Element Retrieval in Digital Libraries: Reality Check

Philipp Dopichaj
dopichaj@informatik.uni-kl.de
University of Kaiserslautern, AG DBIS
Gottlieb-Daimler-Str.
67663 Kaiserslautern

ABSTRACT

Although research on XML element retrieval is steadily gaining popularity, it is not clear if and in what form element retrieval can be useful in real-world scenarios. In this paper, we compare the XML element retrieval models used in the INEX workshop with the search interfaces of two online digital library services. We demonstrate that element retrieval is indeed useful for digital libraries and that there is a lot of room for improvements in this field.

1. INTRODUCTION

The Initiative for the Evaluation of XML Retrieval (INEX) provides the infrastructure for conducting (XML) element retrieval experiments [2, 1]. So far, there has been no consensus about what a real-world application of element retrieval might look like, which was identified as a major obstacle to realistic experiments in this area [3]. In this paper, we try to address this by looking at two commercial online digital library systems that provide search functions similar to those used at the INEX workshops.

We focus on two online library services that offer full-text search for their online books, Books24x7[1] (launched in 1999) and Safari[2] (launched in 2001).[3] Each of them offers access to several thousand technical books through a web interface. The sheer amount of information necessitates good search interfaces so that the users can find relevant books or sections without problems.

In Section 2, we look at the ways in which these library services support different search models and contrast them to what is done at INEX. Section 3 addresses some aspects that are of relevance to INEX user models.

2. REAL-WORLD SEARCH INTERFACES

Both of the library services provide search interfaces that resemble the retrieval tasks at INEX at least to some extent. We first look at the result views that group by documents ("Relevant in Context" or "Fetch and Browse" in INEX terminology) or by sections (similar to "Thorough" retrieval in INEX) and finally examine to what extent structural queries are supported.

2.1 Results Grouped by Document

In traditional (flat) information retrieval, results are typically presented as a list of matching documents. For books, this alone is not a viable option: The user also needs to know *where* in the books he can find the relevant text, so there should be further information about relevant sections. INEX offers the "Relevant in Context" task (formerly "Fetch and Browse"), where the relevant elements are first sorted according to the book's score and then (inside each book) according to the element's score, and the "Best in Context" task, where the best entry point to each document is sought.

Both Books24x7 (see Figure 1) and Safari (see Figure 2) support displaying results in this fashion: They present a list of relevant books, and for each book a list of the titles of the most relevant sections or chapters from that book. In contrast to the "Relevant in Context" task, where the number of elements from a given document is unlimited, only three sections are displayed for each book, so this is probably more comparable to the "Best in Context" task. The user then has the option of navigating to a certain book, or directly to a section from that book. In both services, the in-book results can overlap, that is, it is possible that both a chapter and a section from that chapter appears in the results (note that this would not be allowed in the INEX task).

Comparing the results from Safari's book-based result display to their flat results (described in the following section), the books appear to be ranked by the score of the most relevant section.

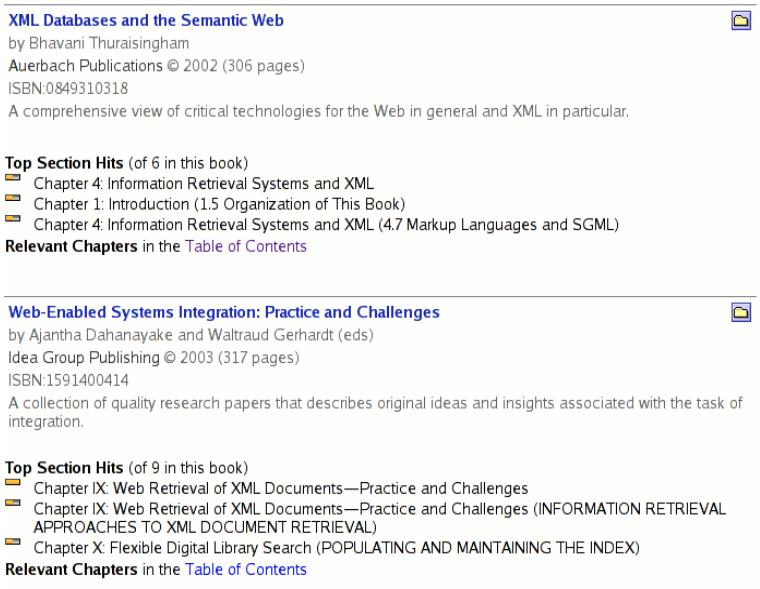
2.2 Flat Results

Another way of displaying results is presenting a flat list of relevant fragments (or snippets with pointers to the fragments). This type of result list has the advantage of being familiar to most searchers, as it is the format that most web search engines use. In contrast to web search engines, however, the results can overlap, so it is possible to have both a chapter and a section from this chapter in the results.

