NEXI, Now and Next

Andrew Trotman
Department of Computer Science
University of Otago
Dunedin, New Zealand
andrew@cs.otago.ac.nz

Börkur Sigurbjörnsson
Informatics Institute
University of Amsterdam
Amsterdam, The Netherlands
borkur@science.uva.nl

ABSTRACT

NEXI was introduced in INEX 2004 as a query language for specifying structured and unstructured queries on XML documents. A language expressive enough for INEX yet simple enough for users to get right. These goals have been achieved. In particular, the error rate in CAS queries has dropped from 63% in 2003 to 12% in 2004. This drop is shown to be a consequence of not only the language, but the tools introduced with it: the source code for a parser was downloaded by 13 IP addresses, while a web implementation was accessed 635 times from 71 addresses.

Although NEXI is suitable for the *ad hoc* track, it is not sufficiently expressive enough for the heterogeneous track, or for question answering. The syntax necessary to extend to these purposes is proposed. This includes weighted terms and weighted paths. The new syntax is strictly an extension so does not invalidate any existing queries.

1. INTRODUCTION

Each of the first three INEX [4] workshops used a different query language. At the first workshop queries were specified in XML [6], at the second in XPath [7], and at the third in NEXI [14]. This succession of languages occurred because, as a consequence of each workshop, new and different query types, and how to specify them, have become clear.

The first INEX workshop was modeled on TREC, and consequently a TREC-like topic format was chosen. Topics were broken into four parts, title, description, narrative and keywords. Of these, the title contained the IR query, and is consequently of focus. For Content Only (CO) queries, the title was a two or three word description of the topic. For Content And Structure (CAS) queries, the title was further marked up in XML. The optional <te> tag was used to specify target elements for the search, while <cw> was used to identify content words that were optionally associated with a container element, <ce>.

```
<Title>
  <te>tig</te>
  <cw>QBIC</cw><ce>bibl</ce>
  <cw>image retrieval</cw>
</Title>
```

Figure 1: An INEX 2002 query fragment (INEX topic 05).

An example query, the title element from INEX topic 05 is given in Figure 1. In this example, the user is searching for documents that contain the phrase "image retrieval", contain the word QBIC in a <bibl> element, and asking for <tig> elements to be retrieved. It was quickly established that this query language was insufficient for the need [11].

First, the XML format allowed the user to specify queries that were simple mechanical processes. In the above example, once relevant documents have been identified, the process of extracting the <tig> (or title group) is mechanical. There is one, and only one, <tig> element in each document. Identifying and extracting it can be done with a simple text search.

Second, the language was not expressive enough. The target element was specified irrespective of the context of the query. It was not possible to specify a query of the nature "find sections about sunny New Zealand"; the nearest such query was "find sections from documents about sunny New Zealand" – two quite different queries.

For the second workshop XPath [1] was adopted in the hope it would alleviate these problems, and it did. With the addition of a function for ranked information retrieval (*about*), and the elimination of non-IR functions (e.g. *contains*) XPath proved sufficiently expressive.

XPath introduced new problems! O'Keefe and Trotman [10] provide an analysis of the failure of XPath as a query language for INEX. Perhaps the most damming evidence is the error rate in the official topics. Of the 30 CAS topics, 19 contained errors; that is a 63% error rate in queries written by IR experts.

Subsequently, the INEX 2003 Queries Working Group identified the requirements for a query language suitable for INEX [13]. In brief, it had to look like XPath, be easier to use, and oriented to IR.

Considerable effort was spent defining the query language NEXI [14], used at the 2004 workshop. Designed with the sole purpose of satisfying the requirements of INEX (and the Queries Working Group), this language is a simplified XPath containing only the descendant axis; while at the same time an extended XPath containing the *about* function. NEXI is in use at the current (2004) workshop.

The use of NEXI within and without INEX is examined. From this, the conclusion is drawn that it has successfully proven to be a suitable language for XML retrieval. Future requirements are examined, and extensions are proposed. Adoption of these extensions is recommended.

2. CURRENT STATE OF PLAY

The *ad hoc* track at INEX consists of two tasks, the Content Only (CO) and Content and Structure (CAS) tasks.

