Chapter I

Using Logic for Querying XML Data

Nick Bassiliades and Ioannis Vlahavas
Aristotle University of Thessaloniki, Greece

Dimitrios Sampson
Informatics and Telematics Institute, Greece

ABSTRACT
In this chapter, we propose the use of first-order logic, in the form of deductive database rules, as a query language for XML data, and we present X-DEVICE, an extension of the deductive object-oriented database system DEVICE, for storing and querying XML data. XML documents are stored into the OODB by automatically mapping the DTD to an object schema. XML elements are treated either as classes or attributes based on their complexity, without losing the relative order of elements in the original document. Furthermore, this chapter describes the extension of the system’s deductive rule query language with second-order variables, general path and ordering expressions, for querying over the stored, tree-structured XML data and constructing XML documents as a result. The extensions were implemented by translating all the extended features into the basic, first-order deductive rule language of DEVICE using meta-data about stored XML objects.

INTRODUCTION
The success of the Internet depends on the availability of applications that will offer valuable e-services. However, applications have always depended on input data and most importantly on their well-structuredness. So far, information is captured and exchanged over Internet through HTML pages, without any conceptual structure. XML is the currently proposed standard for structured or even semi-
structured information exchange over the Internet (W3 Consortium, Oct 2000). However, the maintenance of this information is equally important. Integrating, sharing, re-using and evolving information captured from XML documents are essential for building long-lasting applications of industrial strength.

The story of information management or data management has been told before in the form of DBMSs. Over three decades of research have been devoted to developing theory and systems for capturing, storing, maintaining and retrieving data for a single or multiple users. Such a vast research and development wealth should be re-used with the minimum of effort for managing semi-structured data, i.e., XML or SGML, which is the super-set of XML. There already exist several proposals on methodologies for storing, retrieving and managing semi-structured data stored in relational, object-relational and object databases. Furthermore, there exist quite a few approaches in storing SGML multimedia documents in object databases.

Capturing XML data in traditional DBMSs is one aspect of the story. Effective and efficient querying and publishing these data on the Web is another aspect that is actually more important since it determines the impact this approach will have on future Web applications. There have been several query language proposals (Abiteboul, Quass, McHugh, Widom, & Wiener, 1997; Buneman, Fernandez, & Suciu, 2000; Chamberlin, Robie, & Florescu, 2000; Deutsch, Fernandez, Florescu, Levy, & Suciu, 1999; Hosoya & Pierce, 2000; Robie, Lapp, & Schach; W3 Consortium, Dec 2001d) for XML data. Furthermore, recently the WWW consortium issued a working draft proposing XQuery (W3 Consortium, Dec 2001a), an amalgamation of the ideas present in most of the proposed XML query languages of the literature. Most of them have functional nature and use path-based syntax. Some of them (Abiteboul et al., 1997b; Chamberlin et al., 2000; Deutsch et al., 1999a), including XQuery, have also borrowed an SQL-like declarative syntax, which is popular among users. Some of the problems relating to most of the above approaches is the lack of a comprehensible data model, a simple query algebra, with the exception of (Buneman et al., 2000; Hosoya & Pierce, 2000) and query optimization techniques. There are proposals for a data model (W3 Consortium, Dec 2001b) and a query algebra (W3 Consortium, Jun 2001) for XQuery, however it is not yet clear how these will lead to efficient data storage and query optimization.

In this chapter, we propose the use of deductive rules as a query language for XML data, and we present X-DEVICE, a deductive object-oriented database for managing XML data. X-DEVICE is an extension of the active object-oriented knowledge base system DEVICE (Bassiliades, Vlahavas & Elmagarmid, 2000). DEVICE integrates high-level, declarative rules (namely deductive and production rules) into an active OODB that supports only event-driven rules (Diaz & Jaime, 1997), built on top of Prolog. This is achieved by translating each high-level rule into one event-driven rule. The condition of the declarative rule compiles down to a set of complex events that is used as a discrimination network that incrementally matches the rule conditions against the database.

X-DEVICE extends DEVICE by incorporating XML data into the OODB by automatically mapping DTDs of XML documents to object schemata, without
Related Content

Using Temporal Versioning and Integrity Constraints for Updating Geographic Databases and Maintaining Their Consistency
www.igi-global.com/article/using-temporal-versioning-and-integrity-constraints-for-updating-geographic-databases-and-maintaining-their-consistency/140545?camid=4v1a

Extending Agile Principles to Larger, Dynamic Software Projects: A Theoretical Assessment
www.igi-global.com/chapter/extending-agile-principles-larger-dynamic/74402?camid=4v1a
Object-Oriented System Design
[www.igi-global.com/article/object-oriented-system-design/51111?camid=4v1a](www.igi-global.com/article/object-oriented-system-design/51111?camid=4v1a)

The Impact of Conceptual Data Models on End-User Performance
[www.igi-global.com/article/impact-conceptual-data-models-end/51109?camid=4v1a](www.igi-global.com/article/impact-conceptual-data-models-end/51109?camid=4v1a)