What do Users Think of an XML Element Retrieval System?

Jaap Kamps[1,2] and Börkur Sigurbjörnsson[2]

[1] Archives and Information Science, Faculty of Humanities, University of Amsterdam
[2] ISLA, Faculty of Science, University of Amsterdam

Abstract. We describe the University of Amsterdam's participation in the INEX 2005 Interactive Track, mainly focusing on a comparative experiment, in which the baseline system Daffodil/HyREX is compared to a home-grown XML element retrieval system (xmlfind). The xmlfind system provides an interface for an XML information retrieval search engine, using an index that contains all the individual XML elements in the IEEE collection. Our main findings are the following. First, test persons show appreciation for both systems, but xmlfind receives higher scores than Daffodil. Second, the interface seems to take the structural dependencies between retrieved elements into account in an appropriate way: although retrieved elements may be overlapping in whole or in part, none of the test persons regarded this as problematic. Third, the general opinion of the test persons on the usefulness of XML retrieval systems was unequivocally positive, and their responses highlight many of the hoped advantages of an XML retrieval system.

1 Introduction

In this paper we document the University of Amsterdam's participation in the INEX 2005 Interactive Track. We conducted two experiments. First, we took part in the concerted effort of Task A, in which a common baseline system, Daffodil/-HyREX, is used to study test-persons searching the IEEE collection. Second, as part of the Interactive Track's Task B, we conducted a comparative experiment, in which the baseline retrieval system, Daffodil/HyREX, is contrasted with our home-grown XML element retrieval system, xmlfind.

The rest of the paper is organized as follows. Next, Section 2 documents the XML retrieval systems used in the experiment. Then, in Section 3, we detail the setup of the experiments. The results of the experiments are reported in Section 4, where we focus almost exclusively on the comparative experiment. Finally, in Section 5, we discuss our findings and draw some initial conclusions.

2 XML Retrieval Systems

2.1 Baseline System: Daffodil

The Daffodil system is developed to support the information seeking process in Digital Libraries [1]. As a back-end, the HyREX XML retrieval system was used [2]. For details, see [3].

Table 1. Experimental matrix for the comparative experiment.

#	Rotation	Task 1		Task 2		Task 3	
		Task	System	Task	System	Task	System
1	1	G-1	Daffodil	C-1	xmlfind	Own	choice
2	2	C-1	Daffodil	G-1	xmlfind	Own	choice
3	3	G-1	xmlfind	C-1	Daffodil	Own	choice
4	4	C-1	xmlfind	G-1	Daffodil	Own	choice
5	1	G-2	Daffodil	C-2	xmlfind	Own	choice
6	2	C-2	Daffodil	G-2	xmlfind	Own	choice
7	3	G-2	xmlfind	C-2	Daffodil	Own	choice
8	4	C-2	xmlfind	G-2	Daffodil	Own	choice
9	1	G-3	Daffodil	C-3	xmlfind	Own	choice
10	2	C-3	Daffodil	G-3	xmlfind	Own	choice
11	3	G-3	xmlfind	C-3	Daffodil	Own	choice
12	4	C-3	xmlfind	G-3	Daffodil	Own	choice
13	1	G-1	Daffodil	C-1	xmlfind	Own	choice
14	2	C-1	Daffodil	G-1	xmlfind	Own	choice

2.2 Home-grown System: xmlfind

The xmlfind system provides an interface for an XML information retrieval search engine [4]. It runs on top of a Lucene search engine [5]. The underlying index contains all the individual XML elements in the IEEE collection [6].

Figure 1(top) shows the search box and the result list. The results are grouped per article, where (potentially) relevant elements are shown. A partial view of the document tree, linking retrieved elements to the article root element, is shown. Small text excerpts, or text snippets or teasers, containing query words are generated to give a preview of the XML element's content. Clicking on any of the elements will open a new window displaying the result. Figure 1(bottom) shows the full article with the focus on the selected element. The results display window has three planes. On the left plane, there is a Table of Contents of the whole article. On the right plane, the article is displayed with the selected part of the document in view. On the top plane, the article's title, author, etc. are displayed, as well as a menu for assessing the relevance of the result (added specifically for the Interactive experiments reported in this paper).