In the CO task, it is the task of the search engine to identify relevant document elements that satisfy a user query. By definition, the query does not specify where to look, or what elements to retrieve. A CO query is a sequence of terms, and example of which is INEX Topic 37: "temporal database queries and query processing". For this query, the search engine is expected to identify and return a relevance ranked list of document elements about temporal database queries and query processing.

There are two variants of the CAS task, the Strict CAS (SCAS)[1] and the Vague CAS (VCAS). The queries for both are the same; it is only the interpretation that differs – the reader is referred to Fuhr, Malik, and Lalmas [3] for details. In a CAS query, structural elements are included in the query. If a user wishes to find document abstracts that discuss INEX, it is necessary to specify <abstract> as the target element. If a user is searching for smith, but knows they want Dr. Smith and not an ironmonger, they may specify that Smith is an author.

The Queries Working Group at INEX 2003 [13] identified the requirements of a query language necessary to satisfy CAS queries within the context of INEX. In brief, that language must:

- Be a subset of XPath, so as to be familiar to the XML community. Tag instancing was removed, axes were limited to only the descendant axis, filters remained but the not-equals operator was not permitted with string types.
- Support multiple data types. String and numeric types were specified. XPath filters remained, but a restricted set of operators was included.
- Be vaguely interpretable. It must be an IR language. To this end, the AND operator and OR operator were specified as ANDish and ORish.
- Specify one and only one target element (shown below to have been violated).

Additionally, this language allowed the specification of CO queries. It was also specified as extensible.

Trotman and Sigurbjörnsson [14] proposed NEXI, an IR query language for XML that satisfied the requirements of the Working Group and was subsequently adopted for the 2004 INEX. They also provided the source code to a parser, and for INEX 2004 an on-line parser.

2.1 Query Errors at INEX 2004

Examining the first release of the topics for 2004 (version 2004-01), 4 of the 34 CAS queries contain errors (12%). In the CO queries 6 of 39 contain errors (15%). The error rate in CAS is now lower than that in CO.

2.1.1 Examining CAS errors:

Topics 137 and 158 were missing a close bracket at the end of the query. There are corrected by appending '`]`'.

Topic 138 contained the incorrect expression "`about(.,//sec,thread implementation)`" which is incorrect in the first comma. This is corrected by removing the erroneous comma.

Topic 161 contained the incorrect expression "`about(./atl, database access methods)`" which is incorrect in so far as it uses the child axis. This is corrected by replacing "/" with "//".

2.1.2 Examining CO errors:

Topics 176, 177, and 196 contained illegal punctuation. This is corrected by removing the punctuation.

Topic 190 contained the quoted expression "`"e-commerce"`" which, as the hyphen makes e-commerce a single word, is a single word phrase. Phrases consist of strictly more than one word so this is erroneous. This is corrected by removing the quotes.

Topics 178 and 179 contain phrases delimited with question mark characters "`?`". This is corrected by replacing those characters with quotes.

2.2 Online Parser

In 2004 an online query syntax checker was introduced. Use was logged, with accesses from the University of Otago stripped (to avoid skewing by the developers). Logs were analyzed for the period April 12th through to October 26th; between the date when the parser went online, and when analysis began. Table 1 shows the number of times the parser was accessed each month.

There was a total of 635 requests on 37 distinct dates from 71 internet addresses. Most of the requests occurred during April and May. The topic submission date was May 7th. In Figure 1, the cumulative number of requests on each day of activity is shown. There is a clear burst of activity around the submission date, and finishing on 11th May. Activity immediately after submission date may be caused by late submissions.

Table 1: Parse requests to the online NEXI parser

Month	Requests
April	167
May	447
June	4
July	3
August	5
September	9

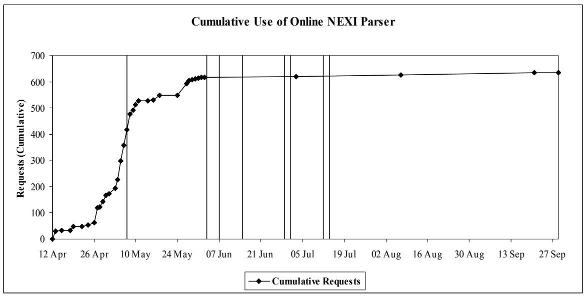

Figure 1: Cumulative use of the online NEXI parser shows considerable use between April 27th and May 11th. The topic submission date was May 7th. Vertical lines are shown for the topic submission date, and each revision date.

