AN EXPLORATION OF

XML IN DATABASE MANAGEMENT SYSTEMS

By

Dare Obasanjo

Introduction: XML and Data

XML stands for eXtensible Markup Language. XML is a meta-markup language developed by the World Wide Web Consortium(W3C) to deal with a number of the shortcomings of HTML. As more and more functionality was added to HTML to account for the diverse needs of users of the Web, the language began to grow increasingly complex and unwieldy. The need for a way to create domain-specific markup languages that did not contain all the cruft of HTML became increasingly necessary and XML was born.

The main difference between HTML and XML is that whereas in HTML the semantics and syntax of tags is fixed, in XML the author of the document is free to create tags whose syntax and semantics are specific to the target application. Also the semantics of a tag is not tied down but is instead dependent on the context of the application that processes the document. The other significant differences between HTML and XML is that the an XML document must be well-formed.

Although the original purpose of XML was as a way to mark up content, it became clear that XML also provided a way to describe structured data thus making it important as a data storage and interchange format. XML provides many advantages as a data format over others, including:

1. Built in support for internationalization due to the fact that it utilizes unicode.
2. Platform independence (for instance, no need to worry about endianess).

3. Human readable format makes it easier for developers to locate and fix errors than with previous data storage formats.
4. Extensibility in a manner that allows developers to add extra information to a format without breaking applications that where based on older versions of the format.
5. Large number of off-the-shelf tools for processing XML documents already exist.

The world of traditional data storage and XML have never been closer together. To better understand how data storage and retrievel works in an XML world, this paper will first discuss the past, present, and future of structuring XML documents. Then we will delve into the languages that add the ability to query an XML document similar to a traditional data store. This will be followed by an exploration of how the most popular RDBMSs have recognized the importance of this new data storage format and have integrated XML into their latest releases. Finally the rise of new data storage and retrieval systems specifically designed for handling XML will be shown.

Structuring XML: DTDs and XML Schemas

Since XML is a way to describe structured data there should be a means to specify the structure of an XML document. Document Type Definitions (DTDs) and XML Schemas are different mechanisms that are used to specify valid elements that can occur in a document, the order in which they can occur and constrain certain aspects of these elements. An XML document that conforms to a DTD or schema is considered to be *valid*. Below is listing of the different means of constraining the contents of an XML document.

SAMPLE XML FRAGMENT

```
<gatech_student gtnum="gt000x">
 <name>George Burdell</name>
 <age>21</age>
</gatech_student>
```

1. **Document Type Definitions (DTD)**: DTDs were the original means of specifying the structure of an XML document and a holdover from XML's roots as a subset of the Standardized and General Markup Language(SGML). DTDs have a different syntax from XML and are used to specify the order and occurence of elements in an XML document. Below is a DTD for the above XML fragment.

DTD FOR SAMPLE XML FRAGMENT

```
<!ELEMENT gatech_student (name, age)>
<!ATTLIST gatech_student gtnum CDATA>
<!ELEMENT name (#PCDATA)>
<!ELEMENT age (#PCDATA)>
```

The DTD specifies that the gatech_student element has two child elements, name and age, that contain character data as well as a gtnum attribute that contains character data.

9. **XML Data Reduced (XDR)**: DTDs proved to be inadequate for the needs of users of XML due to to a number of reasons. The main reasons behind the criticisms of DTDs were the fact that they used a different syntax than XML and their non-existent support for datatypes. XDR, a recommendation for XML schemas, was submitted to the W3C by the Microsoft Corporation as a potential XML schema standard which but was eventually rejected. XDR tackled some of the problems of DTDs by being XML based as well as supporting a number of datatypes analogous to those used in relational database management systems and popular programming languages. Below is an XML schema, using XDR, for the above XML fragment.

```
10.   XDR FOR SAMPLE XML FRAGMENT
11.
12.    <Schema name="myschema" xmlns="urn:schemas-microsoft-com:xml-
   data"
13.                          xmlns:dt="urn:schemas-microsoft-
   com:datatypes">
14.     <ElementType name="age" dt:type="ui1" />
15.     <ElementType name="name" dt:type="string" />
16.     <AttributeType name="gtnum" dt:type="string" />
17.     <ElementType name="gatech_student" order="seq">
18.      <element type="name" minOccurs="1" maxOccurs="1"/>
19.      <element type="age" minOccurs="1" maxOccurs="1"/>
20.      <attribute type="gtnum" />
21.     </ElementType>
22.    </Schema>
23.
```

The above schema specifies types for a name element that contains a string as its content, an age element that contains an unsigned integer value of size one byte (i.e. btw 0 and 255), and a gtnum attribute that is a string value. It also specifies a gatech_student element that has one occurence each of a name and an age element in sequence as well as a gtnum attribute.

24. **XML Schema Definitions (XSD)** : The W3C XML schema recommendation provides a sophisticated means of describing the structure and constraints on the content model of XML documents. W3C XML schema support more datatypes than XDR, allow for the creation of custom data types, and support object oriented programming concepts like inheritance and polymorphism. Currently XDR is used more widely than than W3C XML schema but this is primarily because the XML Schema recommendation is fairly new and will thus take time to become accepted by the software industry.

```
25.   XSD FOR SAMPLE XML FRAGMENT
26.
27.    <schema xmlns="http://www.w3.org/2001/XMLSchema" >
28.     <element name="gatech_student">
29.      <complexType>
30.       <sequence>
31.        <element name="name" type="string"/>
32.        <element name="age" type="unsignedInt"/>
33.       </sequence>
34.       <attribute name="gtnum">
```