XML Structural Summaries

Mirella M. Moro
Univ. Fed. Rio Grande do Sul
Porto Alegre, RS, Brazil
mirella@inf.ufrgs.br

Zografoula Vagena
Microsoft Research
Cambridge, UK
zografv@microsoft.com

Vassilis J. Tsotras
University of California
Riverside, CA, USA
tsotras@cs.ucr.edu

ABSTRACT

This tutorial introduces the concept of **XML Structural Summaries** and describes their role within XML retrieval. It covers the usage of those summaries for Database-style query processing and Information Retrieval-style search tasks in the context of both centralized and distributed environments. Finally, it discusses new retrieval scenarios that can potentially be favorably supported by those summaries.

1. INTRODUCTION

A number of data structures - known as **Structural Summaries** - have been defined to compensate for the XML data repetition and lack of schema. A Structural Summary of an XML document is a dynamically generated and maintained graph structure that preserves the structural characteristics of the document in a compact form. While the most popular usage of those structures is as secondary indexes that can identify XML nodes reachable from specific path patterns, recent proposals have described different and diverse ways to explore them when dealing with XML data, considerably expanding their importance.

The purpose of this tutorial is to provide insights on the applications of Structural Summaries within XML data retrieval. It provides a detailed overview of their usage within XML query processing and then describes the employment of the summaries within other retrieval contexts, such as search systems, distributed environments and stream applications. Finally, it discusses new retrieval scenarios that can potentially further expand the usage of those structures. At the end of the presentation, the audience will be familiar with a very useful data structure that has gained large use for processing XML data and is showing its versatility for processing other types of data (e.g. data streams).

Target Audience. The target audience is VLDB attendees interested in understanding a very useful data structure that has gained large use for processing XML data and is showing its versatility for processing other types of data (e.g. data streams). We do not assume any background and we will provide broad coverage on many key concepts, making it appropriate for graduate students seeking new areas to study and researchers working on similar issues.

PVLDB '08, August 23-28, 2008, Auckland, New Zealand
Copyright 2008 VLDB Endowment, ACM 978-1-60558-306-8/08/08

2. TUTORIAL OUTLINE

Introduction and Motivation. An XML document is formed by a sequence of elements that enclose text values and other elements. While knowledge of the underlying structure of the XML data is indispensable, this information is frequently missing (e.g. when the high variations and irregularities in the structure of the data cannot be effectively captured with a schema). This lack of schema motivated the research for Structural Summaries [2, 7, 8, 10, 12].

Structural Summaries. The intuition for using structural summaries is: the document underlying structure can be captured with a structure whose size is much smaller (e.g. many nodes have a common access path pattern). Hence, operations over the data (e.g. query processing) are more efficient on the summary instead of on the whole document.

Structural summaries encode the entire document into an index (as opposed to individual paths or values). The work on Representative Objects [14] provides the theoretical foundations to the concept of dynamically generated structural summaries directly from graph-structured databases. The Strong DataGuide [8] provides a concise implementation of the ideas presented in [14].

The size of the structural summary can be large (mostly for heterogeneous documents), causing the performance to degrade. In order to keep the size of the index small and manageable, approximate versions may be constructed by relaxing some of the structural constraints. However, such an approximation may cause false positives, i.e. the techniques trade size with precision. The pruning or approximation policies (used to compact the summaries) may differ, but all of them provide that no result is missed. For example, the A(k)-indexes [10] correspond to a family of approximate structural summaries that employ bisimilarity to identify the groups of equivalent data nodes. Likewise, workload-aware pruning techniques, such as APEX [4], are also employed for further pruning.

This part of the tutorial presents the main techniques for processing Structural Summaries. It focuses on those approaches that consider only the document structure to build the summary: DataGuides [8], A(k)-indexes [10], Suffix Trees [11].

XML Query Processing. The efficiency of the structural summaries on XML query processing has been established and widely accepted. First, the structural summaries can be employed as an indexing technique. In this context, DataGuides, A(k)-indexes and suffix trees can be employed for processing path expression queries. Second, in the context of extracting schema from semistructured and XML data, the problem of identifying the structure of the data has also been investigated [2, 7]. Third, the use the summaries as partitioning method (or a data clustering technique) has recently appeared [1, 3, 17].

This part of the tutorial presents different techniques for using structural summaries while processing XML queries. It also

presents the drawbacks that the summaries may present according to a study from [13].

Other Retrieval Contexts. This part of the tutorial presents different research works that employ the summaries within other retrieval contexts: keyword search queries on XML data can be evaluated using structural summaries as graph schema (e.g. XKeyword [9]); heterogeneous summaries may be defined by creating partitions according to explicit criteria obtained from an expression in the complete XPath language (e.g. DescribeX [5]); publish-subscribe systems may also take advantage of structural summaries (e.g. RoXSum [15]).

From all those applications, the publish/subscribe applications offer an unique opportunity for applying the structural summaries on data stream processing. Therefore, this tutorial focuses on that application and details it better in the next section.

Stream Processing Application. Publish/subscribe systems have created opportunities for new applications such as a plethora of alert and notification services that notify users interested in new products in the market, stock price changes, better offer deals and so on. Such systems perform asynchronous message transmission from publishers to subscribers, without any of the parties having knowledge of the other. Message transmission is performed by a sophisticated overlay network of application-level, content-based routers (called message brokers), which match data messages against registered subscriptions and forward those messages based on this matching.