[1] At INEX 2004, SCAS was deprecated

After the submission date, but before the first release of the topic set, there was a clear burst of activity (18th through 28th May), this is likely to be the period in which topics were corrected. There was very little activity during the period in which the topic set was under revision, with only 3 requests between the first release (version 2004-001) and the final release (version 2004-07).

It is hard to account for activity in August and September. The requests were valid and the authors are using the parser for the purpose in which it was designed (users are not hacking the parser).

The parser was in the New Zealand time zone, whereas a time-zone for the due and release dates was not given. Requests from the University of Otago were removed from the logs before analysis.

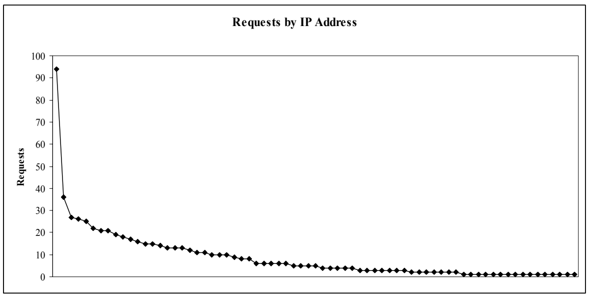

Figure 2: Number of requests from each IP address in decreasing order.

Figure 2 shows the number of requests for each accessing IP address. The number of requests ranged from 94 to 1. The 94 accesses appears to be an outlying point; with the next highest accesses being 36 and then 27 requests. The mean number of requests per address was 8.9, the median being 5.

No effort has been spent trying to resolve IP addresses to institutions; doing so is likely to decrease the number of addresses and increase the mean and median.

2.3 Was NEXI Successful?

The initial error rate in queries has dropped from 63% in 2003 to 12% in 2004. The error rate for CAS topics is now about the same as that in CO topics. The number of topic revisions has halved. From this is would be reasonable to conclude changes made between 2003 and 2004 had a marked effect on syntactic correctness of queries. Those changes were not, however, limited to query language changes.

First, the queries submitted to INEX were checked for syntax errors as part of the selection process. This bias, although present, is not a major contributing factor. Of the originally submitted 84 CAS queries, 18 (21%) contained errors, whereas of the 107 CO topics, 19 (18%) contained errors. These two error rates are about equal. The error rate in the original submissions in 2003 is not known.

Second, having written XPath parsers for 2003, the participants themselves should have been familiar with the language, and therefore more able to write syntactically correct queries than before.

Third, web access to an online parser was made available during the topic development period. This has, no doubt, had an effect on the correctness of the submitted queries.

Fourth, the source for a command line version of the parser was attached to the language specification; and downloadable from the web site. It was downloaded by 13 IP addresses; discussion with some INEX participants suggests it was also used.

The decrease of errors in CAS topics is considered a sign of NEXI success; however, there are still areas that need addressing. During 2003, the topics underwent 12 revisions over a period of 38 days. In 2004, it took only 7 revisions, but 41 days. One can but hope that in future years topics are submitted correctly and on time.

3. THE FUTURE

NEXI was, by design, the simplest query language that could possibly work. The subset of XPath was chosen in order to ensure nothing unnecessary was included. To this end, NEXI has proven a success for *ad hoc* searching, but only for *ad hoc* searching – it has proven unsuitable for other types of search. This shortfall is now addressed with additions for question answering, heterogeneous searching, and a new wildcard.

3.1 Wildcards

The NEXI path wildcard operator, *, is defined as meaning “first or subsequent descendant” [14]. A new “here or below” wildcard, +, is introduced, but it is of limited use.

As //article//+ means “article or below”, //+ must mean “nothing or below”. This nothingness is meaningless, as there must be at least one element present. Specifying the existence of one or more elements is done with //*. Use of //+ is therefore prohibited.

