XML AND ITS APPLICATION IN INFORMATION SYSTEM: AN OVERVIEW

Biswajit Das **Rajesh Das** **Subhendu Kar** **Swarnali Chatterjee**

Abstract

This paper mainly discusses about the information retrieval from the web resources using XML. Also, discusses that how to control the bibliographic data on Internet. XML allows the creating of user-defined tagset or standard structured tagset, apart form HTML system defined tagset for structured information on the Internet. It helps in a library to indexing on web services, import – export on web from database or database to web.

Keywords : Web Resources, XML Applications, Import, Export

1. INTRODUCTION

Internet is a global information system, consisting of billions of web pages from which we can get information. The major activity on the Internet, that is electronic publishing and information uploading, are being done in quiet an unorganized way. The lack of proper organizing huge quantities of information has not retrieve from the Internet. In any information system, we have followed certain tools and techniques to tell us which document contains what information. These tools and techniques can be classification, cataloguing, indexing etc. in the case of traditional information systems (i.e. libraries, Information Centers etc). As Internet is a distributed information system with disjoint information sources spread across the globe, we cannot bring the related documents together in the retrieval process. Hence the tools and techniques meant for pinpointing information on the Internet need to be more accurate.

XML is significant because it makes it much easier to share and search resources that are in different formats. Until fairly recently, this wasn't much of an issue for libraries. Historically, libraries have served as centralized repositories of information. They purchased books, journals, films, and other information resources on physical media, and patrons found what the library owned by consulting a catalog that listed holdings. Most catalogs are designed with the assumption that once a library records some descriptive information about each resource it has purchased, this information won't have to be radically altered. For physical resources, this works pretty well, since the authors, titles, subjects, and physical characteristics of a book don't change.

Once access to the Internet became widespread, it became clear that providing access to remote electronic resources could be very problematic. Catalogs are designed to provide access to physical resources that are under direct control of the library. However, people want to read journal articles, books, and useful Web pages stored in dynamically updated databases that are maintained and owned by other organizations that might be thousands of miles away. Online library catalogs are poorly suited for providing access to these works; so many libraries do not include these types of resources in the catalog. As a result, it is often very difficult for patrons to know what electronic resources they can get t[1]hrough their libraries.

This is where XML comes into the picture. It is impossible to search or display information unless it is structured in a meaningful way. In plain English, this means that information providers need to agree on standards for encoding electronic documents so they can be retrieved in a uniform way. Libraries have encoded bibliographic records in MARC for many years, and that has allowed them to easily share catalog records, which reduces costs while improving services. For a variety of reasons, it is not feasible to encode the new types of resources patrons want access to in MARC. However, when the information is stored in XML, it is possible to share and combine that data in ways that would not otherwise be possible.

2. INTERNET AND ITS' SEARCH ENGINES, SEARCH DIRECTORY AND META SEARCH ENGINES

There are many 'search engines'; 'meta search engines' and 'search directory' are available in the Internet. Example of a search engines is http://www.google.co.in, meta search engines is http://www.mumma.com and search directory is http://www.yahoo.com. All of theses claim to be the information pin-pointer on the Internet. But the approaches of the existing search tools (search engines, meta search engines and search directory) are ill planned and inefficient to say the least. This is obvious from the two following important observations:

i) The result of aparticular search is so high in recall and low in precision.

ii) The output of the search result has no standard of [2]any kind.

The search tools (search engines, meta search engines and search directory) on the Internet use robots to collect information about the documents to generate keyword indexes. Generally, these tools use different algorithms to assign keywords to the document. Some of the robots collect keywords from the title whereas some collect from the first few lines of the document. But the very pre-assumption of the search robots, that the titles are expressive and the beginning few lines of a document focuses the content, do not hold good for maximum documents. To give an example, if we search the Internet for term 'data', we should be sure to get results at least regarding 'database', 'data center', 'data warehouse', etc. Certainly the recall will be very high. The retrieval of irrelevant result can be reduced by subject gateway approach, but cannot be eliminated. The main cause of the problem in HTML (*Hyper Text Markup Language*), the standard for web publication is system-defined tag. This '*tag set'* is mainly for display of the content and HTML provides no tag to address the content precisely. XML (*eXtensible Markup Language*) designed by W3C (World Wide Web Consortium) promises a possible solution to this problem. The major advantage of XML over HTML is its extensibility i.e., provision of user defined tags and attributes to identify the structural elements of a document. XML also provides structural complexity to define document structure that can be nested at any level of complexity.

