Semantic XML query optimization

Ke Geng
Supervisor: Gill Dobbie

The objective of my research is to carry out query optimization of XML queries based on the content of the documents.

Now eXtensible Markup Language (XML) [1] is widely accepted by website constructors and programmers because of its capability and its flexibility in storing and transferring semistructured data. Query transformation is the method used to improve the query execution time of the database system. Some research has been done with XML query transformation. To date, these works concentrate on the structure of XML documents and little work has been done that takes into account the content of XML documents. With XML documents there is a need to concentrate more on the content because XML is more flexible than other databases and XML documents do not typically have a schema. One of the disadvantages of the flexibility is that the structure of XML documents is inconsistent and unpredictable.

Web Ontology Language (OWL) [4] is an ontology language that can formally describe the meaning of terminology used in XML documents. In contrast to other ontology languages, such as RDF (Resource Description Framework) [6], OWL provides more meaningful terms to describe the relationships of elements of the XML document. With these description methods, the elements in XML document can be grouped and described based on the features of their contents.

In this research, we will undertake work in the content analysis of XML documents and how the content analysis can help XML query transformation. All the elements in an XML document will be analyzed and classified with their content features using an On Line Analytical Processing (OLAP) [10] system. We choose the OLAP system as the analysis tool because it has two important features: a multidimensional cube can help us to analyze the XML documents from different dimensions and the hierarchy provides a method to classify the elements with their content. The generated groups will be described with their features in an OWL document. With the generated OWL documents, the input XML query can be transformed to a more efficient one, which has the same function as the original query.

Currently little work on building OLAP systems for XML databases has been done because of XML's unpredictable structure. The existing methods, which can present XML content in an OLAP system, can not store all the information of XML documents, especially the hierarchy relationship between elements.

The optimization method is described in the following steps:

1. Extract and analyze the content of an XML document using OLAP technology.
2. Transfer the result of the analysis and represent it in OWL.
3. Carry out query optimization using the content of the OWL document.

Some of the challenges of this research include:

- What is the best way to describe elements that leads to best result in XML query transformation?

  OWL provides several patterns to describe the elements of the XML document. For example: The element “dot” can be defined as a fundamental class of two subclasses “commercialDot” and “residentialDot”. It can also be described as the union of “commercialDot” and “residentialDot”, which looks like:

```
<owl:Class rdf:ID="dot">
   <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="commercialDot" />
      <owl:Class rdf:about="residentialDot" />
   </owl:unionOf>
</owl:Class>
```

  These two definitions have their own features. When the “dot” is defined as the fundamental class of two subclasses, we do not restrict that only these two subclasses are built based on class “dot”. The element “dot” may have other subclasses. This definition is flexible and easy to update when the original XML document is changed.

  When the “dot” is defined as the union of “commercialDot” and “residentialDot”, it has been restricted such that there are only two kinds of Dots in the composition of element “dot”. The relationship of elements defined with “unionOf” is more restricted than the definition of “subclass”, which may reduce the complexity of the OWL document analyzing procedure and help the database system to derive knowledge efficiently. But it is not as flexible as “subclass” to be maintained.

- How can the size of the OWL document be controlled?

  There is a trade-off between the amount of data that should be represented and the time it takes to deal with that data.

  Deriving knowledge from XML to generate OWL documents is carried out through the implementation of analysis with OLAP. Deriving knowledge from OWL documents for the query transformation is part of the algorithm of the implementation using OWL. The relation between the two implementations is very interesting. The more knowledge can be derived from the XML documents, the simpler the OWL implementation is. But this will increase the difficulty of the implementation of OALP analysis procedure. If the implementation of BI is not powerful enough, we should do more work to design a powerful algorithm to derive enough knowledge for the query transformation from the generated OWL document.

- What are the best data and tools for testing the system adequately?

  A number of benchmarks for XML have been developed recently. These benchmarks can be classified into two classes: application benchmarks and micro benchmarks. Application benchmarks are used to evaluate the overall performance of a database system by testing as many query language features as possible. Currently, most benchmarks of XML are application benchmarks. These

benchmarks include: XMark [8], XBench [2], XOO7 [9] and XMach-1[5]. Micro benchmarks are used to test individual performance critical features of the language. There is only one micro benchmark for XML: Michigan Benchmark [7]. In my research, the content structure is the key point for the optimization. All the existing benchmarks can not satisfy the experiment because the benchmarks are designed for general purposes and the content is built with random algorithm. So seldom can content features be abstracted from the generated XML documents. A tool, which can adapt the existing benchmarks and build some content information for the experiment, is necessary in generating data (XML documents) for the experiments. Also there is no Native XML Database (NXD) [3] system that supports query transformation with information presented in OWL. So an assistant system, which can traverse OWL document and find the related information, must be designed. An algorithm, which can traverse OWL documents effectively without the support of some existing method, such as index, also needs to be designed.

In this paper, a semantic query optimization method is introduced. The challenges of the project are also discussed in detail. Now a tool that can adapt the existing XML documents has been designed, which can modify not only the content but also the structure of the XML document for the experiment. An assistant system, which can traverse OWL document and carry out query optimization with the information stored in the OWL document is being designed. More functions, such as choosing useful description from several descriptions of one element, will be added to the assistant system in the future research.

Reference

[1] Extensible Markup Language (XML). (2006). Retrieved 24/06, 2006, from http://www.w3.org/XML/

[2] Benjamin Bin Yao, M. T. Ö. (2002). XBench - A Family of Benchmarks for XML DBMSs Retrieved 15/09, 2005, from http://se.uwaterloo.ca/~ddbms/projects/xbench/People.html

[3] Bourret, R. (2005). Going native: Use cases for native XML databases. Retrieved 03/05, 2005, from http://www.rpbourret.com/xml/UseCases.htm

[4] Deborah L. McGuinness, F. v. H. (10/02/2004). OWL Web Ontology Language Overview. 20/08/2005, from http://www.w3.org/TR/owl-features/

[5] Erhard Rahm, T. B. (2002). XMach-1: A Multi-User Benchmark for XML Data Management. Paper presented at the Conference Name|. Retrieved Access Date|. from URL|.

[6] Frank Manola, E. M. (2004). RDF Primer. Retrieved 27/08, 2005, from http://www.w3.org/TR/rdf-primer/

[7] Kanda Runapongsa, J. M. P., H.V. Jagadish, Yun Chen, Shurug Al-Khalifa (2001). The Michigan Benchmark: Towards XML Query Performance Diagnostics. from http://www.eecs.umich.edu/db/mbench/mbench.pdf

[8] Ralph Busse, M. C., Daniela Florescu,Martin Kersten,Ioana Manolescu,Albrecht Schmidt, Florian Waas. (2003). XMark — An XML Benchmark Project. Retrieved 18/09, 2005, from http://monetdb.cwi.nl/xml/index.html

[9] Stéphane Bressan, M. L. L., Ying Guang Li, Bimlesh Wadhwa, Zoé Lacroix, Ullas Nambiar, Gillian Dobbie The XOO7 Benchmark Retrieved 17/09, 2005, from http://www.comp.nus.edu.sg/~ebh/XOO7.html

[10] Youness, S. (2000). Professional Data Warehousing with SQL Server 7.0 and OLAP Services. Birmingham: Wrox Press Ltd.