Narrowed Extended XPath I (NEXI)

Andrew Trotman
Department of Computer Science
University of Otago
Dunedin
New Zealand

SYNONYMS

None

DEFINITION

NEXI is an information retrieval (IR) query language for searching structured and semi-structured document collections. The language was first introduced for searching XML documents at the annual INEX [3] evaluation forum in 2004, and it has been used ever since.

Designed as the simplest query language that could possibly work, the language is a tiny subset of XPath [1] with an added *about*() function for identifying elements about some given topic. The language has extensions for question answering, multimedia searching, and searching heterogeneous document collections. NEXI is a language with a strict syntax defined in YACC but it has no semantics; the interpretation of the query is the task of the search engine.

HISTORICAL BACKGROUND

A common information retrieval query language for searching XML documents was needed for specifying information retrieval queries at the first INEX in 2002. There XML markup was chosen as the method of identifying keywords and the elements in which they should appear. It was also chosen as the method of identifying the preferred XML element to return to the user (the target element). The INEX 2002 query from topic 05 is given in Figure 1. In this example QBIC should be in a bibl element, image retrieval may appear anywhere in the document, and the user is interested in a list of tig elements as the result of the query.

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

Figure 1: INEX topic 05 in the 2002 XML format.

Two problems with this format were identified: first it allowed the specification of queries that could be resolved by a simple mechanical process; second the language was not sufficiently expressive for information retrieval queries.

A modified XPath [1] was used at INEX 2003. In this variant the *contains*() function that required an element to contain the given content was replaced by an *about*() function that required an element to be about the content. Changing XPath in this way allowed fuzzy IR

queries to be specified using a highly expressive language. However, an analysis of the XPath queries showed high syntactic and semantic error rates [6].

O'Keefe & Trotman [6] proposed using the simplest query language that could possibly work and a novel syntax. The INEX Queries Working Group [7] rejected the syntax but embraced the philosophy. It identified the minimum requirements of an IR query language for information retrieval queries containing structural constraints. This language, although at the time without syntax or semantics, was to be used at INEX for evaluation purposes.

Trotman & Sigurbjörnsson [9] proposed the Narrowed Extended XPath I (NEXI) language based on the working group report. It was narrowed in so far as only the descendant axis was supported and extended in so far as the *about*() function was added, all other functions and axis were dropped. A formal grammar and parser were published, and an online syntax checker was hosted by the authors.

The decision to reduce XPath resulted in fewer errors because it reduced the chance of making mistakes. NEXI has a precise mathematical formulation which matches intuitive user profiles [4]. For both naïve users with knowledge of just the tag names, and for more advanced users with additional knowledge of the inter-relationships of those tags, the language is safe and complete. That is, the user cannot make semantic mistakes, and can express every information need they have.

SCIENTIFIC FUNDAMENTALS

Web queries typically contain between 2 and 3 terms per query [8]. Formal query languages for semi-structured data tend to be comprehensive. This mismatch became apparent at INEX 2003 where XPath was chosen as the preferred language for information retrieval experts to specify relatively simple queries, but where they were unable to write syntactically and semantically correct queries. Just as SQL is not an end-user query language, neither, it turned out, was XPath.

Requirements

After two years of experimentation with XML query languages at INEX the needs of such a language became apparent. The INEX Queries Working Group [7] specified that the language should:

- Be compatible with existing syntax for specifying content only (keyword) queries.
- Be based on XPath as that language was already well understood, but:
    - Remove all unnecessary XPath axis used for describing paths. Limit to just the descendant axis was suggested. The child operator was considered particularly problematic as it was open to misinterpretation.
    - Drop exact match of strings, and inequality of numbers. XPath path filtering remained, however all strings were expressed as aboutness.
- Support multiple data types including numeric and string.
- Be open for extensions for new data types (including names, locations, dates, etc.).

- Not include tag instancing (for example `author[1]`, the first author).
- Have vague semantics open to interpretation by the search engine.
- Loosen the meaning of the Boolean operators AND and OR.
- Disallow the multiple target elements. Although not explicit in the requirement, the implication is that the target element must be about the final clause in the query. It is a simple mechanical process to add non-target elements that are not about the query to the result – such as the `author`, `title`, `source` details to `sections` about something.
- Allow queries in which the target element was not specified and in which the search engine identified the ideal element.

Content Only (CO) Queries

NEXI addresses two kinds of queries on semi-structured and structured data: Content Only (CO) and Content And Structure (CAS) queries.