3. WHAT IS XML?

XML is a extensible markup language for documents containing structured information. Structured information contains both content (words, pictures, etc.) and some indication of what role that content plays (for example, content in a section heading has a different meaning from content in a footnote, which means something different than content in a figure caption or content in a database table, etc.). Almost all documents have some structure.

A markup language is a mechanism to identify structures in a document. The XML specification defines a standard way to add markup to documents.

4. XML IS JUST LIKE HTML?

No. In HTML, both the tag semantics and the tag set are fixed. An <h1> is always a first level heading and the tag <ati.product.code> is meaningless. The W3C, in conjunction with browser vendors and the WWW community, is constantly working to extend the definition of HTML to allow new tags to keep pace with changing technology and to bring variations in presentation (stylesheets) to the Web. However, these changes are always rigidly confined by what the browser vendors have implemented and by the fact that backward compatibility is paramount. And for people who want to disseminate information widely, features supported by only the latest releases of Netscape and Internet Explorer are not useful.

5. XML IS JUST LIKE SGML?

No. Well, yes, sort of. XML is defined as an application profile of SGML. SGML is the Standard Generalized Markup Language defined by ISO 8879. SGML has been the standard, vendor-independent way to maintain repositories of structured documentation for more than a decade, but it is not well suited to serving documents over the web (for a number of technical reasons beyond the scope of this article). Defining XML as an application profile of SGML means that any fully conformant SGML system will be able to read XML documents. However, using and understanding XML documents *does not* require a system that is capable of understanding the full generality of SGML. XML is, roughly speaking, a restricted form of SGML.

6. WHY XML?

In order to appreciate XML, it is important to understand why it was created. XML was created so that richly structured documents could be used over the web. The only viable alternatives, HTML and SGML, are not practical for this purpose.

HTML comes bound with a set of semantics and does not provide arbitrary structure.

SGML provides arbitrary structure, but is too difficult to implement just for a web browser. Full SGML systems solve large, complex problems that justify their expense. Viewing structured documents sent over the web rarely carries such justification.

This is not to say that XML can be expected to completely replace SGML. While XML is being designed to deliver structured content over the web, some of the very features it lacks to make this practical, make SGML a more satisfactory solution for the creation and long-time storage of complex documents. In many organizations, filtering SGML to XML will be the standard procedure for web delivery.

7. ORIGIN AND GOALS

XML was developed by an XML Working Group (originally known as the SGML Editorial Review Board) formed under the auspices of the World Wide Web Consortium (W3C) in 1996. It was chaired by Jon Bosak of Sun Microsystems with the active participation of an XML Special Interest Group (previously known as the SGML Working Group) also organized by the W3C. The membership of the XML Working Group is given in an appendix. Dan Connolly served as the Working Group's contact with the W3C.

The design goals for XML are:

1. XML shall be straightforwardly usable over the Internet.
2. XML shall support a wide variety of applications.
3. XML shall be compatible with SGML.
4. It shall be easy to write programs which process XML documents.
5. The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
6. XML documents should be human-legible and reasonably clear.
7. The XML design should be prepared quickly.
8. The design of XML shall be formal and concise.
9. XML documents shall be easy to create.
10. Terseness in XML markup is of minimal importance.

8. HOW XML WORKS ?

In XML, Document Type Definitions (DTDs) may accompany a document, essentially defining the rules of the document, such as which elements are present and the structural relationship between the elements. DTDs help to validate the data when the receiving application does not have a built-in description of the incoming data. With XML, however, DTDs are optional. Data sent along with a DTD is known as 'valid' XML. In this case, an XML parser could check incoming data against the rules defined in the DTD to make sure data was structured correctly. Data sent without a DTD is known as 'well-formed ' XML.

With both valid and well-formed XML, XML encoded data is self-describing. The open and flexible format used by XML allows it to be employed anywhere a need exists for the exchange and transfer of information. This makes it powerful.

