LOGIC BASED QUERIES FOR XML DATABASES

By

Qing Wang

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF INFORMATION SYSTEMS
AT
MASSEY UNIVERSITY
PALMERSTON NORTH, NEW ZEALAND
DECEMBER 2005

MASSEY UNIVERSITY
DEPARTMENT OF
INFORMATION SYSTEMS

The undersigned hereby certify that they have read and recommend to the Department of Information Systems for acceptance a thesis entitled "**Logic Based Queries for XML Databases**" by **Qing Wang** in partial fulfillment of the requirements for the degree of **Master of Information Systems**.

Dated: December 2005

Supervisor: ______________________
Klaus-Dieter Schewe

Readers: ______________________

____

MASSEY UNIVERSITY

Date: **December 2005**

Author: **Qing Wang**

Title: **Logic Based Queries for XML Databases**

Department: **Information Systems**

Degree: **M.I.S.** Convocation: **February** Year: **2006**

Signature of Author

Table of Contents

Acknowledgements

I would like to express my gratitude to the many people who have made this thesis possible.

Most of all I would like to thank Professor Klaus-Dieter Schewe, my supervisor, who recommended this interesting research topic to me when I was in confusion at the early stage of Masters study, and introduced me to the world of logic theory. Without his patient guidance and stimulating suggestions, I would have been lost somewhere. I am also very grateful to Dr. Sven Hartmann for being a constant source of support and encouragement at all times, which greatly inspired me to overcome many difficulties in my life.

Special thanks go to Thu Trinh and Madre Chrystall. They not only dedicated their time to being a reader of my manuscript, but also provided many helpful comments. Furthermore, I wish to thank all the many teachers, friends and classmates who helped me get where I am today although it is impossible to mention them here individually.

In particular, I wish to convey my thanks to Mrs Clark, Massey University, and Mr Todd and the NZVCC , for being awarded the Lovell & Berys Clark Scholarship, Massey University Masterate Scholarship and Todd Award for Excellent in 2005. These scholarships are crucial to the successful completion of this research because they not only free me from financial worries, but also motivate me towards higher goals for this research.

Lastly, and most importantly, I am forever indebted to my family for their understanding, endless patience and encouragement when it was most required, especially my three-year-old son Jiajun, he always seems quite understanding toward my study although he is not really aware of that.

Wang, Qing
December 1, 2005

Chapter 1

Introduction

With the significant increase of web-based applications, the eXtensible Markup Language (XML), as a de facto standard for data interchange on the web, has attracted considerable attention in theory and practice. XML deals with irregular and heterogeneous semi-structured data that give rise to trees, i.e. the predominant data structure in complex data models. This leads to new challenges for database research such as new data structures and models, and new database query languages in this area.

1.1 XML Data Challenges

In essence, the major challenge of XML lies in its data structure, which is greatly different from the traditional data structure treated as relations. A comparison with respect to the differences between XML data and relational data has been discussed in [17]. Generally speaking, XML data has the following unique characteristics:

- XML data have irregular and heterogeneous structures leading to optional values which are absent in relational data;
- Metadata that describes the structure of the data is distributed throughout the data in the form of markup, instead of being stored separately as in relational data;
- XML data is nested in a hierarchy, and complex nested structures might need to be decomposed and reconstructed on the fly to facilitate data manipulations;
- XML data are encoded with an intrinsic order that is an important property of data themselves. This means that the order property must be taken into consideration for modelling.

These features require more complicated data models in comparison to the relational data model, which merely deals with flat and rigid relations. In particular, problems such as rational trees, optional cardinality, repeated elements and sequence order demand additional functionalities to be provided to handle XML data in an efficient and sufficient manner.

Since the unique structures of XML data further require that XML query languages must have the capability for querying over schema information. In order to support this extra functionality at the database level, the techniques used in semantic data models have shown to be an appropriate approach, in which the separate notions of schema and instance are combined into an explicit expression. This allows the queries of schema information to be handled in the same manner as querying data at the language level.

Therefore, the first task investigated in this thesis is to design a simple and natural data model, which supports a rich description of XML data from a structural point of view. The relevant techniques for this are derived from semantic data models.

