Journal of International Technology and Information Management

Volume 13 | Issue 3 Article 3

2004

XML: GATEWAY FOR DATA MOBILITY AND UNIVERSAL CONNECTIVITY

Emmanuel Uzoma Opara
Prairie View A&M University

XML: GATEWAY FOR DATA MOBILITY AND UNIVERSAL CONNECTIVITY

Emmanuel Uzoma Opara
Prairie View A&M University

ABSTRACT

Extensible Markup Language (XML) started as an effort to simplify the standard Generalized Markup Language (SGML), which was the International Organization for Standardization's (ISO) standard for defining data vocabularies. This study investigates XML technology as a substitute to SGML that would be Web friendly, and easy to learn and use while supporting a variety of applications for faster and more efficient data mobility on the Web. It describes several compelling reasons why XML is a useful technology for representing structured data mobility. XML is a markup language used for data structure in a textual form. A specific goal of XML is to keep most of the descriptive power of SGML, while removing as much of the complexity as possible. This article includes an overview of XML and its specifications and corresponding components, development of schemas for defining industry standards data definitions, potential problematic impact on information systems and recommended solutions for developers and practitioners.

INTRODUCTION

As the Web has gained more widespread use, the amount of information collected and stored electronically continues to increase, as does the need to share data among disparate applications and non–compatible systems (Travis 2000). The need to develop a system to handle the storage, retrieval, exchange and presentation of data on the Web becomes critical. XML goal is to extend the capabilities of Web technologies to include a standardized methodology of exchanging data without compromising the safety of existing technologies. XML is a meta-language used to define other markup languages. XML is designed to describe data using a Document Type Definition [DTD] or Schema. Markup Languages are designed to add structure and convey information about document and data. In "markup" languages, the main mechanism for supplying structural and semantic information is by decorating the document with "elements" comprising of a "start tag," and occasionally some content, and an "end tag,". Example is <tag> data</tag>.

XML based standards are making true distributed Internet applications possible today (Angeles et al. 2001). Web services and peer-to-peer technologies are changing the way global business develop and integrate software. XML is one of the promising enabling technologies of the next generation of internet application development (Oxley J and Zhoa, 2003). The concept of Web interfaces until recently was limited by the toolsets that Web surfers had available. These toolsets include collections of widgets that consist of windowing systems, buttons, scrollbars, drop-down lists, check buttons, etc. For starters, the Web reduces to nearly nothing many marginal costs of doing business, such as communication and customer service.

XML technology and related specification are on the verge of ubiquity in networked applications, across all architectural tiers and vendor platforms. Currently, this technology has achieved near-universal industry implementation in platform operating systems, application servers, database servers, commerce servers, groupware servers and other networked applications (Travis 2000).

LITERATURE REVIEW

Travis, Dodds, Hollander (2000) among others, noted that XML emerged from two predecessors. These are the Standard Generalized Markup Language (SGML) and the Hype Text Markup language (HTML). According to Dodds (2000), SGML was developed to be the standard for semantic tagging of documents such as cataloging and indexing. SGML is a tedious and complex technology that is supported by expensive software. On

the contrary, HTML as a presentation specific language represents a challenge to those organizations that create and maintain repositories of links to materials on the Web.

Hollander and Sperberg-McQueen (2003), in a W3C article celebrating XML's birthday, describe their effort to make XML a stripped down version of SGML so that it could be easily adoptable and extensible. They concluded that the effect of these are the results of the production of a family of XML technologies, which are now the flagship for emerging web technologies for business-to-business and peer-to-peer (P2P)) communications. According to Walsh and Sang [2002], XML is rapidly becoming the strategic instrument for defining corporate data across a number of application domains. Knight (2003), Kobielu (2003), Walsh and Sang (2002) among others defined XML as a set of syntax rule and guideline for defining text-based markup languages. These authors continued by noting that XML have many capabilities which include exchanging information, defining document type, specifying messages, and having a language that is not limited to the English language.

XML has enabled Web functionalities by its capability of having information that is expressed in a structural text-based format that is easily transmitted, transformed and interpreted by entities that understand the structure. Early misconceptions among some industrial leaders had XML replacing HTML, however others within the industry see them as complementary technologies (Holland 2001, Ravis and Dick 2000). XML is able to represent data found in common data sources, such as data warehousing, data mining, databases and various system applications. It is also represented in non-traditional data sources, such as spreadsheets and word processing documents (Foreschauer et. al, 2000). In the past, these non-traditional data sources were constrained by varieties of proprietary data formats, hardware and operating system platform compatibilities (Tibco 2002).