INEX offers two approaches addressing the issue of overlapping results: The "Thorough" task ignores the issue and

[1] see http://www.books24x7.com/
[2] see http://safari.oreilly.com/
[3] Although it is not certain that they use XML for the storage of their documents, they definitely use a semi-structured format.

SIGIR 2006 Workshop on XML Element Retrieval Methodology
August 10, 2006, Seattle, Washington, USA.



(a) Results page

(b) Table of contents of the first result with markers indicating the relavance of the chapters

Figure 1: Books24x7 search

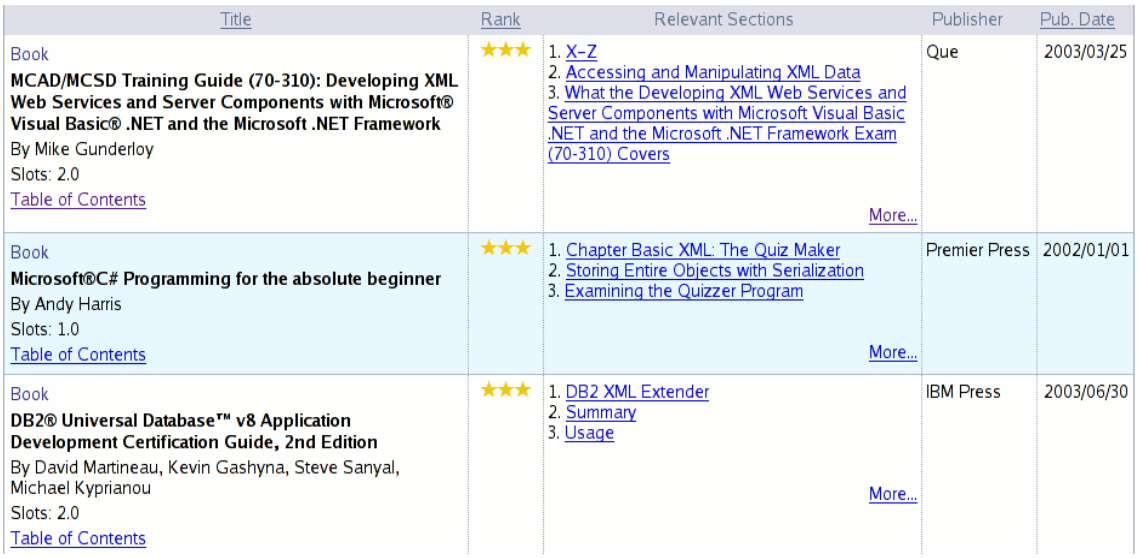


Figure 2: Safari search: View by Book

thus allows overlapping results to be displayed; it is meant to be a system-oriented task that aims at finding out whether a search engine can find *all* relevant results. The "Focused" task disallows overlapping results, so the search engine has to decide which result is more relevant to the user.

Surprisingly, Books24x7 offers no flat results, and Safari offers only a view that closely resembles the "Thorough" task, called "View by Section". This view includes a short snippet from the relevant sections, with the search terms highlighted, and a hyperlink to the complete section; see Figure 3. The search can also be restricted to a single book, so that the relevant sections inside a single book can be identified easily.

2.3 Content-and-Structure Search

One interesting research topic is whether structural hints in the query—for example, "find articles about information retrieval that cite the INEX proceedings"—help the retrieval engine. INEX uses NEXI, a special search language derived from XPath [4]; this language is not suitable for ad-hoc queries, but it can help to evaluate whether structural hints have any positive effects.

Obviously, casual users of online digital libraries do not have intricate knowledge of the internal schemas of the documents, so the library services offer only limited support for structural queries in their advanced search interfaces: You can search in meta information such as author or publisher or in book titles.

Safari's advanced search interface also offers limited content-and-structure search by offering a choice of one of the following options:

- The full text
- Code fragments only
- Section title words only
- Tips and how-tos only

These options appear to be used as retrieval hints only (vague content-and-structure search): Neither is the granularity of the retrieval results affected, nor are only sections returned that fulfil this condition.

