International Journal of Information Technology and Knowledge Management
January-June 2012, Volume 5, No. 1, pp. 135-139

INFORMATION MANAGEMENT WITH APPLICABILITY USING XML

Sudesh Kumar[1], and Ela Kumar[2]

ABSTRACT: XML Based Information Management is a revolutionary concept and has eliminated (or at least reduced) the dependency on relational databases like Oracle, DB2, Sybase etc. It has also emerged as a de-facto standard for developing integration between systems which don't have any common mechanisms to talk to each other. XML has been very successful in allowing one system to send data to another system and vice-versa. Plenty of tools emerged in last decades allows facilitating the transfer of XML data includes various types of parsers, formatters and validators. In addition, various industries have developed their own formats for exchanging data. Now days, XML has reduced a great deal of hassle and time, consumed in mundane tasks. And needless to say that XML has played the Key Role in all these revolutionary technologies and transformation projects in industry. This paper presents a generic scenario about how effectively the huge data can be processed for information mining using eXtensible Markup Language along with applications of XML.

Keywords: Information, DTD (Document Type Definition), Fragment, Metadata, Document

1. INTRODUCTION

An important role for XML is in managing not only documents but also the information components on which documents are based. Document management as a technology and a discipline has traditionally augmented the capabilities of a computer's file system. By enabling users to characterize their documents, which are usually stored in files, document management systems enable users to store, retrieve, and use their documents more easily and powerfully than they can do within the file system itself. In addition to these functionalities the confidentiality of document along with its authenticity is mostly in demand. With the help of XML encryption the end-to-end security of applications that require secure exchange of structured data is facilitated [16]. This encryption ensures that a message (i.e., information) has not been altered or tampered with, because the one of the strong point of XML is that it allows document owners to create their own structure and tag names [21].

With the prompt development of the Internet, the requirement of managing information based on the web becomes more and more important [12]. XML employs a tree-structured data model, and XML queries specify patterns of selection predicates on multiple elements related by a tree structure. We have so much data produced every day by multiple transactional systems and it's very crucial for us to be able to manage and utilize this data. Researchers, industries and institutions have been trying to apply innovative techniques and methodologies for last several years and come up with alternative mechanisms. An XML document can optionally have a description of its grammar attached. The grammar for an XML document is described using a mechanism known as a "Document Type Definition" (DTD). The DTD describes the allowable elements in the XML document and describes the structure of those elements. An XML document that is structured according to the rules defined in the XML specification is termed "well formed". In addition to being "well formed", an XML document can optionally be "valid". A "valid" XML document must contain a DTD, and the grammar of the document must conform to that specified in the DTD. The process of testing to make sure that an XML document conforms to the description of the grammar described in the DTD is commonly termed "validation". However, as the web became ubiquitous, the limitations of HTML became more apparent. These limitations included the inability to represent structured data, such as the hierarchical elements in a relational database. After a continuous barrage of requests for extending HTML markup tags to be included in the HTML standard the SGML Working Group of the WWW Consortium spent eleven weeks working on an extendable markup language in 1996 that could address the shortcomings of HTML. The result was the first draft of XML.

In order to make the use of XML to represent management information truly useful, an XML Vocabulary must be defined and agreed upon. The XML vocabulary for management would be produced by defining a DTD that dictated the structure or meta-model that all "valid" XML management documents must adhere to.

[1] Associate Professor Department of Information Technology, BRCM CET, Bahal (Bhiwani),

[2] School of Information and Communication Technology Gautam Buddha University Greater Noida (U.P.)
E-mail: sudeshjakhar@gmail.com

The rest of the paper is organized as follows. In the next section we present works related to the advancement in XML. Section III will provides us a motivating scenario about information management including information domain, patterns and attributes along with some applications like information management, information management in health care etc. Section IV concludes with the importance of information management and future work.

