XML is the data model, what is the function of the query language?

Position paper for W3C query language workshop

Karl Aberer, Peter Fankhauser, Erich Neuhold

{aberer,fankhaus,neuhold}@darmstadt.gmd.de

GMD IPSI

18.11.98

Several proposals from different communities exist. With some simplification they can be classified according to their functional goal.

- Text retrieval (e.g. PAT) - identify documents in large document bases
- Addressing documents (e.g. Xlink) - have flexible link models
- Document layout (e.g. XSL) - transform documents for layout
- Document transformation (e.g. tree regular grammars) - transform documents from one DTD to the other
- Databases (e.g. OQL) - map documents into data structures to apply DB techniques
- Knowledge bases (e.g. RDF oriented) - map knowledge bases into XML to perform reasoning
- etc

As a consequence of their different functional goals these proposals base on subtly or substantially different data models. For large document base processing a well-defined and stable model is desirable, at least for a kernel model (see relational algebra and query optimization). To focus, in the following we do not consider the query languages for data models that are encoded within XML for using XML as an data exchange format (such as RDF), and query languages for data models that are used to encode XML documents for using the data model as an implementation vehicle (such as DOM).

To get an assessment of the relevance of the diverse functional goals of a query language for XML we need to assess the following questions.

- what is a database and the subject and the result of a query?
  - a set of document elements
  - a set of document fragments
  - a document
  - a collection of documents according to a DTD
  - a collection of documents according to multiple DTDs
  - a collection of well-formed documents
  - a web site
  - the Web

Assuming this question is clarified we use the notion of document as a place holder for the chosen subset of the above.

- what are relevant query criteria?
• the hierachical structure of the document
  • the sequential structure of document
  • attribute values and element contents
  • interpreted contents of elements and values of attributes (object ids, links, multimedia contents etc.)

Selecting these criteria lays the foundation for the expressivness of the query language.

- what is the purpose of queries?
  • selection on documents
  • restructuring of documents
  • combination of documents
  • aggregation of documents
  • ranking of documents
  • inferencing on documents

Our main interest in XML lies in using it as an infrastructure for information integration and brokering. This requires means to retrieve, restructure, merge and view documents from the most diverse sources and thus at least the first five points. With these rather ambitious functional goals we think it is beneficial to be pragmatic in other respects, in particular with respect to the data model. Therefore, we suggest the following procedure to arrive at an XML query language.

Step 1:

Data Model = XML, that means in particular that element sequences (as opposed to unordered sequences), element containment, and attributes with their domains (notations!), need to be taken into account. This restriction to a stable, well defined data model provides a notion of completeness against which the above listed functionality can be judged; everything beyond should be seen as extension.

Step 2:

URIs as the W3C addressing scheme should be taken into account for navigational queries as first extension.

Step 3:

Query language extensions requiring features such as Object Ids, attribute types etc. inherited from other data models (such as OO, semistructured, relational, extensible, knowledge-based) should be clearly seen as extensions to the data model and therefore be discussed with the according working groups.