For instance, XML can be used to describe information about HTML pages, or it can be used to describe data contained in business rules or objects in an electronic - commerce transaction, such as invoices, purchase orders and order forms. XML is separate from HTML, but XML could also be added inside HTML documents. By embedding XML data inside an HTML page, multiple views could be generated from the delivered data, using the semantic information contained in the XML. Moreover, XML can be used for such compelling applications as distributed printing, database searches, and others.

When most people think of the things that can be done with XML, they are actually thinking about a family of related technologies rather than a single markup language. On a related note, it is better to think of XML as a grammar than as a language. XML establishes rules for defining new formats. In regular markup languages such as the HyperText Markup Language (HTML), authors must use certain tags that make the text bold, create clickable links, draw tables, apply style sheets, etc. With XML, authors are free to make up their own tags and attributes if they don't feel that those that have been created by others meet their needs. (See Figure 1.)

```
<?xml version="1.0" encoding="UTF-8"?>
<MyPhotoArchive>
<photo title="Walking on the Beach at Sunset" filename="http://home.earthlink.net/~banerjek/images/
SunsetBeach.jpg">
  <dogs>
    <name>Keiko</name>
  </dogs>
  <people>
    <name>Banerjee, Kyle</name>
  </people>
  <place>
    <country-state>OR</country-state>
    <city>Manzanita</city>
  </place>
  <date>1999-08-13</date>
</photo>
<photo title="Charming snakes" filename="http://home.earthlink.net/~banerjek/images/india/
Snakes.jpg">
  <people>
    <name>Banerjee, Kyle</name>
    <name>Lincicum, Shirley</name>
  </people>
  <place>
```

```
    <country-state>India</country-state>
    <city>New Delhi</city>
  </place>
  <date>1998-08-26</date>
</photo>
</MyPhotoArchive>
```

Figure 1:XML document

It's important to recognize that XML only provides a structure for storing information. It does not say anything about how information is displayed, it does not create links that people can click on, and it doesn't bring information resources

together by itself. If I want a Web-based photo archive that allows people to find pictures of my dog or trip to India, I still have to write or acquire a program with these capabilities. If someone else wants to develop an online photo archive that integrates my pictures with those stored in other repositories, she or he will also have to develop the software that will accomplish this task.

To make XML do something useful, other technologies are necessary. The most significant of these is the Document Object Model (DOM). The DOM is a complex subject that is beyond the scope of a short article. In a nutshell, the DOM is an interface between programs and an XML document. The DOM doesn't care what language a program is written in—there's no particular reason why a program that uses the DOM couldn't be written in Java, C, BASIC, or any number of other languages. The DOM is very important because it allows programs to do useful things with XML such as finding, sorting, manipulating, and displaying information. For practical purposes, nonprogrammers do not need to worry about the technical details of DOM. Experienced programmers may find it relatively easy to use DOM to work with XML, but it's not a task for those without significant programming skills.

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<html>
<head>
<title>My Photo Example</title>
</head>
<body>
<table width="100%">
  <xsl:for-each select="MyPhotoArchive/photo[dogs/name='Keiko']">
    <tr>
      <td><b><xsl:value-of select="@title" /></b><br />
        <xsl:value-of select="place/city" />,
        <xsl:value-ofselect="place/country-state" />
      </td>
      <td>
        <img>
          <xsl:attribute name="src">
          <xsl:value-of select="@filename" /></xsl:attribute>
        </img>
      </td>
    </tr>
  </xsl:for-each>
</table>
```

```
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Figure 2: An XSL style sheet

Non-programmers can also find uses for XML, but their options are more limited. Writing eXtensible Stylesheet Language (XSL) style sheets is a relatively easy task for anyone who has worked with any kind of style sheet, such as Cascading Style Sheets (CSS). The important difference between XSL and other types of style sheets is that XSL can perform calculations. XSL style sheets can selectively display or modify any element. For example, the style sheet in Figure 2 can be applied to the XML example in Figure 1 to create a small HTML table that displays in bold text the titles of all photographs containing my dog, along with the pictures. The location of the picture is listed on the line below the title. (See Figure 2.)

