

USING XML SCHEMA FOR PLATFORM-NEUTRAL STRUCTURE TRANSFER

Dr. P V Virparia, Nehal Daulatjada, Priya Swaminarayan,
Dr. V R Rathod

ABSTRACT

Extensible Markup Language (XML) is a simple, very flexible text format, originally designed to meet the challenges of large-scale electronic publishing. XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere[1]. XML's design goals emphasize simplicity, generality, and usability over the Internet[2]. Although XML's design focuses on documents, it is widely used for the representation of arbitrary data structures through the use of Schema. XML provides a widely adopted standard for representing text and data in a format that can be processed without much human or machine intelligence. Information formatted in XML can be exchanged across platforms, languages, and applications, and can be used with a wide range of development tools and utilities. Looking at the huge list of applications and advantages of using XML over proprietary relational databases for platform-neutral data representation, here is an attempt to study and evaluate the XML Schema representation of Microsoft Access 2003® and its ability to transfer all the properties of the table.

Keywords: XML, Schema, XML converter, Microsoft Access 2003.

1. INTRODUCTION

XML is a set of rules for encoding documents electronically. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all gratis open standards [3]. Microsoft Access 2003® combines the relational Microsoft Jet Database Engine with a graphical user interface and software development tools. Access stores data in its own format based on the Access Jet Database Engine. It can also import or link directly to data stored in other Access databases, Excel, SharePoint lists, text, XML and many more. In Microsoft Access 2003®, you can output data to XML by exporting to an XML document. You can also export a schema to specify the structure of the XML document as well as a stylesheet to describe the presentation of the data.

2. EXTENSIBLE MARKUP LANGUAGE:

XML is a standard, simple, self-describing way of encoding both text and data so that content can be processed with relatively little human intervention and exchanged across diverse hardware, operating systems, and applications[4]. Moving information across the Internet and between software applications has always been difficult due to differences in data formats and proprietary

structures. HTML is limited in its ability to define data and data structures. XML defines the data and describes how the data should be structured. XML is a data interchange format in that it allows you to exchange data between dissimilar systems or applications. XML separates the data from the presentation so that the same XML data can be presented in multiple ways by using different presentation files.

XML uses the tags only to delimit pieces of data, and leaves the interpretation of the data completely to the application that reads it. There are many programming interfaces that software developers may use to access XML data, and several schema systems designed to aid in the definition of XML-based languages. As of 2009[update], hundreds of XMLbased languages have been developed[5], including RSS, Atom, SOAP, and XHTML. XML-based formats have become the default for most office-productivity tools, including Microsoft Office (Office Open XML), OpenOffice.org (OpenDocument), and Apple's iWork[6]. Relational database systems cannot meet all the demands of electronic business because they process data independently of its context. Traditional databases may be well suited for data that fits into rows and columns, but cannot adequately handle rich data such as audio, video, nested data structures or complex documents, which are characteristic of typical Web content. To deal with XML, traditional databases are typically retrofitted with external conversion layers that mimic XML storage by translating it between XML and some other data format like the XML Converter facility with Microsoft Access 2003®. This conversion is error-prone and results in a great deal of overhead, particularly with increasing transaction rates and document complexity. XML databases, on the other hand, store XML data natively in its structured, hierarchical form. Queries can be resolved much faster because there is no need to map the XML data tree structure to tables. This preserves the hierarchy of the data and increases performance[7].

2.1 Here are a few reasons of using XML:

Self-describing data[19], Complete integration of all traditional databases and formats, Modifications to data presentation - no reprogramming required, One-server view of distributed data, Internationalization, Open and extensible, Future-oriented technology, Separates content from presentation, Facilitates the comparison and aggregation of data, Can embed existing data, Validation using schema languages such as XSD and Schematron, making software construction easier.

2.2 Disadvantages of XML[9]:

XML syntax is redundant or large relative to binary representations of similar data; The redundancy may affect application efficiency through higher storage, transmission and processing costs; XML syntax is too verbose relative to other

alternative 'text-based' data transmission formats; No intrinsic data type support: XML provides no specific notion of "integer", "string", "boolean", "date", and so on; The hierarchical model for representation is limited in comparison to the relational model or an object oriented graph; Expressing overlapping (non-hierarchical) node relationships requires extra effort; XML namespaces are problematic to use and namespace support can be difficult to correctly implement in an XML parser; XML is commonly depicted as "selfdocumenting" but this depiction ignores critical ambiguities; XML does not support data types in any meaningful sense of the word[17].

