypertext contexts that exist statically in hypertext document collections are various maps, paths, guided tours and focused node lists related to a particular topic or subject domain. These hypertext contexts are usually encoded in concrete nodes, maybe one context in one node, or several contexts in one node. Many such nodes (pages) exist on the current Web but can not be recognized automatically. So are the hypertext contexts described in them.

For instance, the page "DELITE publications" (http://www.darmstadt.gmd.de/delite/Publications/) contains a complete list of all publications from the division DELITE of GMD-IPSI (http://ipsi.gmd.de/), and each item in the list points to a DELITE publication. What the list describes is actually a hypertext context that is composed of all nodes (pages) about DELITE publications.

Moreover, it can be said that the items listed under each year constitute a child context of the above large one.

As for the hypertext contexts that are created dynamically by hypertext based information systems, the most typical examples are various search results, which are attained by computation against certain query criteria. Like static hypertext contexts, they can not be recognized automatically on the current Web. In addition to these typical examples, dynamic hypertext contexts can apparently be attained from Boolean operations performed to hypertext contexts existing in systems.

3 Standard Representation of Hypertext Contexts

Link any other information, hypertext contexts, no matter static or dynamic, can be shared and reused in a large information space like the Web only when they are represented in a standard format. Fortunately, the new Web standards, XML and RDF, have made this possible. With them information providers can describe the hypertext contexts in their Web resources explicitly and browser and search systems can recognize the contexts automatically.

3.1 Representing Hypertext Contexts with XML Extended Links

In the XML model, the linking mechanisms supported are specified in the XML Linking Language (XLink) [20]. A hypertext context that contains Web resources as its components can be described with a linking element for an `extended link.` Each component of the hypertext context is given in a `locator` element, which is a child element of the linking element.

For instance, the following encoding (an out-of-line extended link) describes a hypertext context *mycontext* that is composed of 3 nodes:

```
<mycontext xml:link=”extended” inline=”false”>
  <locator href=”doc1” role=”description”>
  <locator href=”doc2” role=”commentary”>
  <locator href=”doc3” role=”reference”>
</mycontext>
```

A `locator` may indicate a resource, which itself contains an extended link, i.e., also describes a hypertext context. In this way, the *parent-child* relationship between hypertext contexts can be represented.

Especially, hypertext contexts and their parent-child relationship may be described with `extended link group` elements (a special kind of `extended links`) and `extended link document` elements (a special kind of `locator` elements). The `Steps` attribute of the `extended link group` elements can be given a numeric value that serves as a hint from the author to any system as to how many levels hypertext contexts exist.

That is, an `extended link group` element may be used to store a list of links to other resources that together constitute an interlinked group. Each such resource is identified by means of an `extended link document` element and may itself contain an `extended link.` In this case, the `group` element describes a hypertext context that is the parent of the hypertext contexts described in the resources indicated by the `document` elements.

To give an example, suppose the above descriptions about *mycontext* are stored in *mycontext.xml*, a possible parent context of *mycontext* can be described as follows:

```
<group xml:link="group" steps=2>
<document xml:link="document" href="mycontext.xml" role="recommend"/>
<document xml:link="document" href="..." role="..."/>
...
</group>
```

These descriptions are contained in a document other than *mycontexts.xml*.

3.2 Representing Hypertext Contexts with RDF Containers

In the RDF model, a hypertext context can be represented in a `container`. Each component (node) of the context is referred to with a `resource`. (In RDF, the term *resource* is in most cases a metaphor of *node*).

RDF defines three types of container objects: `bag`, `sequence`, and `alternative`. The first two are used to declare the multiple values of a property, and the third is to declare alternatives for the (single) value of a property. For representing hypertext contexts, the first two types of `containers` fit better. Besides, the difference between them, i.e., one declares unordered lists and another declares ordered lists, does not make much sense, as the sequence of components in a hypertext context is not cared.

For example, a hypertext context that consists of resources about people working in the project *delite-online* can be described as follows:

```
<rdf:RDF><rdf:Description
about="http://www.darmstadt.gmd.de/delite/projects/delite-
online.html">
    <s:people><rdf:Bag>
        <rdf:li resource="http://www.darmstadt.gmd.de/~qiu/">
        <rdf:li resource="http://www.darmstadt.gmd.de/~lhuang/">
        <rdf:li resource="http://www.darmstadt.gmd.de/~moelle/">
      </rdf:Bag></s:people>
 </rdf:Description></rdf:RDF>
```

Such RDF descriptions can exist in a separate RDF document, or be contained in the head of an HTML document.

