Incorporating Completeness Quality Support in Internet Query Systems

Sandra de F. Mendes Sampaio, and Pedro R. Falcone Sampaio

School of Informatics, University of Manchester,
Manchester M60 1QD
{S.Sampaio, P.Sampaio}@Manchester.ac.uk

1 Introduction

There has been an exponential growth in the availability of data on the web and in the usage of systems and tools for querying and retrieving web data. Despite the considerable advances in search engines and other internet technologies for dynamically combining, integrating and collating web data, supporting a DBMS-like data management approach across multiple web data sources is still an elusive goal. To buck this trend, internet query systems – IQS [1] are being developed to enable DBMS-like query processing and data management over multiple web data sources, shielding the user from complexities such as information heterogeneity, unpredictability of data source response rates, and distributed query execution.

The comprehensive query processing approach supported by IQS allows users to query a global information system without being aware of the sites structure, query languages, and semantics of the data repositories that store the relevant data for a given query [1]. Despite the significant amount of work in the development of the data integration and distributed query processing capabilities, internet query systems still suffer from inadequate data quality control mechanisms to address the management of quality of the data retrieved and processed by the IQS. Typical examples of data quality issues [2] that need to be addressed when supporting quality aware query processing over multiple web data sources are: Accuracy, Completeness and Timeliness.

We are currently investigating how an internet query system can be extended to support a dynamic data quality aware query processing framework. In particular, we are developing Completeness extensions for the Niagara Internet Query System.

2 Measuring Model and Data Completeness of XML Data

Completeness is a context-dependent data quality dimension that refers to "the extent to which data are of sufficient breadth, depth and scope for the task at hand" [4]. In the context of a database model, two types of completeness dimensions are considered: model completeness and data completeness. Model completeness refers to the measure of how appropriate the schema of the database is for a particular application; data completeness refers to the measurable errors of omission observed

between the database and its schema, checking, for example, if a database contains all entities/attributes specified in the schema. Completeness issues arising in database applications may have several causes, for example, discrepancies between the intent for information querying and the collected data, partial capture of data semantics during data modeling, and the loss of data resulting from data exchange. Potential approaches to address completeness issues include removing entities with missing values from the database; replacing missing values with default values, and completing missing values with data from other sources. Irrespective of the approach taken to deal with poor data completeness, it is crucial that database users formulating queries across multiple data sources are able to judge if a particular query result is "fit" for its purpose, by measuring the level of completeness of the result.

3 Tagging Completeness Information to Data

To enable quality aware query processing, data sources should provide quality information relating to each stored XML document, e.g., the number of missing elements/attributes in the document, the expected total of elements/attributes, as well as the number of missing instance values, and the expected total of instance values, required to measure model completeness and data completeness for the document. The information needs to be tagged and delivered to the Internet Query System mediator so that quality assessment query processing takes place. Figure 3.1 illustrates the mechanism for tagging quality information to XML data. We have adapted the mechanism proposed in [5] for tagging data quality information on relational data.

```
<!ELEMENT carDealerInformation (dealer*, dataQuality)>
<!ELEMENT dealer (name, car*)>
<!ATTLIST dealer id ID #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ELEMENT car (model, price)>
<!ELEMENT model (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT dataQuality (completeness)>
<!ELEMENT completeness (numberElements, missingElements, numberValues, missingValues)>
<!ELEMENT numberElements (#PCDATA)>
<!ELEMENT missingElements (numberMissingElem, elem*)>
<!ELEMENT numberValues (#PCDATA)>
<!ELEMENT missingValues (numberMissingVal, elem*)>
<!ELEMENT numberMissingElem (#PCDATA)>
<!ELEMENT numberMissingVal (#PCDATA)>
<!ELEMENT elem (name, number)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT number (#PCDATA)>
```

Fig 3.1 XML Data Tagging Mechanism.

4 Quality Aware Algebraic Query Processing

The quality aware query processing implementation framework described in this paper is being developed as an extension to the Niagara IQS algebraic operators. When a query is submitted to Niagara as an XML-based query expression, it is transformed into two sub-queries, a search engine query and a query engine query.

While the former is used by the search engine to select the data sources that are relevant to answer the query, the latter is optimized and ultimately mapped into a quality aware algebraic query execution plan that incorporates algebraic operators addressing completeness information. Following data source selection, the process of fetching data takes place, and streams of data start flowing from the data sources to the site of the Internet Query System for query execution. This process is illustrated in Figure 4.1.

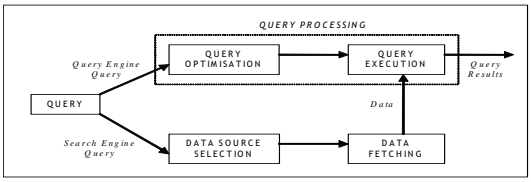

Fig 4.1 Query Processing and Data Search in Niagara.