3. XML AND THE ENTERPRISE DATABASE INFRASTRUCTURE TODAY

The enterprise database infrastructure today consists largely of high-end transaction processing systems such as Adabas and IMS on one hand and RDBMS-based information systems on the other. Over the last few years, however, there has been an explosion of demand for access to these enterprise databases over the internet, and also to use them to support electronic business operations. These include transactions between systems within an enterprise ("enterprise integration"), between businesses in a supply chain ("B2B e-commerce") and directly to customers ("B2C e-commerce"). The initial reaction of most companies was to integrate these diverse operations by building or buying software such as "application servers" that employed protocols such as DCOM or CORBA to perform such integration. More recently, XML offers the option of performing the necessary integration by exchanging standardized data. The SQL standard defines a vendor-neutral way of finding and extracting data from an RDBMS, so it is quite possible to write middleware applications that extract data from one database and put it in another. Nevertheless, these applications must be relatively complex and pay careful attention to the database schemas on each side in order to work properly. XML, on the other hand, is simply a standardized "meta-format" that can represent any kind of data, and for which precise schema definitions are optional[8].

3.1 XML schema:

The Semantic Web, with machine processable information contents, will only be possible when further levels of interoperability are established. Standards must be defined not only for the syntactic form of documents, but also for their semantic content. Such semantic interoperability is facilitated by recent W3C standardization efforts, notably XML/XML Schema and RDF/RDF Schema[20]. The purpose of an XML Schema is to define the legal building blocks of an XML document, just like a DTD. An XML schema provides a means of creating a set of rules that your computer can use to identify document rules governing the

validity of the XML documents that you create. Schemas provide a means of defining the structure, content, and semantics of XML documents that can be shared between many different types of computers and documents[18]. An XML schema is a description of a type of XML document, typically expressed in terms of constraints on the structure and content of documents of that type, above and beyond the basic syntactical constraints imposed by XML itself. These constraints are generally expressed using some combination of grammatical rules governing the order of elements, Boolean predicates that the content must satisfy, data types governing the content of elements and attributes, and more specialized rules such as uniqueness and referential integrity constraints.

There are languages developed specifically to express XML schemas: The Document Type Definition (DTD) language, XML Schema, Schematron and RELAX NG. The mechanism for associating an XML document with a schema varies according to the schema language. The association may be achieved via markup within the XML document itself, or via some external means[12].

4. AN XML SCHEMA[10]:

It defines elements that can appear in a document; defines attributes that can appear in a document; defines which elements are child elements; defines the order of child elements defines the number of child elements; defines whether an element is empty or can include text; defines data types for elements and attributes; defines default and fixed values for elements and attributes. Features of XML Schema[11]: They are Successors of DTDs; Support for Datatypes; Uses XML syntax; Secure data communication; Extensible; Well-formedness and validation.

5. STUDY OF RELATIONAL DATABASE: MICROSOFT ACCESS 2003®

A database is a collection of objects that allow you to store data, organize it and retrieve it in any way you want. What this means is that, with Microsoft Access 2003® you create structures called tables that allow you to organize the data so that it's easy to find later[13]. Microsoft Access 2003® is a computer application used to create and manage computer based databases on desktop computers and/or on connected computers (a network). Microsoft Access 2003® can be used for personal information management, in a small business to organize and manage data, or in an enterprise to communicate with servers[14]. 5. 1 Microsoft Access 2003® XML Converter Facility: Microsoft Access 2003® provides ways to both import and export XML data as well as transforms the data to and from other formats using XML related files. You use XML schemas to describe the structure of data in a common format that customers, other Web browsers, and

any number of XML-enabled software programs can recognize. Specifically, schemas define the rules of an XML data document, including element names and data types, which elements can appear in combination, and which attributes are available for each element. Schemas provide a model for an XML data document which defines the arrangement of tags and text within all documents referencing the schema. Microsoft Access 2003® supports the XML Schema standard (XSD). You can:

1. Import XML schema data to a Jet, SQL Server, or Microsoft SQL Server 2000® Desktop Engine database.
2. Use an Extensible Stylesheet Language Transformation (XSLT) file to transform the data into an XML format that Access supports.
3. Export data to an XML file and, optionally, use an XSLT to transform the data to another format.
4. Export the data schema using XML Schema standard (XSD).

6. OUR OBSERVATIONS:

For this study, the same DBMS was used for both exporting and importing the structure. During our study of the XML and XSD, Each of the Field's datatypes were considered. For every Field datatype, all the applicable properties as depicted in Table-1 were applied. Each of the tables were then exported as .XML and .XSD files. When each of the field properties were set and the table was exported as .XSD and again imported back into Microsoft Access 2003®, following were our observations:

1. When the import on .XML was performed, the data got correctly imported.
2. The Export facility supported exporting data to .XML and Structure to .XSD.
3. The import facility used .XML to import data using the .XSD structure.
4. The import was unable to import all the properties of all the fields back even when the export and import were done with the same database. The observations of the import are shown in the table-2.

7. FURTHER WORK:

This was a test to conform whether the .XSD schema was sufficient enough to carry the database structure back to itself. It remains to be tested as to what details are imported back in various other Relational database management systems(RDBMS) that are available on market to conclude that the XML schema structure provided by Microsoft is reliable for structure transmission.