It is clear that by embedding a RDF `container` in another RDF `container`, the parent-child relationship between hypertext contexts can be represented.

4 Using Hypertext Contexts as Web Search Boundaries

With the possibility to represent hypertext contexts in a standard way, it is time for us to consider how to make use of the information contained in these objects for various purposes. We believe that a few new search methods may be developed for Web users.

A very straightforward consideration is to use hypertext contexts as a mechanism to specify the information space to be examined in a search activity. This is because a hypertext context effectively cuts a boundary between its containing nodes and links from other nodes and links that are out of it (as shown in Figure 1). This mechanism should prove to be useful for improving the results of searches concerning specific topics or subject domains. On the current Web, this kind of search activity is

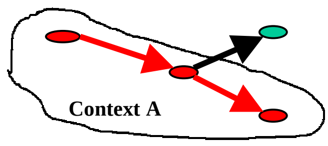

Figure 1. A simple hypertext context

usually frustrating. With the typically keyword-based search engines, users usually get a large amount of pages that they can not easily process, while on the other hand many of the pages are totally irrelevant to their information needs.

4.1 Primary Experiments

To explore our idea of using hypertext contexts as Web search boundaries, we have developed a prototype system called delite-WebGlimpse and performed some experiments with it [14]. This system allows users to define hypertext contexts themselves when they browse and search on the Web and then submit searches within the context boundaries. For instance, if one wants to find out which people in GMD IPSI are doing research related to “data mining”, one can define a hypertext context that contains all the personal homepages of GMD IPSI staff as its components. One then submits query “data mining” and asks the system to search only in the context. In this case, one gets only 3 hits and the precision of this result is 100%. Comparatively, if the search is done in the whole GMD IPSI Web site (start from http://ipsi.gmd.de/), the user will get 375 hits (at the time we performed the experiments) and has to filter the pages himself. Apparently this situation will be much more serious when the search is done in the whole Web with global search engines.

Our experiments have demonstrated that using hypertext contexts as Web search boundaries is really useful. A large amount of non-relevant pages can be filtered out before any further search activities or pattern matching processes take place, and the search results are thus more specific and more relevant to users’ information needs.

4.2 Issues for Supporting the Use of Hypertext Contexts as Web Search Boundaries

In practice, to enable the use of hypertext contexts as Web search boundaries, there are a few issues to be addressed.

The first issue is to represent hypertext contexts in a standard way and make it 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 Web search system that intends to support the use of hypertext contexts as search boundaries should be able to

- extract from the Web sufficient hypertext context information and index the information efficiently so that the hypertext contexts themselves can be queried with the information later,
- efficiently organize the hypertext context information after indexing internally so that the hypertext contexts can be retrieved at a speed acceptable by users,
- provide a user-friendly interface to enable users to search for hypertext contexts and specify hypertext contexts as search boundaries in a comfortable way, and
- implement the searches within the boundaries of specified hypertext contexts with acceptable performance and provide sound search results.

In the following, we will present how our prototype system, ConSearch, addresses these issues.

5 ConSearch - A Prototype System

ConSearch is the prototype system we design to test our idea of using hypertext contexts that can be represented with new Web standards as Web search boundaries. In the following we first give a high-level discussion of its architecture and then introduce several technologies in its implementation. Finally, some evaluation issues about it are discussed.

5.1 Architecture Overview

The overall picture of ConSearch is shown in Figure 2. It includes four basic engines representing different aspects.

The **info agent** is responsible for gathering hypertext context information from the Web and storing the information in databases. It contains a **URI server**, an **extractor**, a **storage server**, and an **indexer**. The URI server sends lists of URIs to be fetched to the extractor. The extractor fetches the data resources (including HTML, XML and RDF documents), extracts hypertext context information from the resources, represents the information with a specific ConSearch internal format, and sends the documents of the internal format to the storage server. The storage server stores the document of the internal format into a repository, from which the indexer reads the documents, parses them and performs indexing functions. The indexing results are sent to the database for hypertext contexts.

The **query engine** receives queries, lets users search for hypertext contexts, specify hypertext contexts as search boundaries, transfers the queries to the retrieval engine, and presents search results derived by the retrieval engine to users. The engine will be implemented as Web browser clients with a form-based user interface.