The Completeness Algebra whose operators compose a query execution plan is an XML algebra extended with an operator that encapsulates the capability of measuring completeness quality of XML data based on completeness factors tagged on the data. The algebraic query processing framework adopted in our implementation extends algebraic quality operators developed for relational systems [5] to devise an XML-based algebra for the Niagara IQS that can take into account completeness quality information during query execution. The Completeness Algebra is similar to an XML-algebra, but it has an additional operator, the Completeness operator, which encapsulates functions for measuring, inserting and propagating completeness information in XML data, provided the data has completeness factors associated with it (IQR tags).

5 Related Work

In [9] an approach for data quality management in Cooperative Information Systems is described. The architecture has as its main component a Data Quality Broker, which performs data requests on all cooperating systems on behalf of a requesting system. The request is a query expressed in the XQuery language along with a set of quality requirements that the desired data have to satisfy. A typical feature of cooperative query systems is the high degree of data replication, with different copies of the same data received as responses. The responses are reconciled and the best results (based on quality thresholds) are selected and delivered to users, who can choose to discard output data and adopt higher quality alternatives. All cooperating systems export their application data and quality data thresholds, so that quality certification and diffusion are ensured by the system. The system, however, does not adopt an algebraic query processing framework and is not built on top of a mainstream IQS.In [8], data quality is incorporated into schema integration by answering a global query using only queries that are classified as high quality and executable by a subset of the data sources. This is done by assigning quality scores to queries based on previous knowledge about the data to be queried, considering quality

dimensions such as completeness, timeliness and accuracy. The queries are ranked according to their scores and executed from the highest quality plan to the lowest quality plan until a stop criteria is reached. The described approach, however, does not use XML as the canonical data model and does not address physical algebraic query plan implementation issues.

6. Conclusions and Future Work

With the ubiquitous growth, availability, and usage of data on the web, addressing data quality requirements in connection with web queries is emerging as a key priority for database research [3]. There are two established approaches for addressing data quality issues relating to web data: data warehouse-based, where relevant data is reconciled, cleansed and warehoused prior to querying; and mediator-based where quality metrics and thresholds relating to cooperative web data sources are evaluated "on the fly" at query processing and execution time. In this paper we illustrate the query processing extensions being engineered into the Niagara internet query system to support mediator-based quality aware query processing for the completeness data quality dimension. We are also addressing the timeliness dimension [6] and extending SQL with data quality constructs to express data quality requirements [7]. The data quality aware query processing extensions encompass metadata support, an XML-based data quality measurement method, algebraic query processing operators, and query plan structures of a query processing framework aimed at helping users to identify, assess, and filter out data regarded as of low completeness data quality for the intended use. As future plans we intend to incorporate accuracy data quality support into the framework and benchmark the quality/cost query optimiser in connection with a health care application.

References

[1] J. Naughton, D. DeWitt, D. Maier, et al: The Niagara Internet Query System. IEEE Data Eng. Bull. 24(2): 27-33 (2001)
[2] J. Olson, Data Quality: the Accuracy Dimension, Morgan Kauffmann, 2003
[3] M. Gertz, T. Ozsu, G. Saake, K. Sattler: Data Quality on the Web, Dagstuhl Seminar, Germany, 2003
[4] Richard Wang, Stuart E. Madnick: The Inter-Database Instance Identification Problem in Integrating Autonomous Systems, Proceedings of ICDE Conference, 46-55, (1989)
[5] R. Y. Wang; M.P. Reddy; H. B. Kon, Toward Quality data: An attribute-based approach, Decision Support Systems 13 (1995), 349-372, 1995.
[6] S. F. M. Sampaio; C. Dong; P. R. F. Sampaio, Incorporating the Timeliness Quality Dimension in Internet Query Systems, WISE 2005 Workshops, LNCS 3807, pp. 53-62, 2005.
[7] C. Dong; S. F. M. Sampaio; P. R. F. Sampaio: Expressing and Processing Timeliness Quality Aware Queries: The DQ^2L Approach, to appear in International Workshop on Quality of Information Systems, ER 2006 Workshops, LNCS, 2006.
[8] F. Naumann, U. Lesser; J. Freytag, Quality-driven Integration of Heterogeneous Information Systems; Proceedings of the 25th VLDB Conference, Scotland, 1999
[9] M. Mecella, M. Scannapieco, A. Virgillito, R. Baldoni, T. Catarci, C. Batini, The DaQuinCIS Broker: Querying Data and Their Quality in Cooperative Information Systems. LNCS 2800. Pages: 208-232. 2003.