Content Only (CO) queries are the traditional IR query containing only keywords and phrases. No XML restrictions are seen and no mention is given of a preferred result (target) element. For these the NEXI syntax is derived from popular search engines: search terms can be keywords, numbers, or phrases (delineated with quotes). Term restrictions can be specified using plus and minus.

Information Retrieval queries are by their very nature fuzzy. A user has an information need and from that need they express a query. There are many different queries they might specify from the same need, some of which might be more precise than the others. If a document in the document collection satisfies the user's information need, that document is relevant regardless of the query. That is, no query term might appear in a relevant document, or all the query terms might appear, either way the document is relevant. When specifying an IR query language it is important to avoid specifying semantics that violate this principle of relevance. The semantics of the terms with and without restriction in NEXI is, for example, specified this way:

"*The '+' signifies the user expects the word will appear in a relevant element. The user will be surprised if a '-' word is found, but this will not prevent the document from being relevant. Words without a sign are specified because the user anticipates such terms will help the search engine to find relevant elements. As restrictions are only hints, it is entirely possible for the most relevant element to contain none of the query terms, or for that matter only the '-' terms.*"

Or, in other words, it is the task of the search engine to identify relevant documents even if this involves ignoring the query.

In INEX topic 210 the author states:

"I'm developing a new lecture for the Master course 'Content Design' and want to discuss the topic "Multimedia document models and authoring". Therefore I want to do a quick background search to collect relevant articles in a reader. I expect to find information in abstracts or sections of articles. Multimedia content is an essential component of my lecture, thus for fragments to be relevant they should address document models of content authoring approaches for multimedia

content. I'm not interested in single media approaches or issues that discuss storing multimedia objects."

The query they give is

```
+multimedia "document models" "content authoring"
```

in which `"document models"` is a phrase and `+multimedia` is a term-restricted search term (is positively selected for by the user).

Content And Structure (CAS) Queries

The second kind of query addressed by NEXI is the Content and Structure (CAS) query. These queries contain not only keywords but also structural constraints know as *structural hints*. Just as the keywords are hints passed to the search engine in an effort to help with the identification of relevant documents, so too are structural hints. CAS queries contain two kinds of structural hints, where to look (support elements), and what to return to the user (target elements).

Formally queries many take one of the forms in Table 1:

Table 1: Valid forms of NEXI CAS queries

Form	Target element	Meaning
`//A[B]`	`A`	Return A tags about B
`//A[B]//C`	`A//C`	Return C descendants of A where A is about B
`//A[B]//C[D]`	`A//C`	Return C descendants of A where A is about B and a C descendant of A are about D

A and C are paths and B and D are filters. Other forms could easily be added, but since NEXI was originally designed to address the INEX query problem, they are not formally included.

Paths (A and C in Table 1) are specified as a list of descendants separated by the descendant axis `//`. Formally, a path is an ordered sequence of nodes $//E_1 \ldots //E_n$ starting with E_1 and finishing at E_n, and for all $e \in n$, E_e is a ancestor of E_{e+1}. An attribute node is indicated by the prefix `@`. Alternative paths are specified $(E_{na}|E_{nb})$. The wildcard `*` is used as a place holder.

For example, the path:

```
//article//*//(sec|section)//@author
```

describes an `author` attribute beneath either a `sec` or `section` element beneath something beneath an `article` element. The interpretation by the search engine is, of course, loose.

Filters (B and D in Table 1) can be either arithmetic or string. Arithmetic filters are specified as arithmetic comparisons (>, <, =, >=, <=) of numbers to relative-paths, for example: `.//year >= 2000`. String filters take the form `about(relative-path, COquery)`. Filters can be combined using the Boolean operators `and`, and `or`. Paths and filters are all considered hints and there is no requirement for the search engine to distinguish between the Boolean operators.

The target elements for the forms given in Table 1 are specified in column 2. Target elements, like support elements, are also hints. If, for example, the user specified paragraphs a subsection element might fulfill the user's information need.

An example of a valid NEXI CAS query (again from INEX topic 230) is:

```
//article[about(.//bdy, "artificial intelligence") and
.//yr<=2000]//bdy[about(., chess) and about(., algorithm)]
```

in which the target element is `//article//bdy`. The user has specified an arithmetic filter `.//yr<=2000`. Several string filters are used including `about(.//bdy, "artificial intelligence`"). A Boolean operator is also used to separate two filters `about(., chess) and about(., algorithm)`.