The **retrieval engine** is responsible for using information in the database to derive hypertext contexts specified, implementing searches within the hypertext contexts, and sending search results to the query engine. This engine is the part in which the system cooperates with other search systems, as described in Section 5.5 later.

The **database manager** is a backbone of the entire system. It receives data from the info agent and provides data to the retrieval engine. In ConSearch, the database manager is an object relational Informix Universal Server DBMS.

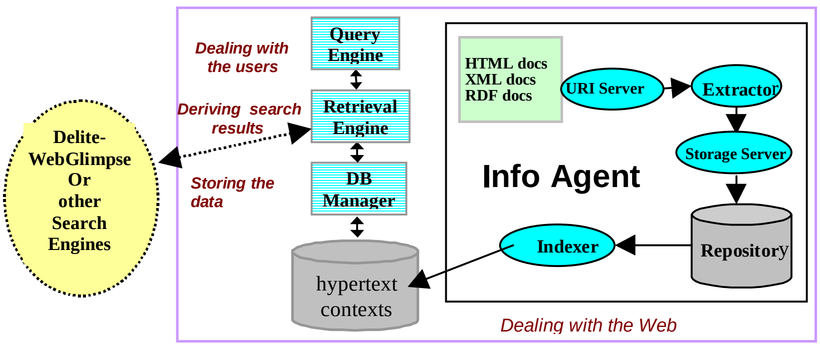

Figure 2. Architecture Overview

5.2 Hypertext Context Information Gathering and Indexing

Corresponding to how XML and RDF represent hypertext contexts, ConSearch gathers and indexes the following information (referred to as hypertext context information in this paper) for each hypertext context:

1. the URI of the Web resource that the context talks about (called aboutURI later),
2. the URI of the Web resource that describes the context (called sourceURI later), and,
3. descriptive keywords, which can be extracted from RDF property types, sourceURI, aboutURI, roles of external links, names of the link elements that contain the context, and special descriptive information about the context with regards to certain Web resources.

All this information is gathered with the JEDI (Java Extraction and Dissemination of Information) [6] tool, which is used in ConSearch as the extractor. The 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.

Since a Web search system like ConSearch is fed by heterogeneous textual information sources, the translation of the incoming texts to an internal format is inevitable. The internal ConSearch format is provided in form of an XML DTD and aims to cover the demand of describing the hypertext context information exhaustively. Figure 3 shows an example of a hypertext context encoded in the ConSearch internal format. The hypertext context is assigned a name. All the descriptive keywords extracted are listed in the content of the *description* element. Besides, its sourceURI, aboutURI, components are all contained in this document.

Such documents are compressed and stored into a repository by the storage server. The indexer reads the repository, uncompresses the documents, parses them and performs keyword-indexing functions to the content of the element *description*. The indexing results and all URIs related to the hypertext contexts are sent to the database, which owns the schema as described in the following section.

```
<context name="DelitePub1999">
  <source>http://www.darmstadt.gmd.de/delite/publication/1999.rdf</source>
  <about>http://www.darmstadt.gmd.de/delite/publication/index.html</about>
  <description>delite, publication, 1999</description>
  <components>
    <component>
       http://www.darmstadt.gmd.de/delite/publication/1999/pub1.html
    </component>
    <component>
       http://www.darmstadt.gmd.de/delite/publication/1999/pub2.html
    </component>
      ...
  </components>
</context>
```

Figure 3. A hypertext context produced by JEDI

5.3 Internal Organization of Hypertext Context Information

In ConSearch, hypertext context information after indexing is stored in an Informix Universal Server database. Every hypertext context has an associated ID number (called contextID) which is assigned whenever a new hypertext context is parsed out of a Web resource. To represent the components of hypertext contexts and the aboutURIs and sourceURIs of hypertext contexts in a non-redundant way, every Web resource also has an ID number (called docID later) whenever a new URI is parsed out. Similarly, every keyword used to describe hypertext contexts is assigned an ID number (called wordID later) in order not to waste space.

Based on the above basic point of view, the database of hypertext contexts is built with the schema as shown in Figure 4. The *URI Table* contains URIs that are parsed and the primary serial numbers assigned to the URIs. The *Word Table* contains keywords that are used to describe the hypertext contexts and their primary serial numbers. The *Context Table* contains the primary serial number of hypertext contexts and IDs for aboutURIs and sourceURIs of the contexts. The *Context Component Table* lists the components of hypertext contexts. The *Context Parent Table* represents the parent-children relations between hypertext contexts. Finally, *The Context-Word Table* builds relationships between hypertext contexts and the keywords used to describe them.