3 Experimental Setup

The whole experiment was run in a single session where test persons for both Task A and Task B worked in parallel. The test persons were first year Computer Science students.

3.1 Task A: Community Experiment

Task A is the orchestrated experiment in which all teams participating in the Interactive Track take part [3]. We participated in Task A with six test persons,

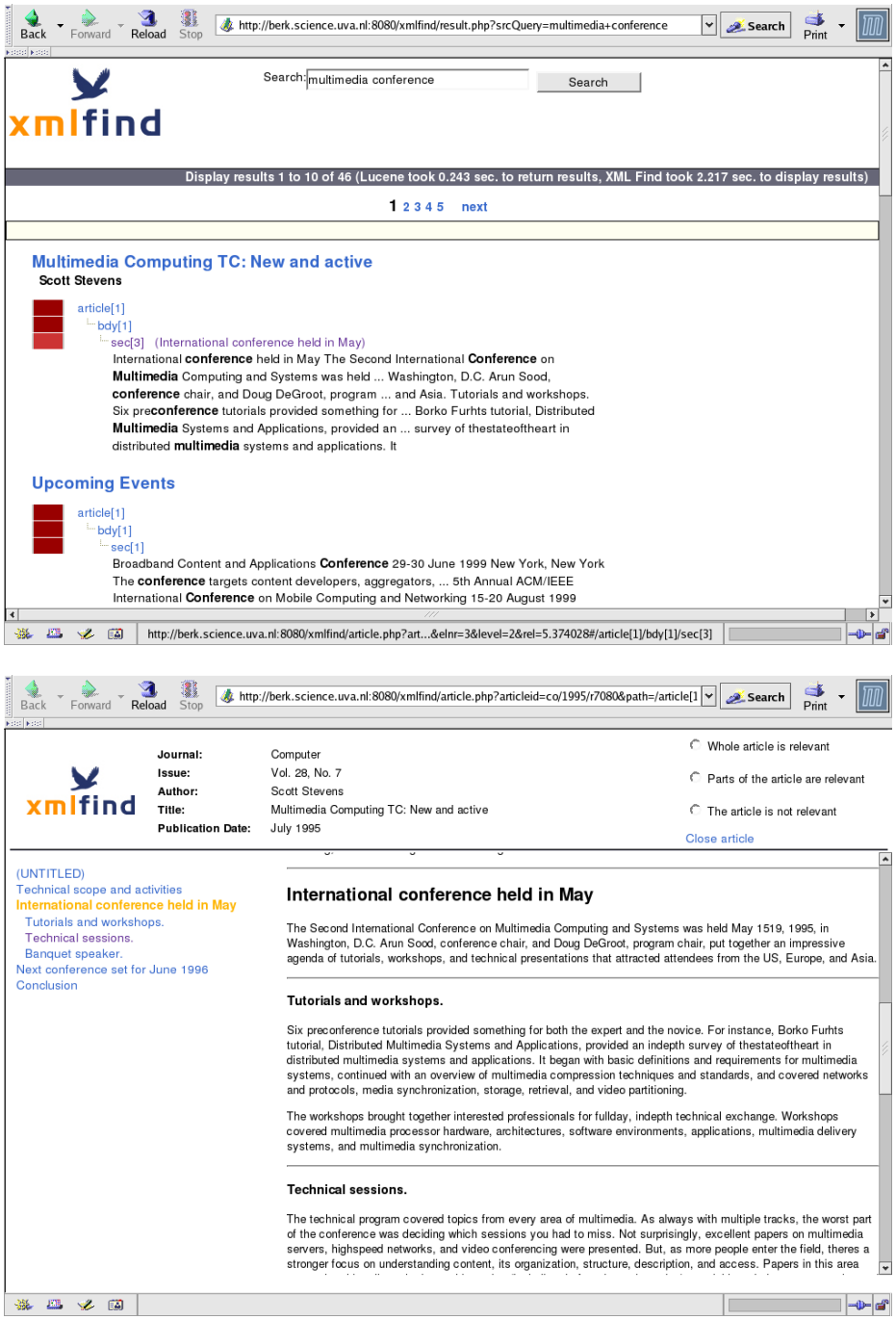

Fig. 1. Screen shots of xmlfind: (top) result list, (bottom) detailed view.