Even this simple form of structured queries is not used as the default search interface. This might indicate that the default (content-only) search interface is sufficient for most queries and users, but that the more complex interface is needed for advanced searchers and more complex information needs.

2.4 Book Search without Element Retrieval

For comparison, we also briefly examined a search engine for books that does not use element retrieval because the books in its index are not available in a semistructured format. Google Book Search[4] differs from Safari and Books24x7 in that it does not offer access to the full text of the books it has indexed. In their own help text[5], they state:

> Google Book Search helps you discover books, not read them online. To read the whole book, we encourage you to use a "Buy this book" link to purchase it or the "Find this in a library" link to look for a local library that has it.

Along the results still under copyright, they provide links to several online book stores. As such, their search service can be seen as a means to find references to works satisfying the information need, instead of fragments that themselves satisfy the information need. Some publishers allow Google to show a few relevant pages scanned from the paper versions, with the matching terms highlighted. The pages do not necessarily correspond to logical units in the text, so it might well happen that the search term appears at the end of a page, but the relevant information is wrapped to the next page. Element retrieval has the potential to offer better results, but it cannot be used in this case because the books are not available to Google in a semi-structured format.

3. FURTHER NOTES

Apart from the search interface, several other aspects are of interest to the element retrieval community. In this section, we speculate how the different subscription models might affect the demands of the users. Next, we look at the history of Safari's search interface to show that the current interface is at least usable (unfortunately, we have been unable to reconstruct old versions of the Books24x7 interface).

3.1 Subscription Models

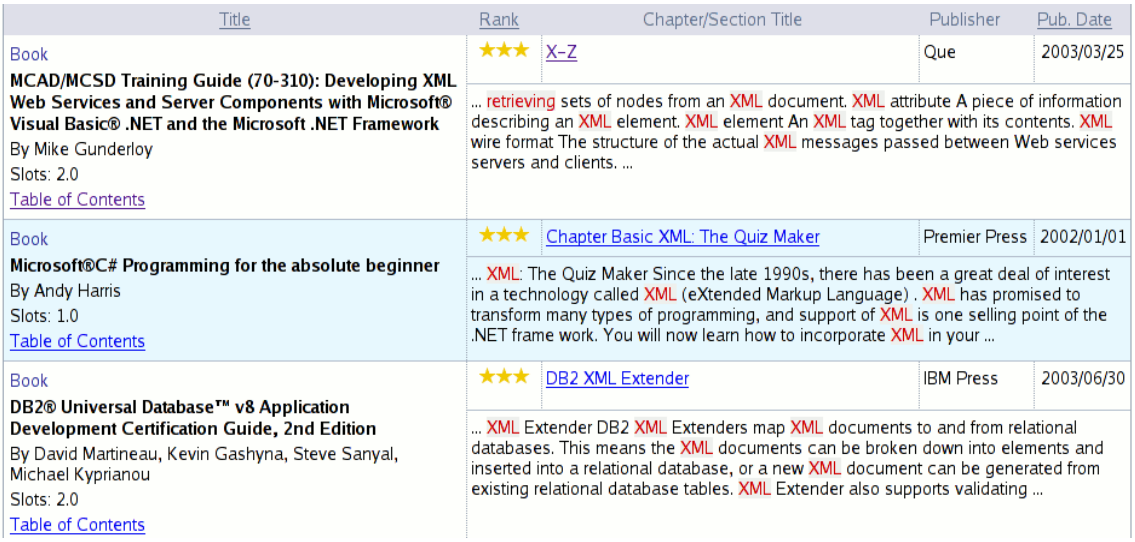
Both Books24x7 and Safari are subscription-based, but the type of subscription differs substantially: Books24x7 subscribers have full access to all books at all times. Safari users only have access to a limited number of books of their own choice at any given point in time: They have a limited number of slots on their virtual bookshelf, and once they put a book on there, it must stay there for at least a month.

These different subscription models affect the users' requirements on the search interfaces: Books24x7 users have no access restrictions, so they might well be interested in locating small, very specific parts of the books to answer the queries; diversity (results from many different books) can be helpful to get the complete picture. Safari users, on the other hand, should avoid putting books on their bookshelf that are not useful to them, so the search interface should help them find books which contain the highest amount of relevant information. Finding relevant sections for a specific query is not such a high priority here, because putting a book on one's bookshelf just for reading a single section might be wasteful. The "View by Section" feature is most probably used mainly for searching the books on one's bookshelf (to find relevant sections in the available books), or possibly to get short fragments of the texts in the result list, which is not available in the "View by Book" result list.