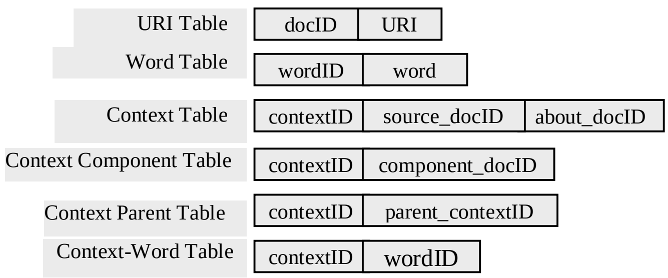

Figure 4. Database for hypertext contexts

With this internal organization of hypertext context information, ConSearch is able to support the specification of hypertext contexts as search boundaries, not only single hypertext contexts, but the combination of hypertext contexts as well. Furthermore, searching for hypertext contexts themselves by inputting keywords and/or the aboutURIs and sourceURIs is also possible.

5.4 User Interface for Querying and Specifying Hypertext Contexts

ConSearch provides a form-based interface to enable users to query hypertext contexts and specify hypertext contexts. As shown in Figure 5, users can query hypertext contexts by specifying sourceURIs, aboutURIs, and keywords. After users confirm the specification, the system will display all the candidate contexts (in some case, maybe not only one context is found) by showing their sourceURIs, aboutURIs, and all their descriptive keywords (as shown in Figure 6). Then users can adjust their queries or go to take a look at the components of the candidate contexts (as shown in Figure 7) and/or make their choice of the contexts to be used as the search boundaries for their queries. As Figure 6 shows, users can also ask the system to do Boolean combination of the hypertext contexts found and use the resulting hypertext context as search boundaries.

5.5 Implementing Searches within Hypertext Contexts

To implement searches within the boundaries of specified hypertext contexts, ConSearch is designed to be able to cooperate with other keyword-based search systems, which are either global or local. For instance, the current version of ConSearch is integrated with our delite-WebGlimpse [14], which uses Glimpse [7] as its search engine.

In this integration, ConSearch is responsible to provide the URLs of the component pages of the specified hypertext contexts to delite-WebGlimpse and delite-WebGlimpse maps these URLs to the file names of the pages it collects[1] and save the file names in a context file. This context file is then compressed and sent to Glimpse, which supports very flexible functions to limit the search to only parts of the files in a collection[2]. In this way, searching within the boundaries of specified hypertext contexts is implemented.

When integrated with global search systems, such as InfoSeek [10] and AltaVista [1], ConSearch works as a meta search engine. It sends queries to one or more other systems according to users' selection (as shown in Figure 6 and 7) to get results, combining the results and using the specified hypertext context as filters. With this process, it provides users more specific final results that are relevant to their information needs and saves much time that the users have to spend to get the results in current normal ways.

In case search engines restrict the number of results to e.g. 200 out of the possible thousands, ConSearch asks for a more tight cooperation with those engines. That is, ConSearch should be allowed to access the whole search engine ranking list or the large part of the list. In this way, the results that are within the specified context but are in the tail of the search engine ranking will not be missed.

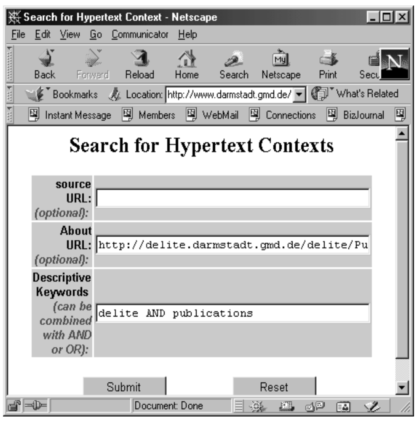

Figure 5. Search for hypertext contexts

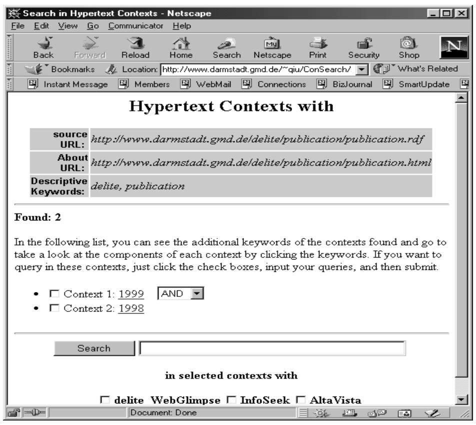

Figure 6. Show found hypertext contexts and enable search in selected hypertext contexts

---

[1] delite-WebGlimpse collects all remote pages locally with a mapping mechanism from urls to file names.