Use of two or more adjacent //+ operators is meaningless; //article//+ and //article//+//+ are semantically equivalent. The two forms //article//+//bm and //article//bm are also equivalent. Use of the + inside a path is meaningless as it simply specifies there might be a node, which is implicit in the descendant operator.

There exists only one place this new operator can be used; the end of a path specification. The form //*//+ is redundant, and equivalent to //*, further restricting the use of +.

The new addition to the path syntax is:

```
zero_any_node: NODE_QUALIFIER '+'
```

which requires the following changes:

```
path: any_node
    | node_sequence
    | node_sequence any_node
    | node_sequence attribute_node
    | node_sequence any_node attribute_node
    | node_sequence zero_any_node

node_sequence: node
    | node_sequence node
    | node_sequence any_node node
```

```
        | node_sequence any_node any_node

    node: named_node | tag_list_node
```

It is unfortunate that the late addition of the + wildcard operator results in * meaning one or more and + meaning zero or more because these two operators have each other's definition in regular expressions.

Strict interpretation: "//A//+" means at or below the "//A" element.

Loose interpretation: "As paths are only hints, feel free to ignore this"

3.2 Multiple Target Elements

The tag list syntax, "//(A|B)" means "either the A or the B element". As this syntax is not forbidden as the target element, it might be exploited by a topic author to identify multiple target elements. This use, although valid, is discouraged.

3.3 NEXI for Question Answering

There is currently no question answering track at INEX, however the authors anticipate there being so. Ogilvie [9] has already discussed the inadequacies of NEXI to fulfill this role. We concede, it was not designed for this purpose and does not fulfill the role. Ogilvie does, however, propose syntax for the purpose.

In place of an *about* function, Ogilvie suggests a *weight* function; which he gives by example:

```
    //sentence[.//event//VBD[weight(0.4 kill 0.3
    assassinate 0.2 murder 0.1 shoot)] AND
    .//patient//person[weight(0.4 'Abraham
    Lincoln' 0.4 'President Lincoln' 0.1 'honest
    Abe' 0.1 Lincoln)]]//agent//person
```

weight differs from *about* in three ways. First, phrases are specified using single quotes in place of double quotes. Second, the path occurs outside the clause rather than inside it. Third, weights for each term are given. Altering the *weight* to resemble *about* results in:

Example:

```
    //sentence[weight(.//event//VBD, 0.4 kill 0.3
    assassinate 0.2 murder 0.1 shoot) AND
    weight(.//patient//person, 0.4 "Abraham
    Lincoln" 0.4 "President Lincoln" 0.1 "honest
    Abe" 0.1 Lincoln)]//agent//person
```

the formal syntax of which is:

```
    decimal: NUMBER | NUMBER '.' NUMBER

    WEIGHT: "weight"
    weighted_co: decimal term
        | weighted_co decimal term

    weight_clause: WEIGHT '(' relative_path ','
        weighted_co ')'
```

additionally, the definition of filter is altered to:

```
    filter: about_clause
        | weight_clause
        | arithmetic_clause
```

Strict interpretation: "In the example, only a //sentence//agent//person element is correct, that said, it will most likely tell me who killed honest Abe".

Loose interpretation: "What I want is most likely a //sentence//agent//person element that will tell me who assassinated honest Abe. I know several ways of saying assassinate, and honest Abe, here are some and how likely I think you are to see them – but I might be wrong about this".

3.3.1 QA Paths

Ogilvie notes that path semantics may require relaxation for Question Answering. The paths may, instead, refer to a structural annotation of the document content. In no way should NEXI be interpreted as prohibiting any such interpretation of paths – this is the loose interpretation embraced.

3.4 NEXI for Heterogeneous Searching

The heterogeneous track chose a subset of topics from the *ad hoc* track, and added to them some special purpose topics. Of the chosen topics, 161 and 196 contained errors (discussed above). In version 2 of the heterogeneous topics there are 4 added topics, one of which contains spurious punctuation (topic 4). Topics should be checked for syntax errors before inclusion in any topic list.

The heterogeneous track has four types of queries, Content Only (CO), Basic CAS (BCAS), Complex CAS (CCAS) and Extended CCAS (ECCAS).