8. CONCLUSION:

Looking at the results of export done from Microsoft Access 2003® and import done back into Microsoft Access 2003®, both on the same system, the XML

schema representation does not fully support the transmission of the table structure details. XML has been widely used for data transfers, which was evident from the import conducted. A schema representation, which includes properties and attributes constituting all the available popular RDBMSs need to be undertaken to fully qualify for semantic web openness. As of now, it is only partially supported.

Table-1: List of Field Types and Applicable properties in Microsoft Access 2003®

	Text	Memo	Number	Date/Time	Currency	Autonumber	Yes/No	Ole Object	Hyperlink
Field Size	✓		✓			✓			
Format	✓	✓	✓	✓	✓	✓	✓		✓
Input Mask	✓		✓	✓	✓				
Caption	✓	✓	✓	✓	✓	✓	✓	✓	✓
Default Value	✓	✓	✓	✓	✓		✓		✓
Validation Rule	✓	✓	✓	✓	✓		✓		✓
Validation text	✓	✓	✓	✓	✓		✓		✓
Required	✓	✓	✓	✓	✓		✓	✓	✓
Allow zero length	✓	✓							✓
Indexed	✓	✓	✓	✓	✓	✓	✓		✓
Unicode Compression	✓	✓							✓
IME Mode	✓	✓		✓					✓
IME Sentence Mode	✓	✓		✓					✓
Smart Tags	✓	✓	✓	✓	✓	✓			✓
Decimal Places			✓		✓				
New Values						✓			
Primary Key	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table-2: List of Fields properties available after importing the .XSD

	Text	Memo	Number	Date/Time	Currency	Autonumber	Yes/No	Ole Object	Hyperlink
Field Size	✓		✓			✓			
Required	✓	✓	✓	✓	✓			✓	✓
Indexed	✓	✓	✓	✓	✓	✓	✓		✓
Primary Key	✓	✓	✓	✓	✓	✓	✓		✓

REFERENCES:

1. World Wide Web Consortium site : www.w3.org/XML [accessed on 27 Dec 2009]

2. World Wide Web Consortium site : www.w3.org/TR/REC-xml/#sec-origin-goals [accessed on 5 April 2010]
3. World Wide Web Consortium site : www.w3.org/Consortium/Legal/2002/copyrightdocuments-20021231 [accessed on 4 April 2010]
4. Software Ag's Community site : communities.softwareag.com/ecosystem/communities/public/Developer/webmethods/products/tamino/faq/XMLStarter/XMLBasics.html [accessed on 22 Feb 2010]
5. [5] OASIS's Cover Pages Site: xml.coverpages.org/xmlApplications.html [accessed on 13 Jan 2010]
6. [6] Apple's Developer site : developer.apple.com/mac/library/documentation/AppleApplications/Conceptual/iWork2- [accessed on 29 March 2010]
7. Software Ag's Community site : communities.softwareag.com/ecosystem/communities/public/Developer/webmethods/products/tamino/faq/XMLStarter/WHYXML.html [accessed on 23 Feb 2010]
8. Software Ag's Community site : communities.softwareag.com/ecosystem/communities/public/Developer/webmethods/products/tamino/faq/XMLStarter/XMLDatabase.html [accessed on 23 Feb 2010]
9. Techmynd's site : www.techmynd.com/advantages-disadvantages-of-xml/ [accessed on 25 Jan 2010]
10. [w3schools site : www.w3schools.com/schema/schema_intro.asp [accessed on 30 April 2010]
11. w3schools site : www.w3schools.com/schema/schema_why.asp [accessed on 29 April 2010]
12. Wikipedia site : en.wikipedia.org/wiki/XML_schema [accessed on 1 May 2010]
13. Profsr's Free IT training site : www.profsr.com/access/acless0.htm [accessed on 2 May 2010]
14. FunctionX site : www.functionx.com/access/Lesson01.htm [accessed on 11 Jan 2010]
15. Database solutions & downloads for Microsoft Access site : www.databasedev.co.uk/fields_datatypes.htm [accessed on 10 Jan 2010]
16. Microsoft Office online site : office.microsoft.com/en-us/access/ [accessed on 9 Jan 2010]
17. Bourret's site : www.rpbourret.com/xml/XMLAndDatabases.htm [accessed on 11 Jan 2010]
18. Williamson. H., XML-The complete reference, Pg.428, TMH
19. Harold.E.R, XML Bible-Gold Edition,, Pg.7
20. Decker.S, Harmelen.F, Broekstra.J, Erdmann.M, Fensel.D, Horrocks.I, Klein.M, Melnik.S, "The Semantic Web – on the respective Roles of XML and RDF" retrieved from http://www.mpiinf.mpg.de/departments/d5/teaching/ws01_02/proseminarliteratur/decker00semantic.pdf [accessed on 1 May 2010].