[2] Glimpse provides several options to enable users to filter files in search. One option is –f, which reads a list of file names from a given file and uses only those files in search.

5.6 Evaluation Issues

The most important measure of a search engine is the quality of its search results. It is quite certain that by supporting the use of hypertext contexts as Web search boundaries, ConSearch provides users with more accurate search results. Our experiments with delite-WebGlimpse [14] have proved this. As described in 4.1, if one asks the system to search within a hypertext context containing the personal homepages of GMD-IPSI (http://ipsi.darmstadt.gmd.de/) staff (at present 81) for one's query "data mining", one will get only 3 hits (as shown in Figure 8). And the precision of the result is 100%. Similarly, for the query "information retrieval", one will get 5 hits with also the precision 100%. In this way, the special information needs of searching for GMD-IPSI members who have research interests in the fields of "data mining" and "information retrieval" are satisfied. Comparatively, if the searches are done on the whole GMD-IPSI site, for such queries, one will get several hundreds of hits and has to filter the hits themselves.

Aside from search quality, a thorough evaluation about a system which intends to support hypertext context-based search should cover its storage requirements, its performance in extracting and indexing the information, its quality and performance in searching hypertext contexts, and its performance in implementing searches within specified hypertext contexts. Until now, we have not done such a thorough evaluation for ConSearch. However, there is some significance that ConSearch can scale well to the size of the Web as the Web grows, as it chooses a scalable DBMS and stores all hypertext context information gathered and indexed in databases.

As the new Web standards become more adopted and more hypertext contexts represented with the standards will be provided on the Web, a thorough evaluation about the method and the system presented in this paper will be done.

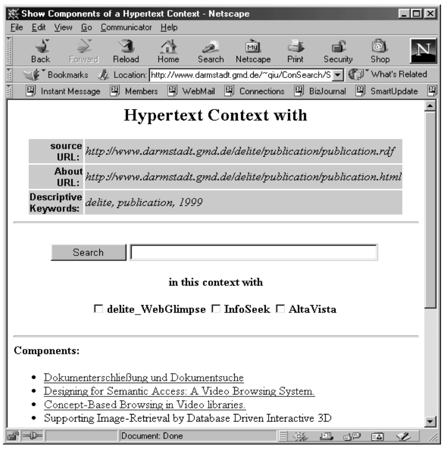

Figure 7. Show components of a hypertext context and enable search in the context

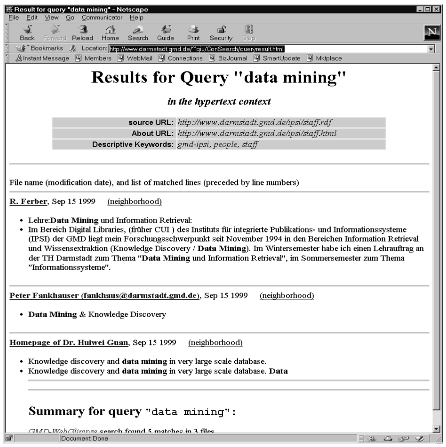

Figure 8. Result for query "data mining" in the context "GMD-IPSI staff"

6 Related Work

The idea of drawing a boundary for the information space to be examined in search activities has been reflected in some popular search systems which provide category search, fielded search, "search within results", “find similar pages", and so on, such as Yahoo [22], InfoSeek [10] and Lycos [12]. It can also be seen in search agents that traverse the Web looking for specific information in real time [17] or allow users to set up their own local searches [13]. In comparison, the schema we present in this paper for searching in the Web space by using hypertext contexts is more general and covers the schemas proposed by other systems. Furthermore, it meets the development of the Web in its ability of expressing structures and semantics.

There is a trend of making use of additional structural information to improve Web searching. Structural information has so far been used for enhancing relevance judgements, ranking Web pages or other purposes. Among the work in this area the achievements of Google [8; 2] and Clever [5; 11] are most attractive. Both systems use weighted link popularity as a primary criteria in their ranking mechanism. As far as we know, none of these systems have taken into account the Web’s new abilities in expressing structural and semantic information in their search algorithms yet.