Table 2. Topic created by test person.

A. *What are you looking for?* Who build the first computer and what did it look like?
B. *What is the motivation of the topic?* I would like to know how the history of the computer began and what the first computer looked like, was it very big or very small, did it have a monitor?
C. *What would an ideal answer look like?* The name of the inventor and a picture of how the first computer looked.

who searched the IEEE Collection with the Daffodil/HyREX baseline system. There were three tasks: two simulated work tasks (a 'general' task and a 'challenging' task) and the test person's were asked to think up a search topic of their own. The experiment was conducted in accordance with the guidelines, for further details we refer to [3].

3.2 Task B: Comparative Experiment

Task B is a comparison of the home-grown xmlfind system with the Daffodil/HyREX baseline system. We participated in Task B with fourteen test persons. The experimental setup largely resembles the setup of Task A. Again, test persons did two simulated work tasks (a 'general' and a 'challenging' task) and they searched for a topic they were asked to think up themselves. The experimental matrix is shown in Table 1. Every test person searched for two simulated tasks, each one with a different system, following a standard two treatment matrix. Next, the test persons searched for their own topic with a system of their choice.

Due to the number of test persons involved, we were unable to conduct individual exit interviews. Instead, we used an extended post-experiment questionnaire.

4 Results

A large amount of data was collected during the experiments. Each test person searched with four different accounts, one for each task, plus one or two additional accounts for training. This generated in total 94 search logs (24 for Task A and 70 for Task B). In additional, each person filled in questionnaires before and after each task, and before and after the experiment, resulting in, in total, 160 questionnaires (48 for Task A and 112 for Task B). For a discussion of the results of Task A, we refer to the INEX 2005 Interactive track overview paper [3]. Here, we will focus on the results for the comparative evaluation in Task B.

4.1 Own topics

As part of the experiments, test persons were asked to think up a search topic of their own interest, based on a short description of the IEEE collection's content. Some topics created by test persons were excellent. Table 2 shows an example of a topic being (i) within the collection's coverage, (ii) reflecting a focused information need, and (iii) even containing potential structural retrieval cues. However, most topic were not so perfect. Even though test persons were asked to think up two different topics, almost half of the test persons (9 out of 20) did not create a very suitable topic. At least six topics addressed very practical advice on computer components or software, typically the sort of computer science related issues that users may search for on the web (targeting product reviews, FAQs or discussion boards). Examples of such created topics are *Latest video cards for best performance gaming* or *How to integrate .net applications in corporate environments*. Evidently, the IEEE Computer Society journals are not the most likely place to find relevant information for these topics. At least three topics were clearly outside the scope of the collection. Examples are *How many flights go from New York to Los Angeles a day?* and *How much energy does a rocket use to orbit?* Again, it is unlikely to find relevant information for these topics in the collection at hand. Perhaps more positively, the vast majority of the topics developed by our test persons were focused, asking for very specific information that could, in principle, be contained in a relatively short piece of text.