2. BACKGROUND WORK

Plenty of works have been proposed by several researchers to manage information using XML. The study reported by Liu and Murthy impacts on the various use cases of data management by XML, because XML is widely used as a format for data transfer and thus needs to be generated from relational data for business data exchange and report generation. They showed the importance of separating XMLDB applications into data centric and document centric XMLDB applications. On the other hand, a set of algorithms and structures has been reviewed in [17], and they showed their superiority when processing independent as well as interlinked XML documents. Whereas, Shahriar et al. have proposed XML Functional Dependency (XFD), specially for the purpose of XML data transformation for semantic integration of schemas with integrity constraints. Because functional dependency is one of the integrity constraints for any data model, especially in relational model. As we all know that XML-based services with flexible and intelligent structures for data expression and exchange are gaining popularity. By considering this scenario Wang and Li proposed a XML firewall on embedded network processor, because XML enables convenient data sharing, regardless of the platforms. Maarouf and Chung also proposed a method called XML Integrated Environment (XIE), which was a general-purpose service-oriented architecture for processing XML documents in a scalable and efficient fashion. As we all are aware of the security concerns, which are one of the major challenges in information management. So that, for providing a better secure environment while processing data, [2, 14, and 16] have proposed security schemes, which provides users a better and secure environment to manage information. Over the years technology enhances rapidly but still there is a need for XML due to its various application area i.e. Maintaining of the hierarchical structure of data [1], mapping the structure, content and relationship in the relational database [6], overcome of verbosity problem using data compression method [23], the incremental fragmentation validation regardless of the entire update of XML document from scratch [20] and last but not the least data among the healthcare professionals is deemed as one of the most challenging problem in eHealth, indeed data storage in health information system is mainly performed in relational databases, whereas XML is seen as the de facto standard for exchanging between relational databases [9].

3. XML AS A SOURCE OF DATA MANAGEMENT

Information management is generally achieved in XML using data management activities, so that XML can also make a useful source of management data. For devices that currently have no way of supplying management information it may be the case, that XML is a convenient mechanism for surfacing this data.

It may also be the case that as more XML vocabularies are defined that some of this information might be useful from a management perspective. It should be possible to provide mappings between these XML vocabularies and the vocabulary used for management by the use of XSL style-sheets. XML expresses information using four basic components -- tags, attributes, data elements, and hierarchy. Each of these components serves a unique purpose; each represents a different "dimension" of information.

(A) Information Domain

XML allows us to model information systems in a natural and intuitive way. This is because XML allows us to express information in ways that better match the way we do business. We now have an information-modelling mechanism that allows us to characterize what we want to do, rather than how we have to do it. XML simply does a much better job of reflecting the way the real world operates than the data-modelling mechanisms that preceded it. XML brings a number of powerful capabilities to information modelling:

- Heterogeneity: Where each "record" can contain different data fields. The real world is not neatly organized into tables, rows, and columns. There is great advantage in being able to express information, as it exists, without restrictions.
- Extensibility: Where new types of data can be added at will and don't need to be determined in advance. This allows us to embrace, rather than avoid change.
- Flexibility: Where data fields can vary in size and configuration from instance to instance. XML imposes no restrictions on data; each data element can be as long or as short as necessary.

XML is also self-describing and informationally complete; applications can use this feature to automatically build themselves with little or no programming required. Companies such as BEA, TIBCO, and Microsoft offer frameworks for building applications, with a minimum of effort, that use XML as the basis for expressing information. In environments like these, XML becomes a universal information structuring tool where system components no longer need to be programmed separately as discreet silos of functionality.

(B) Patterns in XML

In order to effectively model information using XML, we must learn how to identify the natural patterns inherent to it. First, we must determine whether we have used XML elements properly. To do this we will analyse the XML fragment shown in Listing 1.

Listing 1: Example XML fragment

```
<colorimeter_reading>
  <device> X-Rite Digital Swatchbook </device>
  <patch> cyan </patch>
  <RGB resolution=8>
      <red> 0 </red>
      <green> 255 </green>
      <blue> 255 </blue>
  </RGB>
</colorimeter_reading>
```

We examine each data element and ask the following question:

- Is this data, or is it actually metadata (information about another data element)? We examine every attribute and ask the following questions:
- Does the attribute tell us something about or describe how to interpret, use, or present data elements?
- Is the attribute truly metadata, and not actually a data element?
- Does it apply to all data elements in its scope?

We examine every tag and ask the following question:

- Does this tag help describe what all data elements in its scope are?

We examine the groupings we have created (the sibling relationships)

- Are all members of the group related in a way the parent nodes describe?
- Is the relationship between siblings unambiguous?

If the answer to any of the preceding questions is “no”, then we need to cast the offending components differently.

After insuring that information has been expressed using the components of XML appropriately, we examine how everything has been stitched together. To do this we create an information context list from the XML fragment. This is done by simply taking each data element and writing down every tag and attribute leading up to it. The resulting lines will give us a flattened view of the information items contained in the XML fragment. A context list for the example XML fragment in Listing 1 would look like the one shown in Listing 2.

