XML Schema Integration and E–Commerce

Kalpdrum Passi
Laurentian University, Canada

Louise Lane
Laurentian University, Canada

Sanjay Madria
University of Missouri-Rolla, USA

Mukesh Mohania
IBM India Research Lab, India

INTRODUCTION

XML (eXtensible Markup Language) is used to describe semi-structured data, i.e., irregular or incomplete data whose structure may be subject to unpredictable changes. Unlike traditional semi-structured data, XML documents are self-describing, thus XML provides a platform-independent means to describe data and, therefore, can transport data from one platform to another (Bray, Paoli, & Sperberg-McQueen, 1998). XML documents can be both created and used by applications. The valid content, allowed structure, and metadata properties of XML documents are described by their related schema(s) (Thompson, Beech, Maloney, & Mendelsohn, 2001). An XML document is said to be *valid* if it conforms to its related schema. A schema also gives additional semantic meaning to the data it is used to tag. The schema is provided independently of the data it describes. Any given data set may rely on multiple schemas for validation. Any given schema may itself refer to multiple schemas.

In e-commerce, XML documents can be used to publish everything from product catalogs and airline schedules to stock reports and bank statements. XML forms can be used to place orders, make reservations, and schedule shipments. XML eliminates the need for custom interfaces with every customer and supplier, allowing buyers to compare products across many vendors and catalog formats, and sellers to publish their catalog information once to reach many potential buyers. XML can also enable online businesses to build on one another's published content and services to create innovative virtual companies, markets, and trading communities. With a global view of the Internet-wide shopping directories, a query system can locate all merchants carrying a specific product or service and then query each local schema in parallel to locate the best deals. The query system can sort the offers according to criteria set by the buyers–the cheapest flight, the roomiest aircraft, or some weighted combination. The traditional method used for business-to-business (B2B) information exchange is through Electronic Data Interchange (EDI), which is complex, expensive, and necessitates a custom integration solution between each pair of trading partners. A query-based system that uses XML as the common format to enterprise integration is simpler and more open than traditional EDI, as it eliminates the proprietary message formats used by each company. A complete business integration solution also requires metadata for each commerce community, a means to map each local schema into an integrated global view, and a server for processing XML documents and invoking appropriate applications and services.

RELATED WORK

The problem of schema and integration of heterogeneous and federated databases has been addressed widely. Several approaches to schema integration exist as described in Batini, Lanzerini, and Navathe (1986); Behrens, 2000; Christophides, Cluet, and Simon, (2000); Haas, Miller, Niswanger, Roth, Schwarz, and Wimmers (1999); Miller, Ioannidis, and Ramakrishnan (1993); Parent and Spaccapietra (1998); and Ram and Ramesh (1998). A global schema in the general sense can be viewed as a regular schema, the rules of which encompass the rules of a common data model. A global schema eliminates data model differences and is created by integrating local schemas. The creation of a global schema also helps to eliminate duplication, avoid problems of multiple updates, and thus minimize inconsistencies.

Most schema integration approaches decompose integration into a multi-layered architecture like the one followed in this paper constituting *pre-integration*, *comparison*, and *integration* (Batini et al., 1986; Miller, 1998).

There have been some recent systems (Adali, Candan, Papakonstantinou, & Subramanian, 1996; Papakonstantinou, Garcia-Molina, & Widom, 1995; Tomasic, Raschid, & Valduriez, 1996) that integrate data from multiple sources. Most of these systems provide a set of mediated/global schema(s). Some systems like *Garlic* (Roth & Schwarz, 1997) use wrappers to describe the data from different sources in its repositories and provide a mechanism for a middleware engine to retrieve the data. The *Garlic* system also builds global schema from the individual repositories. The *comparison* and *restructuring* phase of integration is handled in some systems through human interaction using a graphical user interface as in Clio (Hernandez, Miller, Haas, Yan, Ho, & Tian, 2001; Miller, Haas, & Hernandez, 2000; Miller et al., 2001; Yan, Miller, Haas, & Fagin, 2001) and in others semi-automatically through machine learning techniques such as in *Tukwila* data integration system at University of Washington. The *Tukwila* integration system reformulates the user query into a query over the data sources, which are mainly XML documents corresponding to DTD schemas and relational data.

INTEGRATION REQUIREMENTS, ARCHITECTURE AND METHODOLOGY

XML Schema (Thompson et al., 2001) has recently been recommended as the standard schema language to validate XML documents. It has a stronger expressive power than the DTD (Document Type Definition) schema for the purpose of data exchange and integration from various sources of data.

Since here we assume that the schemas to be integrated currently validate a set of existing XML documents, data integrity and continued document delivery are chief concerns of the integration process, thus closely linking XML Schema integration to the theoretical requirements and process of database integration.

Satisfying the condition that the global schema meets the requirements of all the initial schemas is the most difficult part of integration, and is worsened by data model heterogeneity.

We define an object-oriented data model that we call XSDM (XML Schema Data Model) for the purpose of XML Schema integration. We use the three-layered architecture of *pre-integration*, *comparison*, and *integration* to achieve XML Schema integration.

The XML Schema integration process developed in this paper uses a *one shot n-ary* (Batini et al., 1986) strategy. The *one shot n-ary* style integrates all the initial schemas at once. Schema integration should be both *extensible* and *scalable*. It should be easy to add or remove sources of data (i.e., schemas), to manage large numbers of schemas, and to adjust the resulting global schema. With the XML Schema integration approach, multiple schemas can be integrated at one time.

Any global integrated schema must meet the following three criteria: *completeness*, *minimality*, and *understandability*. In order to meet the first criteria of *completeness*, all the elements in the initial schemas should be in the merged schema. The merged schema can be used to validate any of the XML instance documents that were previously validated by one of the initial schema specifications. To satisfy the second criterion, *minimality*, each unique element is defined only once in the schema. Redundancy is eliminated wherever possible through the identification of equivalent elements and attributes, and the subsequent use of substitution groups. Datatypes for terminal elements are expanded through the use of constraint facet redefinition, or unions of incompatible datatypes, only to the point necessary to satisfy boundary conditions. Optionality of elements (i.e., minOccurs and maxOccurs values) is expanded to meet boundary restrictions only. Finally, to comply with the third criterion, *understandability*, in the case of XML Schema integration, the global schema is formulated in a referenced style, rather than an inline style (nested definitions), for ease of reading and assessment.

During *pre-integration*, an analysis of the schemas to be integrated occurs. Priority of integration is determined if the process is not to be *one shot*. Preferences may be given to retaining the entire or certain portions of schemas as whole parts of the global schema. Designer interaction occurs in view integration as assertions of relationships and constraints of elements are discovered.

During the *comparison* stage of integration, correspondences as well as conflicts between elements are identified. There are four *semantic relationships* defined by Batini et al. (1986). The schematic representations can be viewed as *identical*, *equivalent*, *compatible*, or *incompatible*. We identify six types of semantic relationships, which apply to XML Schema elements – *identical*, *equal*, *equivalent*, *subset*, *unique*, and *incompatible*.

The fundamental activity in the *comparison* phase of integration is *conflict resolution*. Conflict identification and resolution is central to successful integration. *Naming conflicts*, *datatype conflicts* & *scale differences*, and *structural conflicts* can occur during XML Schema Integration.

During the *conformance* phase, the *semantic relationships* and *conflicts* identified in the comparison phase are resolved. Initial schemas may be transformed in order to make them more suitable for integration. The XML