Although visual and user interface standards are a necessary layer, Walsh [1998], XMI/EDI Group [1998], and Hollander [1999] among others found evidence that the visual and user interface standards are insufficient for representing and managing data.

Earlier studied by Walsh, [1998], and Hollander [1999] found that HTML provide rich facilities for data display, but does not provide any standards-based format to manage data as data. More recent studies have paid considerable attention to the need for a standard for data representation that will expand the Internet capabilities in such manner that HTML standard displayed in the past. Such data standard will set the tone and act as the medium for Business-to-Business (B2B) transactions, publication of personal preference profile, automated collaboration, and database sharing. Chen [2000] noted that those electronic transactions, such as payments of bills, medical and pharmaceutical data, semi-conductor part and sheets, purchase orders, etc. will be written in this format. Agital [2001], Boumphrey and Tittel [2000], Dodds [2000], Dick [2000], among others, found evidence that the data format or standard is eXtensible Markup Language [XML].

PURPOSE OF THIS STUDY

The purpose of this paper is to examine the Extensible Markup Language (XML) as a simplified version of Standardize General Markup Language (SGML) that would be Web friendly, easy to learn while supporting a variety of applications and also to investigate the viability of an XML platform that can help business owners avoid being locked in with a particular Enterprise Resource Platform (ERP) vendor.

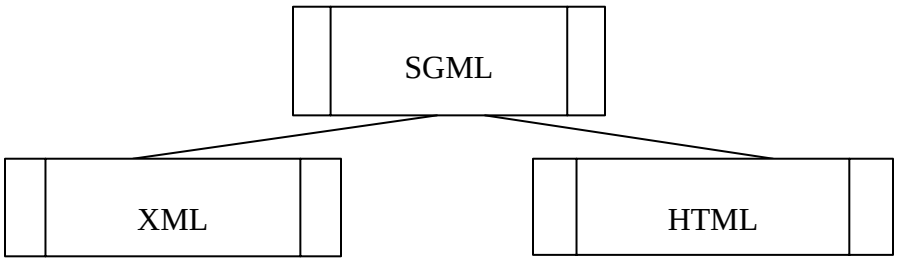
MARKUP LANGUAGES

Travis [2000], Sterling Commerce [2001], Tibco [2002] among others, defined a markup language as a special set of indicators, called tags that indicate how information should be interpreted. These are codes that specify how text, graphics and other elements are displayed on the Web and indicate where links lead. From a lay-man's point of view, it is similar to using a highlighter to mark an important subject matter in a text book. For example, the new office XP word processor uses XML to specify formatting and layout. In connectivity functions, programs use markups to express the meaning of data across networks. XML is different from HTML because it permits authors to create individualized markups for any type of information. Since an XML document contains only data, applications decide how to display the data. HTML is an "application" of SGML while XML is a simplified version of SGML. Enterprise integrated database systems and multimedia interactive processing

programs are examples of markup applications. Enterprise integrated database systems associate meanings and relationships with data, while multimedia interactive processing programs, express meta-data about images and sound (Jones et. al. 2000). More of the markup languages created with XML include MathML (for mathematics), VoiceXML™ (for speech), SMIL™ (the Synchronous Multimedia Interface Language – for multimedia presentations), CML (for chemistry), and XBRL (Extensible Business Reporting Language – for financial data exchange).

Due to the descriptive nature of XML tags, it is easy to manipulate, sort, search and generate XML documents with related technologies (Mirchandani and Motwani 2001). An example of these technologies is the extensible style sheet language (XSL). Below is the relationship between SGML (the father of markups), HTML (the son) and XML (the brother).

Figure 1. Standard Generalized Markup Family [SGML]



XML AS A METALANGUAGE

Since XML is a meta-language or a language about language, it allows programmers to describe (tags) data in such a way that the owner of the data could make the document very powerful and meaningful. XML share some syntactic similarities to HTML, because it makes use of tags (words bracketed by < and >) as well as attributes (name="value").