With the recognition of XML as the standard for data exchange, specialized, XML-aware information dissemination services become necessary [6]. Those services can be implemented as pub/sub systems in which the information to be routed is encoded using XML, and the user profiles are expressed using XML query languages. The message filtering task has received the most attention from researchers for two reasons. First, it is the most critical task for the performance of the pub/sub system. At the same time, it is the most complex task due to the tree structure of the XML data. Considering the filtering task, the use of structural summaries has been recently extended for processing XML streams [15, 16].

Specifically, [15] proposes the RoXSum (Routing XML Summary) structure as a new message (composed by XML documents) representation scheme. It also presents novel filtering algorithms that combine the advantages of content aggregation and batch processing. Then, in [16], the RoXSum structure is extended to support value predicates as well. This part of the tutorial focuses on such works and discusses how the summaries can be adapted to successfully handle the data streams on XML pub/sub systems.

Conclusions and Future Directions. More and more applications are creating, manipulating and exchanging XML data, for example Web Services, Digital Libraries and mashups. At the same time, schemas become more and more complex, heterogeneous and even obsolete. Furthermore, textual content constitutes a large part of web documents. Considering all these aspects, we claim that the importance of structural summary for XML retrieval will increase. Also, methods to better incorporate textual information are still necessary. Note that current proposals focus mostly on value predicates. However, full-text search is needed to query large proportions of text, but it must consider the structural information of the data as well.

3. FURTHER INFORMATION

Authors' Biography. Mirella M. Moro holds a Ph.D. in Computer Science (University of California Riverside - UCR). She is currently working as a posdoc at Instituto de Informatica - UFRGS (Porto Alegre, Brazil) and her research areas of interest include XML query optimization, content-based dissemination systems, version management and temporal databases.

Zografoula Vagena holds a Ph.D. in Computer Science (University of California, Riverside). She spent two years working as a Post-Doctoral Researcher at IBM Almaden Research Center (San Jose, CA, USA) and she is working at Microsoft Research. Her research interests include databases and IR; text indexing and retrieval, query processing and optimization, XML data management, preference-based query processing.

Vassilis Tsotras is a professor at the Department of Computer Science and Engineering in the University of California, Riverside. His research interests include temporal and spatiotemporal databases, sensor data, querying complex objects, evolving systems design, schema evolution, multi-version schema evolution, XML query processing, and wireless data dissemination.

Acknowledgements. The work presented here was partially supported by NSF grant IIS-0705916. Mirella M. Moro was supported by CNPq, Brazil, grant 151708/2007-0.

4. REFERENCES

[1] A. Barta, M. P. Consens, and A. O. Mendelzon. Benefits of Path Summaries in an XML Query Optimizer Supporting Multiple Access Methods. In *VLDB*, 2005.

[2] P. Buneman et. al. Adding Structure to Unstructured Data. In *ICDT*, 1997.

[3] T. Chen, J. Lu, and T. W. Ling. On Boosting Holism in XML Twig Pattern Matching using Structural Indexing Techniques. In *SIGMOD*, 2005.

[4] C.-W. Chung, J.-K. Min, and K. Shim. APEX: an adaptive path index for XML data. In *SIGMOD*, 2002.

[5] M. P. Consens and F. Rizzolo. Fast Answering of XPath Query Workloads on Web Collections. In *XSym*, 2007.

[6] Y. Diao, S. Rizvi, and M. J. Franklin. Towards an Internet-Scale XML Dissemination Service. In *VLDB*, 2004.

[7] M. F. Fernandez and D. Suciu. Optimizing Regular Path Expressions Using Graph Schemas. In *ICDE*, 1998.

[8] R. Goldman and J. Widom. DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases. In *VLDB*, 1997.

[9] V. Hristidis, Y. Papakonstantinou, and A. Balmin. Keyword Proximity Search on XML Graphs. In *ICDE*, 2003.

[10] R. Kaushik et.al. Exploiting Local Similarity for Indexing Paths in Graph-Structured Data. In *ICDE*, 2002.

[11] E. M. McCreight. A Space-Economical Suffix Tree Construction Algorithm. *Journal of ACM*, 23(2):262–272, 1976.

[12] T. Milo and D. Suciu. Index Structures for Path Expressions. In *ICDT*, 1999.

[13] M. M. Moro, Z. Vagena, and V. J. Tsotras. Evaluating Structural Summaries as Access Methods for XML. In *WWW*, 2006.

[14] S. Nestorov et.al. Representative Objects: Concise Representations of Semistructured, Hierarchial Data. In *ICDE*, 1997.

[15] Z. Vagena, M. M. Moro, and V. J. Tsotras. RoxSum: Leveraging Data Aggregation and Batch Processing for XML Routing. In *ICDE*, 2007.

[16] Z. Vagena, M. M. Moro, and V. J. Tsotras. Value-Aware RoXSum: Effective Message Aggregation for XML-Aware Information Dissemination. In *WebDB*, 2007.

[17] B. Yang et.al. Virtual Cursors for XML Joins. In *CIKM*, 2004.