Delite-WebGlimpse [14] is our starting effort to explore our idea of using hypertext contexts as Web search boundaries. It aims at the current Web (with no explicitly represented hypertext contexts) and provides facilities to help users to define hypertext contexts themselves and enable them to search in those hypertext contexts. The results we get from the experiments done with it are the first proof for the value of our idea.

XML [21] and RDF [15] are ongoing effort of W3C [19] to improve Web’s ability in expressing structure and semantics. Accompanying W3C’s effort, a number of XML and RDF software tools have been developed or under development. A list of such tools can be found under http://www.w3.org/XML/ and http://www.w3.org/RDF/. Most of them are parsers, generators, editors and browsers. They can be integrated into more complicated systems that intend to handle XML and RDF resources.

7 Summary and Future Work

This paper describes a part of our effort to answer the open question how to make use of the structural and semantic information that can be represented with new Web standards efficiently for search purposes.

The general schema presented in this paper of using hypertext contexts as Web search boundaries has been proved to a large degree by our work and some work from other people. It may further prove to be useful by being integrated into large search engines and allowing users to get more accurate results, as the new standards become more widely adopted and a large number of hypertext contexts are provided with the standard formats.

The prototype developed for implementing the schema is currently still primitive. More facilities to help users to specify a hypertext context for a given query should be provided. Further thorough evaluation will be done to determine to what degree hypertext contexts work in their role as search boundaries, and how Web search engines will index data efficiently if hypertext contexts are to be supported. It will be also a concern of this work how to enable information providers to produce hypertext contexts easier and efficiently.

In addition to using hypertext contexts as Web search boundaries, we intend to develop more search methods that make use of this kind of information. Furthermore, the value of other kinds of high-level hypermedia structures that can be represented with new Web standards for search purposes will also be explored in our future work.

References

1. AltaVista: http://www.altavista.com/
2. Brin, S. and Page, L., "The anatomy of a large-scale hypertextual Web-search engine," *Proc. 7th International World Wide Web Conference*, 1998.
3. Campbell, B., Goodman, J. M., "HAM: A general purpose hypertext abstract machine," *Communications of the ACM*, 31(7), July 1988, pp. 856-861.
4. Casanova, M. A., Tucherman, L., "The nested context model for hyperdocuments," *Proc. of Hypertext'91*, pp. 193-201.
5. Clever: http://www.almaden.ibm.com/cs/k53/clever.html
6. 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.
7. Glimpse: http://glimpse.cs.arizona.edu/
8. Google: http://google.stanford.edu/
9. Halasz, F.G., "Reflections on Notecards : seven issues for the next generation of hypermedia systems," *Communication of the ACM*, 31(7), pp.836-852, 1988.
10. InfoSeek: http://www2.infoseek.com/
11. Kleinberg, J., "Authoritative sources in a hyperlinked environment," *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 1998. Also appears as IBM Research Report RJ 10076(91892), May 1997.
12. Lycos: http://a2z.lycos.com/
13. Miller, R., Bharat, K., "SPHINX: A framework for creating personal, site-specific Web crawlers," *Proc. 7th International World Wide Web Conference*, Brisbane Australia, April 1998.
14. Qiu, Z., Hemmje, M., Neuhold, E., "Towards Supporting User-Defined Hypertext Contexts in Web Searching", Proceedings of IEEE ADL'2000 , May 22-24, 2000, National Institutes of Health, Bethesda, MD, USA.
15. RDF: http://www.w3.org/TR/REC-rdf-syntax/
16. Schwartz, M., Delisle, N., "Contexts – A partitioning concept for hypertext," *ACM Transactions on Office Information Systems*, 5(2), April 1987, pp. 168-186.
17. Sumner, R. G., Yang, K, and Dempsey, B. J., "An interactive WWW search engine for user-defined collections," *Digital Libraries 98*, Pittsburgh PA, USA.
18. Trigg, R. H., "Hypermedia as integration: Recollections, reflections and exhortations," Keynote Address in *Hypertext'96 Conference*. Xerox Palo Alto Research Center.
19. W3C: http://www.w3.org/
20. XLink: http://www.w3.org/TR/xlink/
21. XML: http://www.w3.org/XML
22. Yahoo: http://www.yahoo.com/