XQuery Midflight: Emerging Database-Oriented Paradigms and a Classification of Research Advances

ADVANCED SEMINAR PROPOSAL TO ICDE 2005 by
Ioana Manolescu, INRIA
Yannis Papakonstantinou, UCSD

Duration: Preferably **3 hours**, can do reduced to outline in **90 minutes**.

ABSTRACT: MOTIVATION AND HIGH-LEVEL GOALS

XQuery processing is one of the prime research topics of the database community, as is evident from the number of systems and publications. Systems, architectures, principles, and algorithms rapidly emerge for all its incarnations; be it in message/file transformations, XML caching, XML content management, or XML-based publishing and mediator systems.

At the same time, XQuery research is still in a "pre-paradigmatic" stage, where the conventional symptoms of the stage are observed: It is hard to piece together point efforts into a big picture. Similarities and interplay opportunities between parallel efforts are "lost in the translation" across the different paradigms. This is a natural stage in the evolution of most science and technology topics and our references to the classic 1930's works of Kuhn on the development and evolution of science will make sure the audience gets rid of any guiltiness we may accidentally create. Nevertheless, the time is ready for the next stage: The goal of this tutorial is to "federate" among the plethora of works, and categorize existing work and future topics along a few reference paradigms that fuse existing works around a reference architecture.

The focus will be on database-oriented issues, in the sense of focusing on issues and drawing parallels with the principles and techniques of database systems, as explained in the outline below.

OUTLINE

1. Quick overview of standards and abstractions
    a. XQuery/XPath data model and its labeled tree abstraction. Includes a brief commentary on benign and non-benign differences between the reality of the model and the (typically used in research works) labeled tree abstraction

    b. XQuery/XPath type system and typical abstractions. Same commentary with above.

    c. XPath and its tree pattern abstractions (including abstractions that capture sibling and parent navigation)

2. What is XQuery for? Short overview of typical uses:
    a. Transformer of files, message streams

    b. Query language for XML caches, XML databases and XML content management

c. Query and view definition language in mediators/relational publishing

3. What is pre-paradigmatic science? (Quick non-database intermission)
   a. Did you feel these pre-paradigmatic symptoms recently?

   b. No need to worry: it happens at some stage in *all* scientific communities!

4. XQuery reference architecture with emphasis on typical database research themes
   a. We quickly go over parsing, typing, and PL-oriented rewritings

   For each of the following topics, we follow the same pattern. First, we present a paradigm (e.g. a model, an algebra, etc) that fuses existing works, then we discuss differences between the constituting works and the paradigm, and finally we outline open issues. Note we do not imply that all of the following *need* to be part of an implementation. Instead, we outline pluggable modules that lead to various XQuery incarnations.

   b. Logical Level Optimization
      i. Overview of the works on logical-level tree pattern (XPath) optimization.
      ii. Generalized Tree Patterns paradigms capturing XQuery subsets.
      iii. Common sub-expression factorization.
      iv. Answering queries using views and caches, containment.

   c. Physical Level
      i. Storage and Indexing Schemes for caches and DBs. Includes:
         1. path indices,
         2. XML data shredding into relational.
         3. unifiying storage, indexing and materialized view creation for XML

      ii. Wrapper issues for mediators and relational publishers (small topic)

      iii. Stream and Navigation Models

   d. Physical Level Plans
      i. A relational algebra-like paradigm that fuses the multiple tuple-oriented algebras proposals

   e. Automata-oriented Plans (will be given less time, since they were discussed extensively in the ICDE04 tutorial)

   f. Structural Join algorithms

   g. Dealing with Nesting

REMARK

This tutorial is designed as a continuation of last year's ICDE tutorial, titled "XML Query Processing", by Daniela Florescu and Donald Kossman. Unlike the ICDE 2004 tutorial, issues

primarily pertaining to programming languages (e.g. typing and PL-oriented optimizations) will not be covered. The same applies to information retrieval-related issues (e.g., keyword search in XML).

SHORT BIOS

Ioana Manolescu is a researcher in the Gemo group in INRIA Futurs, France.
Ioana has obtained her PhD in 2001 from University of Versailles and INRIA, France, working on query optimization for distributed databases, and XML. Her thesis work on distributed query optimization was incorporated into a new start-up, Medience. She has worked as a post-doc in Politecnico di Milano, Italy, on extending the WebML model to cope with Web services and workflow specification. Her current research topics include XML data storage, query algebras and query processing, XML compression, and distributed data and process management based on Web services. Ioana has (co-)authored several tutorials and advanced courses for the EDBT database summer school. More information on Ioana's projects and can be found at http://www-rocq.inria.fr/~manolesc.

Yannis Papakonstantinou is an Associate Professor of Computer Science and Engineering at the University of California, San Diego. His research is in the intersection of database and Internet technologies. Yannis has published over fifty research articles in scientific conferences and journals, given tutorials at major conferences, and served on journal editorial boards and program committees for numerous international conferences and symposiums. He was the co-Chair of WebDB 2002, the co-Chair of XIME-P 2004, the General Chair of ACM SIGMOD 2003 and the Vice PC Chair for the ``XML, Metadata and Semistructured Data" track of IEEE ICDE 2004. In 1998, Yannis received the NSF CAREER award for his work on integrating heterogeneous data. In 2000 Yannis founded Enosys Software, which built the first generally available distributed XQuery processor, along with software for XML-based integration of distributed sources, and was sold in 2003 to BEA Systems. Yannis holds a Diploma of Electrical Engineering from the National Technical University of Athens and MS and Ph.D. in Computer Science from Stanford University (1997). His complete bio is available at http://www.db.ucsd.edu/people/yannis.htm

Most notably, Ioana and Yannis have organized the First International Workshop on XQuery Implementation, Experience and Perspectives (XIME-P 2004) in cooperation with ACM SIGMOD 2004. This tutorial proposal is to a large extent motivated by the comment we received from many panelists and participants for a systematic approach to XQuery research.

[Kuhn] Thomas S. Kuhn. "The Structure of Scientific Revolutions"