9. XML AND LIBRARIES

For years, libraries have been quietly using XML to perform functions such as improving access to archival materials, simplifying interlibrary loan processing, and enhancing digital collections, but increased reliance on the Internet for delivering information resources has brought XML into the mainstream, where its impact is starting to be felt by libraries of all sizes. As early as 1993, the library at the University of California- Berkeley started developing a method for encoding archival materials in XML. The outcome of this project was the development of the Encoded Archival Description (EAD) standard, which is now maintained by the Library of Congress. Use of EAD has been increasing steadily over the years as a growing number of archival finding aids have been moved to the Web.

For the past several years, individual libraries have been improving their services and saving money by developing their own XML applications. Since 1998, Oregon State University has been using an application called InterLibrary Loan Automated Search And Print (ILL ASAP) to automatically search interlibrary loan requests and print request forms sorted by location and call number, complete with availability information, scannable Ariel addresses, shipping labels (if no Ariel address is present), and billing data customized to the borrowing library or consortium involved. This free application has been adopted by dozens of libraries around the country.

More-ambitious XML projects have also been successfully implemented. The Washington Research Library Consortium uses XML to provide access to subscription databases, digital collections, materials requested via interlibrary loan, and library catalogs that run on a combination of commercial, open source, and locally developed platforms. This system, known as ALADIN (Access to Library And Database Information Network) not only delivers content to seven academic research libraries, but also performs critical related tasks such as patron authentication using XML messages transmitted between applications over the Web.

In the spring of 2002, the Library of Congress announced an official specification for representing MARC data in an XML environment, MARC XML. Even though sharing data between catalogs is relatively easy because of widespread support for the MARC format, the ability to express MARC data in XML is useful for any library trying to develop tools or access mechanisms that combine MARC data (e.g., the online catalog) with non-MARC resources (e.g., a locally maintained database or special collection). Now that a standard has emerged for representing MARC, it is reasonable to assume that vendors and others will develop tools that take advantage of the huge amount of data already stored in MARC format. As a matter of fact, within a few weeks of the announcement of LC's specification, well-known tools used for manipulating MARC records such as JAMES (Java MARC Events) and MarcEdit already contained support for the new standard.

As more libraries use XML, they are finding more uses for it. The eScholarship initiative at the California Digital Library not only uses XML to store books in a standardized format, but it also uses XML technologies to allow users to define their own displays. The Open Archives Initiative (OAI), an effort supported by OCLC, has developed a protocol that makes it easy to send a query to a database over the Web and receive the results in XML. OAI effectively makes it possible to perform searches of multiple databases simultaneously without the need for proprietary hooks into local databases. While this functionality is very similar to what was promised with Z39.50, OAI is much easier to implement, so the hope is that it will be widely used for many kinds of databases.

10. CONCLUSION

Is is described about the possibilities of creating standard tagset for bibliographic data of Internet resources using XML. This will in turn lead to more effective and efficient functioning of Internet. For the same purpose a standard syntax, which can bring together the existing tag codes, has to be worked out. The search engines have to accept one such standard syntax and train themselves about the semantics of each element. This will help in getting far more precise search results. If the new syntax does not disturb the existing standards and their rules, two -way conversions i.e. database to the Web and Web to database, can be done easily.

Over the next few years, the impact of XML on libraries is certain to increase. More likely than not, it will not be obvious when XML is used to improve library services, much as it is not obvious what kind of hardware and software a library uses for its catalog. The simplicity and flexibility of XML make it possible to integrate services and resources in ways that would have been impossible just a few years ago. Vendors, libraries, and open source programmers are all interested in finding ways to search many kinds of resources with a single query, and XML represents a major step forward in making this goal a reality.

12. REFERENCES

1 http://www.w3.org/TR/REC-xml/

3. http://www.xml.com/

4. HERWIJINEN (Eric van). The impact of XML on library procedures & services. CERN: Geneva, 2000.

5. http://www.w3.org/TR/1998/WD-xlink.htm

About Authors

Biswajit Das is a Student of PGDDLM of Department of Library & Information Science, Jadavpur University, Kolkata , West Bengal.

Rajesh Das is a Student of PGDDLM of Department of Library and Information Science, Jadavpur University, Kolkata , West Bengal.

Subhendu Kar is a Student of PGDDLM of Department of Library & Information Science, Jadavpur University, Kolkata , West Bengal.

Swarnali Chetterji is a Student of PGDDLM of Department of Library and Information Science, Jadavpur University, Kolkata , West Bengal.