The NEXI CAS query from INEX topic 210 is an alternative expression of the information need given in the previous section. That query is:

```
//article//(abs|sec)[about(.,+multimedia "document models"
"content authoring")]
```

in which the target element is either `//article//abs` or `//article//sec`. The same documents and elements are relevant to both queries as relevance is with respect to the information need and not the specific query.

KEY APPLICATIONS

Information retrieval from structured and semi-structured document collections.

FUTURE DIRECTIONS

Although proposed as an XML query language for use in an evaluation forum, there is evidence it may also be an effective end-user language. Van Zwol *et al*. [11] compared NEXI to a graphical query language called Bricks. They found that a graphical query language reduced the time needed to find information, but that users were more satisfied with NEXI. Inherent in text query languages is the problem that users are required to know the structure (the DTD) of the documents. In a heterogeneous environment this may not be possible, especially if new and different forms of data are constantly being added. Graphical query languages that translate into an intermediary text-based query language are one solution. This solution is seen with graphical user interfaces to relational databases.

Woodley *et al*. [13] further the model of NEXI as an intermediate language and compare NLPX (a natural language to NEXI translator) to that of Bricks (a graphic to NEXI translator). They show that users prefer a natural language interface, and that the performance of the two is comparable.

Ogilvie [5] examined the use of NEXI for Question Answering and proposed extensions to the language for this purpose. Dignum & van Zwol [2] proposed extensions for heterogeneous searching. Trotman & Sigurbjörnsson [10] unified these proposals and formally extended the language to include both – however these extensions are not considered core to the language (language extensions philosophically deviate from the principle of simplest that could possibly

work). Multimedia extensions to the language have also been used at INEX [12], again the extensions are not considered core to the language.

EXPERIMENTAL RESULTS

The analysis of XPath queries used at INEX 2003 showed 63% of queries containing either syntactic or semantic errors [6]. An analysis of the errors in NEXI queries used at INEX 2004 showed that only 12% contained errors [10]. NEXI has been in use at INEX ever since.

DATA SETS

INEX queries from NEXI 2004 onwards can be downloaded from the INEX web site:
http://inex.is.informatik.uni-duisburg.de/

INEX queries for 2003 and 2002 were translated into NEXI (where possible) and can be downloaded from the NEXI web page hosted by the University of Otago:
http://metis.otago.ac.nz/abin/nexi.cgi

URL TO CODE

An online syntax checker, lex and yacc scripts, and a command line syntax checker can be downloaded from the NEXI web page hosted by the University of Otago:
http://metis.otago.ac.nz/abin/nexi.cgi

CROSS REFERENCES

Content-Only Queries,
Content-And-Structure Queries,
Document Path Query,
Evaluation Initiative For XML Retrieval (INEX),
Processing Structural Constraints,
Query By Humming,
Query Languages For Biological Data,
Semi-Structured Query Language,
Temporal Query Languages,
XML,
XPath/XQuery,
XQuery Full Text,
XSL/XSLT

RECOMMENDED READING

[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.

[2] Dignum, V., & van Zwol, R. (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., Gövert, N., Kazai, G., & Lalmas, M. (2002). INEX: Initiative for the evaluation of XML retrieval. In *Proceedings of the ACM SIGIR 2002 Workshop on XML and Information Retrieval*.

[4] Kamps, J., Marx, M., Rijke, M. d., & Sigurbjörnsson, B. (2006). Articulating information needs in XML query languages. *Transactions on Information Systems,* 24(4):407-436.

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

[6] 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).*

[7] 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).*

[8] Spink, A., Wolfram, D., Jansen, B. J., & Saracevic, T. (2001). Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology,* 53(2):226-234.

[9] Trotman, A., & Sigurbjörnsson, B. (2004). Narrowed Extended XPath I (NEXI). In *Proceedings of the INEX 2004 Workshop,* 16-40.

[10] Trotman, A., & Sigurbjörnsson, B. (2004). NEXI, now and next. In *Proceedings of the INEX 2004 Workshop,* 41-53.

[11] van Zwol, R., Baas, J., van Oostendorp, H., & Wiering, F. (2006). Bricks: The building blocks to tackle query formulation in structured document retrieval. In *Proceedings of the 28th European Conference on Information Retrieval (ECIR 2006),* 314-325.

[12] Westerveld, T., & van Zwol, R. (2007). Multimedia retrieval at INEX 2006. *SIGIR Forum,* 41(1):58-63.

[13] Woodley, A., Geva, S., & Edwards, S. L. (2007). Comparing XML-IR query formation interfaces. *Australian Journal of Intelligent Information Processing Systems,* 9(2):64-71.