4.2 Information seeking behavior

During search, we logged the behavior of the test persons. Here, we will report on data from the xmlfind logs. In total, we have 24 sessions with xmlfind (see the experimental matrix in Table 1). In these 24 sessions, the test persons issued 91 queries in total, leading to an average of 3.8 queries per task. In the result list, a total of 172 elements were selected for further inspection. Note that this is, on average, only 1.9 per query, indicating that test-persons consulted only information from a very small number of articles. If we break down this number by the entry point into the article, we see that in 77 cases (44.8%) a test person selected an element, and in 95 cases (55.2%) an article was selected. That is, the test persons do use the option to deep-link particular XML elements in the articles. Finally, we asked the test persons, only once per viewed article, to give their assessment of its usefulness. We gathered 141 assessments in this way, which is 92.8% of all articles which were read in whole or in part. If we break down these judgments, we see that in 54 cases (38.3%) the article was regarded as not relevant, in 22 cases (15.6%) the whole article was regarded as relevant, and in the remaining 65 cases (46.1%) only parts of the article were regarded as relevant. Especially the last category, where relevant information is retrieved from an off-topic article, clearly demonstrates the potential of focused XML element retrieval techniques.

Table 3. Responses on user satisfaction: mean scores and standard deviations (in brackets). Answers were on a 5-point scale, ranging from 1 ("Not at all") to 5 ("Extremely").

	Q3.1	Q3.2	Q3.3	Q3.4	Q3.5
All tasks	3.4 (1.1)	3.0 (1.4)	3.1 (2.2)	3.2 (2.0)	3.6 (0.7)
First task	3.4 (1.2)	3.1 (1.1)	3.0 (2.5)	3.3 (1.9)	3.6 (0.4)
Second task	3.3 (1.5)	2.9 (1.5)	3.2 (1.6)	3.3 (1.3)	3.6 (0.4)
First two tasks	3.4 (1.1)	3.0 (1.1)	3.2 (2.1)	3.3 (1.6)	3.6 (0.4)
General task	3.4 (1.2)	2.9 (1.1)	3.1 (2.2)	3.1 (2.1)	3.9 (0.4)
Challenging task	3.4 (1.2)	3.1 (1.2)	3.3 (2.1)	3.5 (1.2)	3.4 (0.3)
Own task	3.4 (1.2)	3.0 (2.0)	3.1 (2.7)	3.1 (2.8)	3.5 (1.3)
Daffodil (task C and G)	3.1 (0.7)	2.7 (0.5)	3.1 (2.1)	3.1 (1.8)	3.6 (0.3)
xmlfind (task C and G)	3.6 (1.5)	3.4 (1.6)	3.3 (2.2)	3.5 (1.5)	3.7 (0.5)
Daffodil (first task)	3.0 (0.6)	3.0 (0.6)	3.3 (3.1)	3.4 (2.6)	3.8 (0.2)
Daffodil (second task)	3.2 (1.0)	2.3 (0.3)	2.8 (1.0)	2.8 (1.0)	3.3 (0.3)
xmlfind (first task)	4.0 (0.8)	3.5 (1.1)	3.0 (2.8)	3.3 (1.9)	3.5 (0.7)
xmlfind (second task)	3.4 (2.0)	3.3 (2.2)	3.5 (2.0)	3.6 (1.4)	3.9 (0.4)

4.3 Appreciation of the searching experience

After each completed task, test persons filled in a questionnaire. There were a number of questions on the testperson's satisfaction:

Q3.1 *Was it easy to get started on this search?*
Q3.2 *Was it easy to do the search on the given task?*
Q3.3 *Are you satisfied with your search results?*
Q3.4 *Do you feel that the task has been fulfilled?*
Q3.5 *Do you feel that the search task was clear?*

Table 3 shows the responses of the test persons. First, we look at the responses over all sessions. The test persons are fairly positive with average results in the range 3.0 to 3.6. Second, we look at responses for the different tasks. Here we see that the reponses for the first and second task are comparable, and in sync with the overall responses. The third task was always the Own task. When we look at the responses for the different task types, General, Challenging, or Own, we see a similar pattern as for the two simulated work tasks. Interestingly, the General task is regarded as clearer (Q3.5), but the search results for the Challenging task are valued higher (Q3.3 and Q3.4). The responses for Own task are surprizing: although formulated by the test person herself, they are not regarded as clearer (Q3.5). The responses for the Own task are, on average, similar to the simulated work tasks. The standard deviation, however, is much larger. The reason for this seems to be the inability of a large fraction of test persons to come up with a topic that is suitable for the collection at hand. Third, we look at the responses for the different search engines, focusing on the simulated work tasks where a proper matrix was used. Over all sessions with the search engines, xmlfind was regarded as easier to use (Q3.1 and Q3.2), and more effective (Q3.3 and Q3.4)

Table 4. Responses on searching experience: mean scores and standard deviations (in brackets). Answers were on a 5-point scale, ranging from 1 ("Not at all") to 5 ("Extremely").