This year CO topics from the *ad hoc* track were used for the heterogeneous track. As the IEEE collection is part of the heterogeneous collection, this decision avoids any additional relevance assessing on that collection. Consequently, all CO topics in the heterogeneous track are already in NEXI.

Basic CAS topics contain one structural constraint and one textual constraint. They can all be specified in the form

```
    //constraint[about(., content)]
```

where constraint and content are single terms. This is a subset of NEXI which was, consequently, chosen for specifying BCAS topics.

Compex CAS topics are the heterogeneous equivalent of *ad hoc* CAS topics. They are in the form //A[B] or //A[B]//C[D]. CCAS topics are specified in NEXI.

Extended Complex Content and Structure (ECCAS) topics allow the query author to specify a belief in the correctness of a structural constraint. The example given in the track guidelines [2] is:

```
//author(0.8)[about(title(0.5), 'Information
Retrieval')],
```

in which the user has an 80% certainty the answer is an author element, thinks the article will be about information retrieval, but has only a 50% certain that this will be discussed in the title. There were no ECCAS topics submitted and NEXI did not support syntax for them.

ECCAS topics are expected in future years. To this end, syntax supporting user certainty in tag specification is needed. Extending NEXI would require only small changes from the syntax proposed in the heterogeneous track guidelines.

First, in NEXI phrases are specified using double quotes, phrases in ECCAS should be specified in the same way. Second, paths in a NEXI *about* function are relative to the context path (the path being filtered) but in the example given in the heterogeneous track guidelines [2], the path is an absolute path. The change to absolute paths prevents the specification of queries that can be resolved through a mechanical process, however it also restricts the expressiveness of the query – these kinds of queries can't be written. This tradeoff is considered acceptable.

The syntax requires only small changes:

```
weight: '(' decimal ')'

tag: XMLTAG | XMLTAG weight
```

Strict interpretation "//A(0.5)" is a 0.5 certainty in the correctness of "//A" for the purpose in which it is being used. "//A(0.5)//B(0.3)" is a 0.3 certainty of "//A//B" for its purpose and a 0.5 certainty in "//A" for its purpose. In the expression "//(A(0.2) | B(0.5))", the certainty of being "//A" is given along with the certainty of "//B". The certainty values are only hits, and are open to interpretation.

Loose interpretation "I'm not sure where to look, these places might be good"

3.5 Uncertain NEXI

The heterogeneous additions combined with the question answering additions provide the syntax necessary for certainty of path and certainty of search term combinations. A query of this nature can be considered super-loose or utterly uncertain; the user is uncertain of everything (a THISish search?).

Example:

```
//bb(0.3)[weight(., 0.2 "Information
Retrieval")]
```

Strict interpretation: There is no strict interpretation.

Loose interpretation: "The answer is probably a <bb> element, and it probably says something about Information Retrieval, but I'm not certain about this"

3.6 Relevance Feedback NEXI

In relevance feedback it is not uncommon to add additional search terms or to weight search terms. The natural analogue for structured searching is adding paths and weighting paths. Syntax for both weighting terms and paths is suggested above. Here the applicability to relevance feedback is identified.

4. OTHER NEXI RELATED WORK

Kamps *et al*. [5] suggest adding the ancestor axis to NEXI. They call this superset Positive Temporal XPath. Although this syntax is not more expressive (all queries specifiable in Positive Temporal XPath can be expressed in NEXI), they suggest specifying a path from child to parent is more natural to some users than *vice versa*. They conjecture that paths specified using both ancestor and descendant may be more succinct than using just one or the other.

It is unfortunate that some users prefer parent to child, while others prefer child to parent; using one or the other is simpler than using either or both. In an effort to remain simple, the introduction of an ancestor axis to NEXI is left as future work.

Mihajlović *et al.* [8] choose to store the INEX collection in a relational database. Between the relational database and NEXI they introduce an algebra. With this approach it is possible to change (and experiment with) the underlying relational structure independent of the algebraic optimization of query expressions. It also allows the introduction and optimization of XML IR operators such as *about*. They choose the range approach for searching structured documents and consequently their introduced algebra is an algebra of regions. Piwowarski and Gallinari [12] prefer a probabilistic implementation and introduce a probabilistic algebra for a subset of XPath which is a superset of NEXI.

