Chapter 14
XML Data Integration:
Schema Extraction and Mapping

Huiping Cao
Arizona State University, Arizona, USA

Yan Qi
Arizona State University, Arizona, USA

K. Selçuk Candan
Arizona State University, Arizona, USA

Maria Luisa Sapino
University Of Torino, Torino, Italy

ABSTRACT

Many applications require exchange and integration of data from multiple, heterogeneous sources. eXtensible Markup Language (XML) is a standard developed to satisfy the convenient data exchange needs of these applications. However, XML by itself does not address the data integration requirements. This chapter discusses the challenges and techniques in XML Data Integration. It first presents a four step outline, illustrating the steps involved in the integration of XML data. This chapter, then, focuses on the first two of these steps: schema extraction and data/schema mapping. More specifically, schema extraction presents techniques to extract tree summaries, DTDs, or XML Schemas from XML documents. The discussion on data/schema mapping focuses on techniques for aligning XML data and schemas.

INTRODUCTION

Data integration is the process of combining multiple heterogeneous and autonomous data sources. Its purpose is to provide a logically unified view of the data to the users who need to search or analyze disparate data sources. Data integration is a well studied problem in the data management community (Doan & Halevy, 2005; A. Halevy, Rajaraman, & Ordille, 2006; Lenzerini, 2002). Despite decades of work in the area, however, the problem is still open. In this chapter, we focus on techniques for eXtensible Markup Language (XML) data integration. As we will see, XML provides opportunities in improving compatibilities across data sources; we will however see that XML also introduces unique challenges that require innovative solutions.

DOI: 10.4018/978-1-61520-727-5.ch014

APPLICATIONS

Many applications require effective and efficient data integration and, as the number and diversity of available data sources increase, this requirement gains further significance. In what follows, we briefly introduce a sample of contemporary applications which require data integration solutions.

- *Data warehousing and business intelligence*. A data warehouse is a repository storing large amounts of data collected from different sources (Devlin & Murphy, 1988). The primary goal of a data warehouse is to provide users unified view of (and efficient access to) data collections that were originally located at different sources. Data warehouses are especially useful in enabling large scale data analysis, for example in support of business intelligence applications. Obviously, unless the contributing data sources are identical in structure or are partitions of a single schema, to build the data warehouse, we first need to integrate the data by identifying the correspondences between the data sources and the data warehouse.
- *Peer-to-peer (P2P) systems*. P2P systems leverage autonomous data sources (peers) as if they are part of a single unified data management system (Koloniari & Pitoura, 2005; Pankowski, 2008). Common usage of such systems includes a user initiating a query through one of the autonomous peer system, but getting answers from all relevant peers. Natural challenges include identifying relevant peers across heterogeneous schema and managing the mappings among the schemas of the peers (Anand & Chawathe, 2004; Cherukuri & Candan, 2008). In addition, queries and answers need to be routed within the peers in the system in a way that eliminates redundant query processing (Anand & Chawathe, 2004).
- *Service oriented architectures (SOA) and web information integration*. Service oriented architectures abstract recurring (e.g., business) activity flows, make them available as independent services, and leverage these services as modules within large software systems. This approach reduces costs of developing and deploying new applications and promotes reuse. Consequently, today, the "web" is not only a collection of hyperlinked pages, but rather a collection of dynamic services that one can use to develop web-based applications and mashups (Jhingran, 2006). These web services, with their descriptions, are published so that other people can locate and integrate them into end-to-end information products. Meanwhile, data spaces (Franklin, Halevy, & Maier, 2005; A. Y. Halevy, Franklin, & Maier, 2006) help reduce the cost of managing loosely structured Web data by eliminating the need to impose strict structures on the integrated data. These, however, require resolving potential differences between the data service interfaces and underlying data structures.
- *Scientific data management*. In many scientific domains (e.g., archeology (Kintigh, 2006) and biology (Achard, Vaysseixm, & Barillot, 2001)), individual researchers or communities have different data management conventions, standards, and taxonomies (Qi, Candan, & Sapino, 2007). For example, bioinformatics data have many new data types (e.g., microarrays, interaction maps of proteins, etc.) stored in different databases and in different formats (Achard et al., 2001). In archaeology, there is almost no universally agreed structure or ontology to help support integration and eliminate conflicts that occur due to