	Q3.9	Q3.10	Q3.11	Q3.12	Q3.13
All tasks	3.2 (1.4)	3.0 (1.3)	3.3 (1.0)	3.4 (1.4)	3.4 (1.3)
First task	3.1 (1.8)	3.0 (1.4)	3.2 (1.3)	3.4 (1.6)	3.6 (1.6)
Second task	3.0 (1.4)	3.0 (0.8)	3.4 (0.9)	3.3 (1.0)	3.5 (0.7)
First two tasks	3.1 (1.5)	3.0 (1.0)	3.2 (0.9)	3.4 (1.3)	3.4 (1.3)
General task	2.8 (1.6)	2.6 (1.0)	3.2 (1.1)	3.1 (1.6)	3.4 (1.5)
Challenging task	3.4 (1.3)	3.4 (0.7)	3.2 (0.8)	3.6 (1.0)	3.4 (1.2)
Own task	3.5 (1.3)	3.0 (2.0)	3.5 (1.2)	3.5 (1.5)	3.2 (1.3)
Daffodil (task C and G)	2.9 (1.5)	2.8 (1.1)	3.2 (0.8)	3.4 (1.5)	3.3 (1.8)
xmlfind (task C and G)	3.2 (1.6)	3.2 (1.0)	3.2 (1.1)	3.3 (1.3)	3.6 (0.9)
Daffodil (first task)	3.1 (1.6)	2.9 (1.3)	3.3 (0.8)	3.6 (2.3)	3.4 (2.6)
Daffodil (second task)	2.7 (1.5)	2.7 (1.1)	3.2 (1.0)	3.2 (0.6)	3.2 (1.0)
xmlfind (first task)	3.2 (2.2)	3.2 (1.8)	2.8 (1.4)	3.2 (1.4)	3.3 (1.5)
xmlfind (second task)	3.3 (1.4)	3.3 (0.5)	3.5 (0.9)	3.4 (1.4)	3.8 (0.5)

than Daffodil. We also look at whether earlier experience with the other search engine did influence the responses. We see that responses for the first task, either using Daffodil or using xmlfind, are much closer; Daffodil gets higher scores on effectiveness (although the standard deviation is large). However, we see that test persons that used Daffodil for the first task, were more positive than those that used Daffodil for the second task (after searching with xmlfind for the first task). Conversely, the test persons that used xmlfind for the second task (after using Daffodil for the first task), were more positive than those that used xmlfind for the first task.[3]

The questionnaire also contained a number of questions on the search experience of the test persons:

Q3.9 How well did the system support you in this task?
Q3.10 On average, how relevant to the search task was the information presented to you?
Q3.11 Did you in general find the presentation in the result list useful?
Q3.12 Did you find the parts of the documents in the result list useful?
Q3.13 Did you find the Table of Contents in the Full Text view useful?

Table 4 shows the responses, using a similar breakdown as before. First, we look at responses over all sessions. The test persons are again fairly positive with averages ranging from 3.0 to 3.4. Second, we look at responses for the different tasks. Responses for the first and second simulated work task are very

[3] Here, we compare the responses of different test persons, and hence it may be the case that test persons starting with Daffodil were simply more positive than those starting with xmlfind. Note, however, that the group starting with Daffodil gave higher scores to xmlfind in the second task, and the group starting with xmlfind gave Daffodil lower scores in the second task.

Table 5. Responses on the system comparison: mean scores and standard deviations (in brackets). Answers were on a 5-point scale, ranging from 1 ("Not at all") to 5 ("Extremely"). Statistical significance is based on a paired t-test (two-tailed).