Listing 2: Context list for example XML fragment

```
<colorimeter_reading><device> X-Rite Digital Swatchbook
<colorimeter_reading><patch> cyan
<colorimeter_reading><RGB resolution=8><red> 0
<colorimeter_reading><RGB resolution=8><green> 255
<colorimeter_reading><RGB resolution=8><blue> 255
```

If we convert these lines to what they mean in English, we can see that each information item, and its context, makes sense and is contextually complete:

1. This colorimeter reading is from an X-Rite Digital Swatch book.
2. This colorimeter reading is for a patch called cyan.
3. This colorimeter reading is RGB-red and has an 8-bit value of 0.
4. This colorimeter reading is RGB-green and has an 8-bit value of 255.
5. This colorimeter reading is RGB-blue and has an 8-bit value of 255.

(C) Attributes Used as Data Elements

Attributes should be used to describe how to interpret data elements, or describe something about them -- in other words, attributes are a form of metadata. They are often used to contain data elements, and that runs counter to the purpose of attributes.

Listing Contains no data elements from readings at all; the attributes apply to nothing. Attributes that apply to nothing, obviously, describe how to interpret nothing.

Listing 3: XML with no data elements

```
<colorimeter_reading>
  <device> X-Rite Digital Swatchbook </device>
  <patch> cyan </patch>
  <RGB resolution=8 red=0 green=255 blue=255 />
</colorimeter_reading>
```

If we examine each attribute, especially the data portion (the part to the right of the equal sign), we can determine whether they actually represent data, or metadata:

- Resolution=8: This is a true attribute because the value does not mean anything by itself; rather it is an instruction for interpreting data elements, and therefore it is metadata.

- Red=0: This is clearly actually data because it is a reading from the colorimeter; moreover, in order to be correctly interpreted, it requires the "resolution=8" attribute. This attribute does not tell us how to interpret data -- it is data. Consequently it should be recast as a tag/data element pair.
- Green=255, blue=255: The previous analysis of "red=0" applies.

This brings us the reason why new level of business analysis put users to closer to data.

(D) XML Based Data Management in Data Warehousing

A large amount of data needed in decision-making processes is stored in the XML data format, which is widely used for E-commerce and Internet-based information exchange. Thus, as more organizations view the web as an integral part of their communication and business, the importance of integrating XML data in data warehousing environments is becoming increasingly high. Here we will explain how the design of a data mart can be carried out starting directly from an XML source.

Two main issues arise: on the one hand, since XML models semi-structured data, not all the information needed for design can be safely derived; on the other, different approaches for representing relationships in XML DTDs and Schemas are possible, each with different expressive power. However, XML documents can be associated with and validated against either a Document Type Definition (DTD) or an XML Schema, both of which allow the structure of XML documents to be described and their contents to be constrained. DTDs are defined as a part of the XML 1.0 Specification, while XML Schemas have recently become a W3C Recommendation. XML Schemas considerably extend the capabilities of DTDs, especially from the point of view of data typing and constraining. With DTDs or Schemas, the applications exchanging data can agree about the meaning of the tags and, in that case, XML reaches its full potential.

(E) Distributed Information Management with XML

XML and Web services are revolutionizing the automatic management of distributed information, somewhat in the same way HTML, Web browser and search engines modified human access to world wide information.

We can take an example of an Active XML that is based on embedding Web service calls inside XML documents.

The field of distributed data management has centred for many years around the relational model. More recently, the Web has simplified a worldwide (or intranet) publication of data based on HTML (the backbone of the Web) and data access using Web browsers, search engines and query forms. However, because of the inconvenience of a document model (HTML is a model of document and not a data model) and limitations of the core HTTP protocol, the management of distributed information remains cumbersome. The situation is today dramatically improving with the introduction of XML and Web services.

(F) Use of XML in Healthcare Information Management

Extensible Markup Language (XML) is an emerging Internet standard that is gaining momentum in many industries, including healthcare.

The purpose of the exchange of clinical data includes, but is not limited to provision of clinical care, support of clinical and administrative research, execution of automated transaction oriented decision logic (medical logic modules), support of outcomes research, support of clinical trials, and to support data reporting to government and other authorized third parties.

4. CONCLUSION

In order to efficiently process the information using XML there is a need to know about its framework. This paper presents a theoretical scenario about eXtensible Markup Language using listings and its various applications like data management, distributed information management and in health care etc. The work can be extended by including a temporal library of XQuery functions to facilitate the writing of the more complex queries and hide some implementation details

REFERENCES

[1] AlGhamdi N., Rahayu W. and Pardede E., "Object-Based Methodology for XML Data Partitioning (OXDP)", in proc. of IEEE Intl Conf. on Advanced Information Networking and Applications, 2011, pp. 307-315.

[2] Ammari F.T. and Lu J., "Advanced XML Security", in proc. of Seventh IEEE Intl. Conf. on Information Technology, 2010, pp. 120-125.