In the theory of relational databases, query languages have been developed mainly following three directions: an algebraic direction e.g. the Relational Algebra, a logical direction e.g. the Relational Calculus, and a logic programming direction e.g. deductive languages such as DATALOG. Many issues such as recursion, negation and the combination of them have been well investigated in relational query languages. However, some new problems have emerged with respect to XML query languages.

- the possible absence of a database schema;
- the capability to support querying over schema information;
- restructuring and creating XML data in a flexible approach;
- recursion under the feature of optionality.

Not surprisingly, the second task investigated in this thesis aims at developing declarative and powerful XML query languages in the context of XML data. My focus in this thesis is on a logical direction and a logic programming direction.

1.2 Objectives

The main goal of this thesis is to investigate the logical grounds of query languages over XML databases on the basis of a representation by object bases. In general, there are three objectives I want to achieve.

–Define a data model at the conceptual level for XML data aiming at capturing semantic capabilities on object bases.

–Develop the formal syntax and semantics for a logic programming query language for XML, which exploits object identifiers as primitives in the context of XML data structures.

–Propose a higher-order predicate calculus language tailored to XML data by studying the mathematical techniques used in relational calculus which is regarded as a specification of the first-order predicate calculus.

To clarify how the objectives are fulfilled in this thesis, we provide a brief introduction regarding each objective in this subsection.

1.2.1 The Semantic XML Object Model

The semantic XML object model aims at a natural and simple data model for XML-based databases that can capture and express XML data from a relatively high abstraction level with well-defined semantics. Furthermore, it can also serve as a formal specification for further establishing XML data manipulation languages.

The proposed semantic XML object model mainly has the following key features:

- It provides considerably rich structuring capability by using objects as basic semantic data abstractions that are associated with concise but flexible interpretations.
- It provides classes as the only construct to accommodate objects, thus enabling richness through class tuples and class schemata that link XML database instances and XML database schemata together.

- It provides attributes at the object level, which are used to incorporate values and objects into a unit, and attributes at the class level, which facilitate to capture objects having common semantics but variant structures within a specific class schema.

- It provides object identifiers that are used in a dual role as both structure and order primitive.

1.2.2 The XML Identity Query Language

The XML Identity Query Language (XIQL) is greatly influenced by the spirit of IQL in [6, 7] on a key design point that objects can be manipulated through having object identifiers as a powerful programming primitive. New objects can be created in a manner that is essentially equivalent to Skolem function techniques. However, more work needs to be done with respect to IQL to reflect the characteristics of XML data.

Compared with IQL, XIQL provides the following additional functionalities:

- XIQL incorporates the capability for type creation in queries. On one hand new types can be easily added into a database, while on the other hand types can be queried as a special part of an instance.

- XIQL enriches the language with union types, which can capture the optionality feature of XML data. Further, the underlying intersection types facilitate the flexible expressions for objects with class atoms.

- XIQL treats set and non-set variables in a unified fashion unless a particular interest is indicated. In this case, a set operator can be used to handle the transformation between set and non-set variables.

1.2.3 The XML Calculus Language

The XML calculus language is a pure declarative language incorporating higher-order logics on the basis of the SXO model, in which semantic structures can be elegantly encapsulated into objects.

The design goal of the XML calculus language is to obtain highly expressive power, but using a relatively simple interpretation. To achieve this goal, the combination of a higher-order syntax and a first-order semantics within a language becomes a suitable choice for the proposed XML query language. Furthermore, through a connection of two semantics of objects, an equation between the first-order and higher-order interpretations can be established.

To handle higher-order logics, it is essential to define a fundamental type system in the language, on which higher-order notions can be precisely developed. Moreover, for an XML query language, the ability to query schema information must be taken into consideration. For this reason, type variables are introduced into the type system.

Another interesting design point is to naturally reflect fixpoint semantics in this language by means of universal quantification, which is associated with variables under restricted domains. With this capability, the expressive power of the language will be greatly enhanced.