	Q4.4	Q4.5	Q4.6
Daffodil	3.1 (0.9)	2.9 (1.1)	3.4 (0.6)
xmlfind	4.2 (0.8)	4.2 (0.3)	4.2 (0.3)
Significance	$p < 0.01$	$p < 0.001$	$p < 0.05$

similar to the overall responses. When we look at the three task types, we see that the responses for the General task deviate for system support (Q3.9) and relevance (Q3.10). Perhaps suprizingly the systems are more appreciated for the Challenging task than for the General task. Responses for the Own task, always searched after the two simulated work tasks, do not differ much from the overall responses. Third, we look at responses for the diffent systems. We see that both systems receive comparable scores on the presentation issues (Q3.11, Q3.12, and Q3.13). There is, however, a marked difference in the responses for support (Q3.9) and relevance (Q3.10), where xmlfind is prefered over Daffodil. When looking at the interaction between the search experience for both systems, we see, again, that earlier exposure to xmlfind leads to lower scores for Daffodil, and earlier exposure to Daffodil leads to higher scores for xmlfind.

4.4 Comparative Evaluation

Test persons in Task B were free to select with which of the two system they searched for the third topic. Out of the 14 test persons, 4 (28.6%) choose to search with the Daffodil/HyREX system, the other 10 (71.4%) choose to search with the xmlfind system.

In the post-experiment questionnaire, each test person was asked a number of questions about the two systems that they used:

Q4.4 *How easy was it to learn to use the system?*
Q4.5 *How easy was it to use the system?*
Q4.6 *How well did you understand how to use the system?*

Table 5 shows the responses of the test persons. We see that the test persons give a significantly higher score to xmlfind with respect to the easiness to learn (Q4.4), the easiness to use (Q4.5), and the understandability of the system (Q4.6).

4.5 General Views

As part of the extended post-experiment questionnaire, test persons in Task B were asked a number of questions about their opinions on the concept of an XML retrieval engine. Table 6 lists the responses to two of the questions, where each row represents the same test person. The responses where unequivocally positive, and the responses highlight many of the hoped advantages of an XML retrieval system.

Table 6. Responses on the usefulness of focused retrieval.

13. *Did you like the idea that the search engine takes into account the structure of the documents? Why?*	14. *Do you find it useful to be pointed to relevant parts of long articles? Why?*
Yes, you will have a good overview of the total article/document.	Yes, because you are able to see which articles are worth reading and which are not.
Yes, for specific information this is very useful.	Yes, gives the user an idea about the article in question.
Yes, easier to see how long the article is.	You don't need to see other parts.
Yes, its easier to see the contents of the document, better navigation.	Yes, you don't have to dig into the article yourself.
Yes, it didn't bother me.	Yes, it's more easy to find what you're looking for.
Yes, less reading time, clear overview.	Yes, saves time.
Yes, it shortens search time.	Yes, because if scan-read long articles, you easily miss some relevant parts.
Yes, saves work.	Yes, works faster.
Yes, because its much faster.	Yes, its faster.
Yes, this way of finding information takes less time.	Yes, now you don't have to read the whole article. You can get straight to the part where the information is.
Yes, its easier to see where relevant information is located.	Yes, it takes less time to find the relevant parts.
Yes, it makes it easier to find specific paragraphs.	Yes, if programmed right it can save time.
Yes, it makes it a lot easier to find what you are looking for.	Yes, it is lots more easier.
Yes, because makes me have to search less.	Yes, to search less.

5 Discussion and Conclusions

This paper documents the University of Amsterdam's participation in the INEX 2005 Interactive Track. We participated in two tasks. First, we participated in the concerted effort of Task A, in which a common baseline system, Daffodil/-HyREX, was used by six test-persons to search the IEEE collection. Second, we conducted a comparative experiment in Task B, in which fourteen test persons searched alternately with the baseline retrieval system, Daffodil/HyREX, and our home-grown XML element retrieval system, xmlfind.