5. CONCLUSIONS

NEXI has proven to be successful for INEX. This success is due to a combination of the simple XPath like syntax, the online parser, and the command-line parser. The online parser was used a total of 635 times from 71 IP addresses, the command line parser was downloaded from 13 IP addresses. As a consequence of this use the error rate in CAS queries dropped from 63% in 2003 to 12% in 2004.

Although NEXI has proven suitable for *ad hoc* retrieval, it has also proven inadequate for question answering and heterogeneous searching. New syntax is added for these purposes. In essence, this new syntax adds weighted paths and weighted search terms. These extensions might also be used for relevance feedback.

Wildcards in paths are extended to include a zero or more descendants wildcard, +. The new wildcard is meaningless except at the end of a path.

The adoption of the extensions proposed herein will allow tracks in addition to *ad hoc* to use NEXI. This use, and continued use in the *ad hoc* track, is recommended.

6. REFERENCES

[1] Clark, J., & DeRose, S. (1999). XML path language (XPath) 1.0, W3C recommendation. The World Wide Web Consortium. Available: http://www.w3.org/TR/xpath [2004.

[2] Dignum, V., & Zwol, R. v. (2004). Guidelines for topic development in heterogeneous collections. Available:

http://inex.is.informatik.uni-duisburg.de:2004/internal/hettrack/downloads/hettopics.pdf.

[3] Fuhr, N., Malik, S., & Lalmas, M. (2003). Overview of the initiative for the evaluation of XML retrieval (INEX) 2003. In *Proceedings of the INEX 2003 Workshop*, (pp. 1-11).

[4] Gövert, N., & Kazai, G. (2002). Overview of the initiative for the evaluation of XML retrieval (INEX) 2002. In *Proceedings of the 1st Workshop of the INitiative for the Evaluation of XML Retrieval (INEX)*, (pp. 1-17).

[5] Kamps, J., Marx, M., Rijke, M. d., & Sigurbjörnsson, B. (2004). Best-match querying for document-centric XML. In *Proceedings of the 7th International Workshop on the Web and Databases (WebDB 2004)*, (pp. 55-60).

[6] Kazai, G., Lalmas, M., & Malik, S. (2002). INEX guidelines for topic development. In *Proceedings of the 1st workshop of the initiative for the evaluation of XML retrieval (INEX)*, (pp. 178-181).

[7] Kazai, G., Lalmas, M., & Malik, S. (2003). INEX '03 guidelines for topic development. In *Proceedings of the 2nd workshop of the initiative for the evaluation of XML retrieval (INEX)*.

[8] Mihajlovic, V., Hiemstra, D., Blok, H. E., & Apers, P. M. G. (2004). An XML-IR-db-sandwich: Is it better with an algebra in between? In *Proceedings of the SIGIR workshop on Information Retrieval and Databases (WIRD'04)*.

[9] Ogilvie, P. (2004). Retrieval using structure for question answering. In *Proceedings of the 1st Twente Data Management Workshop - XML Databases and Information Retrieval*, (pp. 15-23).

[10] O'Keefe, R. A., & Trotman, A. (2003). The simplest query language that could possibly work. In *Proceedings of the 2nd workshop of the initiative for the evaluation of XML retrieval (INEX)*.

[11] Pehcevski, J., Thom, J., & Vercoustre, A.-M. (2003). XML-search query language: Needs and requirements. In *Proceedings of the AusWeb03: Changing the Way We Work*.

[12] Piwowarski, B., & Gallinari, P. (2004). An algebra for probabilistic XML retrieval. In *Proceedings of the 1st Twente Data Management Workshop - XML Databases and Information Retrieval*.

[13] Sigurbjörnsson, B., & Trotman, A. (2003). Queries: INEX 2003 working group report. In *Proceedings of the 2nd workshop of the initiative for the evaluation of XML retrieval (INEX)*.

[14] Trotman, A., & Sigurbjörnsson, B. (2004). Narrowed Extended XPath I (NEXI). In *Proceedings of the 3rd workshop of the initiative for the evaluation of XML retrieval (INEX)*.