1.3 Principals

Throughout this thesis two main principles will be followed: considering XML databases as object bases and following a logic paradigm for querying them.

1.3.1 Object Bases

The data model developed in my thesis is built upon object bases. Instead of encapsulating behavioral aspects of objects in object-oriented databases, this data model concentrates on encapsulating structural aspects from a semantic point of view. The reasons for preference for an object-based model, rather than a relational model, principally arise from the following considerations.

Firstly, as argued in [63], the relational data model fails to capture much of the semantics associated with data because the fundamental modelling construct, i.e. the tuple does not constitute an atomic semantic unit. Hence, additional integrity constraints are required to establish the intended semantics of the database. In contrast to tuples, the features of objects determine themselves to be the natural semantic carriers in XML databases.

Secondly, to tackle the irregular and heterogenous structure of XML data, objects grouped in a class can provide richer structure modelling facilities than tuples in a relation. By specifying classes, which represent collections of objects encoded with heterogenous structures, a uniform framework on object bases can be established.

Finally, it is customary that XML data has a graphical representation, which is also thought of as a representation for objects in the object-based model. Hence, they coincide with respect to this intuition. Furthermore, both objects and graphs can be constructed to be self-descriptive, which is an important feature identified in XML data. However, a tabular representation provided in the relational data model is not suitable in this case due to the separation of schema and instance.

In addition, although many XML query languages have been proposed such as XQuery [73], XQL [56], LOREL [8] and XML-QL [29], most of them aim at providing the XML-era analogue of SQL. However, I believe that the investigation on XML query languages from a perspective of object bases has not been sufficiently conducted yet so far, especially for exploiting object identifiers as structure and order primitives.

1.3.2 Logic Paradigm

In the thesis, a study towards logic-based query languages in the framework of an object-based XML data model will be conducted. The reason for considering the logical ground of query languages is based on the following two points:

- Firstly, logic-based approaches have been recognized as a means to provide remarkable simplicity and conciseness of syntax for query languages, as were the case in relational calculus and Datalog for the relational data model.

- Furthermore, logic-based approaches are essential for evaluating the expressive power and computable complexity of query languages.

Most of the current XML query languages focus on path-based query processing with pattern matching techniques. Research regarding theory, well-defined semantics and expressiveness is still needed.

1.4 Overview of the Thesis

The remainder of the thesis is organized according to the following logical sequence.

In Chapter 2, we focus on modelling XML data in a natural way by applying techniques of semantic data model. The developed data model, called the SXO model, is built upon object bases, in which identifier and value semantics of objects are formalized. Moreover, XML database instances and schemata are formalized along with the notions of class schema and class tuple. At the end of the chapter, some insights are given into dominant relation, class hierarchy and object order. Based on this model, XIQL, a simple and powerful logic programming language for manipulating XML data, is proposed in Chapter 3. A basic framework of this language with the formal syntax and semantics is presented there. Furthermore, we discuss the object creation mechanism adopted in XIQL in details. Chapter 4 investigates the issues about structured value duplicates and copies. It turns out that XIQL is complete with respect to determinate transformation since both structured value duplicates and copies can be eliminated in XIQL queries. To obtain a pure declarative XML query language, Chapter 5 incorporates higher order logics into a novel XML query language called XML calculus as a counterpart of relational calculus in the relational data model. This language is developed with higher-order syntax and first-order semantics. After introducing the formal syntax and semantics of the language, a logical reflection regarding the relationship between first-order semantics and higher-order semantics is discussed. Chapter 6 reviews the literature from three aspects: semantic data models, object-creating languages and pure declarative languages, as a complement to some related work having been introduced during Chapter 2, 3, 4 and 5. In the end, we summarize the main results exploited in this thesis in Chapter 7. Several issues left for future work are also identified at the end of Chapter 7.

Throughout the thesis we will use a simple XML database as presented in Figure A.1, Figure A.2 and Figure A.3 of Appendix A. All the data stem from XML Query Use Case [72] provided by the World Wide Web Consortium (W3C) with some minor modifications. The purpose of the running XML database is to apply the proposed query languages to specific application scenarios, and illustrate the capabilities of our languages.