We detailed the experimental setup of the comparative experiment. Both experiments, involving twenty test persons in total, were conducted in parallel in a single session. This ensured that the experimental conditions for all test persons are very equal. Unplanned external causes, such as the down-time of the Daffodil/HyREX system equally affected all test persons. Due to the large number of test persons present at the same time, we had to minimize the need for experimenter assistance. This was accomplished by generating personalized protocols for all test persons. In these protocols, test persons were guided through the ex-

periment by means of verbose instructions on the transitions between different tasks. Four experimenters were available, if needed, to clarify the instructions or provide other assistance. This worked flawlessly, and allowed us to handle the large numbers of test persons efficiently.

A large amount of data was collected during the experiments, both in questionnaires and in search log files. In this paper we focused mainly on the results of the comparative experiment. As for the comparison between the Daffodil/HyREX system and the xmlfind system, we see that the test persons show appreciation for both systems but that xmlfind receives higher scores than Daffodil. It is difficult to pin-point what's the deciding factor in the system comparison, in the questionnaires the ease of use, the speed and stability, and the quality of the search results are mentioned by test persons.

Over the whole experiment, perhaps the most striking result is that some expected problems did not surface in the questionnaires. Note that the xmlfind system retrieves potentially overlapping elements, and that in the result list even all ascendants of found elements are added. Hence, one might have expected the so-called overlap problem that plagues XML retrieval metrics [7] to rear its head. For example, in the Interactive track at INEX 2004 test persons complained about encountering partly overlapping results scattered through the ranked list of elements [8,9]. Clustering found elements from the same article seems to be an effective way for an interface to deal with the structural dependencies between retrieved elements.

The general opinion on the XML retrieval systems was unequivocally positive. Departing from earlier systems that return ranked lists of XML elements, both the Daffodil/HyREX and xmlfind group the found XML elements per article (similar to the Fetch & Browse task in the Ad hoc Track). Test persons seem to conceive the resulting system as an article retrieval engines with some additional features—yet with great appreciation for the bells and whistles!

Acknowledgments

This research was supported by the Netherlands Organization for Scientific Research (NWO) under project numbers 017.001.190, 220-80-001, 264-70-050, 354-20-005, 612-13-001, 612.000.106, 612.000.207, 612.066.302, 612.069.006, 640.001.-501, and 640.002.501.

References

1. Daffodil: Distributed Agents for User-Friendly Access of Digital Libraries (2006) http://www.is.informatik.uni-duisburg.de/projects/daffodil/.
2. HyREX: Hyper-media Retrieval Engine for XML (2006) http://www.is.informatik.uni-duisburg.de/projects/hyrex/.
3. Larsen, B., Malik, S., Tombros, T.: The interactive track at INEX 2005. In: This Volume. (2006)
4. Bakker, T., Bedeker, M., van den Berg, S., van Blokland, P., de Lau, J., Kiszer, O., Reus, S., Salomon, J.: Evaluating XML retrieval interfaces: xmlfind. Technical report, University of Amsterdam (2005)

5. Lucene: The Lucene search engine (2006) http://lucene.apache.org/.
6. Sigurbjörnsson, B., Kamps, J., de Rijke, M.: An Element-Based Approch to XML Retrieval. In: INEX 2003 Workshop Proceedings. (2004) 19–26
7. Kazai, G., Lalmas, M., de Vries, A.P.: The overlap problem in content-oriented XML retrieval evaluation. In: Proceedings of the 27th Annual International ACM SIGIR Conference, ACM Press, New York NY, USA (2004) 72–79
8. Tombros, A., Larsen, B., Malik, S.: The interactive track at INEX 2004. In: Advances in XML Information Retrieval. Third Workshop of the INitiative for the Evaluation of XML Retrieval, INEX 2004. Volume 3493 of Lecture Notes in Computer Science., Springer Verlag, Heidelberg (2005) 410–423
9. Tombros, A., Larsen, B., Malik, S.: Report on the INEX 2004 interactive track. SIGIR Forum **39** (2005) 43–49