3.2 Development of Safari Search

We can assume that the search interfaces of the book services are demand-driven, which means that unhelpful features would be removed after some time. Thus, it is interesting to see that the search interface of Safari has been virtu-

[4]see http://books.google.com/
[5]see http://books.google.com/intl/en/googlebooks/help.html



Title	Rank	Chapter/Section Title	Publisher	Pub. Date
Book **MCAD/MCSD Training Guide (70-310): Developing XML Web Services and Server Components with Microsoft® Visual Basic® .NET and the Microsoft .NET Framework** By Mike Gunderloy Slots: 2.0 Table of Contents	★★★	X-Z	Que	2003/03/25
	... retrieving sets of nodes from an XML document. XML attribute A piece of information describing an XML element. XML element An XML tag together with its contents. XML wire format The structure of the actual XML messages passed between Web services servers and clients. ...			
Book **Microsoft®C# Programming for the absolute beginner** By Andy Harris Slots: 1.0 Table of Contents	★★★	Chapter Basic XML: The Quiz Maker	Premier Press	2002/01/01
	... XML: The Quiz Maker Since the late 1990s, there has been a great deal of interest in a technology called XML (eXtended Markup Language) . XML has promised to transform many types of programming, and support of XML is one selling point of the .NET frame work. You will now learn how to incorporate XML in your ...			
Book **DB2® Universal Database™ v8 Application Development Certification Guide, 2nd Edition** By David Martineau, Kevin Gashyna, Steve Sanyal, Michael Kyprianou Slots: 2.0 Table of Contents	★★★	DB2 XML Extender	IBM Press	2003/06/30
	... XML Extender DB2 XML Extenders map XML documents to and from relational databases. This means the XML documents can be broken down into elements and inserted into a relational database, or a new XML document can be generated from existing relational database tables. XML Extender also supports validating ...			

Figure 3: Safari search: View by Section

ally unchanged since at least October 2002, as witnessed by the Web Archive's page from October 13[6]. Even the newest changes from June 2006 do not affect the basic search model, the changes are mostly cosmetic. The most notable difference is that the default view switched from "View by Book" to "View by Section".

Unfortunately, the oldest version of the documentation is available in the Web Archive's cache from August 2002[7] lacks the relevant screen shots, so it is hard to tell what exactly was changed from this version to the next; from the textual description, it appears that a variant of the "View by Book" interface was available, whereas "View by Section" was missing. If this is the case, it may indicate that this feature was requested by users. Along with the recent change of the default view, this suggests that a flat result list is an important user interface for element retrieval.

The stability of the search interface does not imply that the current version is the best possible interface, but it does suggest that element retrieval is useful, and that there is some use to both book-based and section-based result lists.

4. DISCUSSION POINTS

We have seen that the models of element retrieval that are used in the real world do not always match the models assumed in the research community and INEX. This does not imply that one side is right and the other side is wrong; in particular, the commercial entities using element retrieval do not appear to have conducted extensive usability studies for their user interfaces. The fact that these user interfaces have been in use for several years implies that they are at least acceptable, so we can assume they are reasonable starting points for further refinements. We still need to investigate what we should adapt, and we definitely need to do more usability studies.

[6]see http://web.archive.org/web/20021209040844/safari.oreilly.com/?mode=Help

[7]see http://web.archive.org/web/20020818170606/safari.oreilly.com/mainhlp.asp?help

The following questions might be starting points for a discussion:

- Is element retrieval useful for texts of all lengths, or is it primarily useful for long texts such as books?
- Is a document-based display of results more natural than an element-based one?
- Is the "Thorough" task *really* system-oriented? Both online library services present overlapping results, so users apparently do not mind too much.
- What user models can we derive from these use cases?
- Can we cooperate with a provider of a digital library for INEX? (How do our results compare to those returned by the default search engines?)

5. REFERENCES

[1] Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Gabriella Kazai, editors. *INEX 2005 Proceedings*. Springer, 2006.

[2] Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Zoltán Szlávik, editors. *INEX 2004 Proceedings*. Springer, 2005.

[3] Andrew Trotman. Wanted: Element retrieval users. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, 2005.

[4] Andrew Trotman and Börkur Sigurbjörnsson. Narrow extended XPath I (NEXI). In Fuhr et al. [2].