[3] Birahnu L., Atnafu S. and Getahun F., "Native XML Document Fragmentation Model", in proc. of Sixth Intl. Conf. on Signal-Image Technology and Internet Based Systems, 2010, pp. 233-240.

[4] Cao Y., Majumdar S. and Lung C.H., "Caching Techniques for XML Message Filtering", in proc. of 28th IEEE Performance Computing and Communications Conference, 2009, pp. 315-322.

[5] Dai L., Lung C.H. and Majumdar S., "Bfilter - A XML Message filtering and Matching Approach in Publish/Subscribe Systems", in proc. of IEEE Globecom, 2010, pp. 1-6.

[6] Gao L., Xing B., Zhang J. and Li H., "Developing Efficient XML-SNMP Model: An XML-Template Based Approach", in proc. of IEEE Intl. Conf. on Computer Applications and System Modeling (ICCASM 2010), 2010, pp. 4-731-734.

[7] Haiwei Z. and Xiaojie Y., "Schemas Extraction for XML Documents by XML Element Sequence Patterns", in proc. of First Intl. conf. on Information Science and Engineering (ICISE), 2009, pp. 5096-5099.

[8] Harrusi S., Averbuch A. and Yehudai A., "XML Syntax Concious Compression", in proc. of IEEE Data Compression Conference (DCC), 2006, pp. 411-421.

[9] Jumma H., Rubel P. and Fayn J., "An XML-Based Framework for Automating Data Exchange in Healthcare", in proc. of 12th IEEE Intl. Conf. on e-Health Networking Applications and Services (Healthcom), 2010, pp. 264-269.

[10] Kyu Z.M. and Nyunt T.T.S., "Storing DTD-Independent XML Data in Relational Database", in proc. of IEEE Symposium on Industrial Electronics and Applications (ISIEA), 2009, pp. 197-202.

[11] Liu H.Z. and Murthy R., "A Decade of XML Data Management: An Industrial Experience Report from Oracle", in proc. of IEEE Intl. Conf. on Data Engineering, 2009, pp. 1351-1362.

[12] Liu J., Ma Z.M. and Yan Li, "FTwig: Efficient Algorithm for Processing Fuzzy XML Twig Pattern Matching", in proc. of Seventh IEEE Intl. Conf. on Fuzzy Systems and Knowledge Discovery (FSKD 2010), 2010, pp. 2270-2274.

[13] Lv T. and Yan P., "A Survey Study on XML Functional Dependencies", in proc. of First Intl. Symposium on Data, Privacy and E-Commerce", 2007, pp. 143-145.

[14] Maarouf M.Y. and Chung S.M., "XML Integrated Environment for Service-Oriented Data Management", in proc. of 20th IEEE Intl. Conf. on Tools with Artificial Intelligence, 2008, pp. 361-368.

[15] Mabanza, "Analyzing the Impact of XML Storage Models on the Performance of Native XML Databaswe Systems - A Case Study", in proc. of Seventh IEEE Intl. Conf. on Information Technology, 2010, pp. 210-215.

[16] Onashoga and Sodiya, "A Confidential Electronic Result Transfer Using a Hybrid XML Security Scheme", in proc. of Eight IEEE Intl. Conference on Information Technology: New Generations, 2011, pp. 397-402.

[17] Quadah G.Z., "Processing Independent and Inter-Linked Documents in XML Databases", in proc. of IEEE IRI, 2009, pp. 305-311.

[18] Shahriar M.S. and Anam S., "Quality Data for Data Mining and Data Mining for Quality Data: A Constraint Based Approach in XML", in proc. of Second Intl. Conf. on Future Generation Communication and Networking Symposia, 2008, pp. 46-49.

[19] Shahriar M.S. and Liu J., "On Defining Functional Dependency for XML", in proc. of IEEE Intl. conf. on Semantic Computing, 2009, pp. 595-600.

[20] Sun B., Yuan X., Kang H., Huang X. and Guan Y., "Incremental Validation of XML Document Based on Simplified XML Element Sequence Pattern", in proc. of Seventh IEEE Web Information Systems and Applications Conference, 2010, pp. 110-114.

[21] Vacharaskunee S. and Intakosum S., "An Approach to XML Tag Recommendation", in proc. of Eight IEEE Intl. Conference on Information Technology: New Generations, 2011, pp. 18-23.

[22] Wang W. and li J., "An XML Firewall on Embedded Network Processor", in proc. of Fourth Intl. conf. on Networking and Services, 2008, pp. 1-6.

[23] Zhang S., Chen S. and Liang Y., "CGT Code-Based XML Data Compression Method", Second IEEE Intl. Symposium on Electronic Commerce and Security, 2009, pp. 456-459.