The disparity between XML and HTML is that HTML uses tags and attributes to describe how a text will appear in a browser. This is shown in the following example; <p> is a paragraph in HTML. In contrast, in XML, <p> could mean a paper, page, pen, or any word that starts with a "p," because XML uses tags and attributes to establish boundaries for data and leaves the interpretation of such data to the application that reads it. Such an application might be a web CT interface, a web based soft phone system, or an online banking system. XML is a widely-supported, open technology (i.e, non-proprietary technology) for data exchange (Zviran and Erlich 2003). That's why XML is an enterprise integrated web delivery system.

SAMPLE CONFIGURATION OF XML

XML is built on a tree or hierarchy structure because it was designed to provide a language that is user friendly and can be understood by humans although processed by machines. Here is an example of a simple XML document:

```
<?xml version="1.0"?>
    <manuscript>
        <isbn>0-03-033078-5</isbn>
        <title>XML for JITIM</title>
        <author>Opara, Emmanuel Uzoma</author>
        <price cost="105.00"/>
    </manuscript>
```

The example above indicates that <?xml version="1.0"?> tag shows the beginning of an XML document and specifies the version. The < manuscript >and</ manuscript > tags identifies the beginning and end of a section. As shown in this section, there are four tags namely : <isbn>, <title>, <author>, and <prize>. All the tags are

considered XML elements and have specific values given by the attributes of the element. They are also organized by the hierarchy of the document. As have been illustrated, XML is easy to read and comprehend once the user becomes accustomed to the different components.

The power of XML is not only in its ability to organize and format information, but in its ability to interact with the information. For illustration purposes, here is another example of an XML in action. "Mary Carter makes a request from an online library. The customer makes a request for a manuscript on XML from the *Journal of International Technology and Information Management* (JITIM) written by "Opara Emmanuel." The query would look like this: <query isbn= "0-93-033078-5" zipcode="77083"/>

The query is an XML element, while the data is represented as the value of the two attributes: isbn and zipcode. To process the request, the query is sent to an XML interpreter that reads the query and access the internal information to formulate a response that would look like this: <reply time="3 days" price="105.00"/>

Namespace http://www.onlinemanuscripts.com	Namespace http://www.music.com
<manuscript> <title>Deliverance</title> </manuscript>	<recordalbum> <title>Deliverance</title> </recordalbum>

XML also provides the capacity to distinguish elements and attributes from others that have similar names through the use of namespaces. There are many titles and descriptions of goods and services, which use the same elements and attributes. An illustration is shown the following example. A manuscript and a record album might have the same name. Again in Mary Carter's example, we will have: XML is a Meta language for structuring data (Dornan 2003). Unlike HTML, XML documents do not contain formatting information. When loaded into a browser such as Internet Explorer 5 (IE5), it is parsed by IE5's parser classified as **msxml** code and displayed. Also, when XML documents are parsed by the Document Object Model (DOM), the data are placed into a tree structure. Here is another illustration of a simplistic format of an XML markup language:

```
<?xml version="1.0"?>
<helloamerica>
<opara>Is anybody out there?</opara>
<nobody>No</nobody>
<singing>song</singing>
</helloamerica>
```

In the above illustration, the <?xml version="1.0"?> tag identifies the beginning of an XML document and specifies the version. The <helloamerica> and </helloamerica> identifies the beginning and end of a section. In that section, there are three tags: <opara>, <nobody>, and <singing>. In this example, the <opara> and <nobody> tags identify who is speaking the text, while the singing tag identifies the sound of a song.

The XML document illustrated above is a very basic example. Currently, there are several sophisticated applications in which XML is used to communicate different types of information on the Web. These include the capabilities of displaying text in a web browser, providing input for an interactive application, online banking, streaming videos, text to speech conversion, etc. XML will revolutionize the manner organizations structure its information enhancing communication infrastructure.

XML ACTION IN A CYBERNETIC SYSTEM

XML documents end with the .xml extension (Wang et. al. 2002). To process an XML document, a software program called XML parser or processor is needed. Parsers check XML document's syntax and also support the Document Object Model (DOM) or the Simple API for XML (SAX) or both. Further, DOM-based parsers build a tree-like structure that contains the XML document's data in memory. This process makes it possible for data to be programmatically manipulated. SAX-based parsers process the document and

systematically generate events, when tags, text, comments, etc., are encountered. A cybernetic cycle is complete when the events return data from the XML document.

XML documents can reference optional Document Type Definition (DTD) files. These files define how XML documents are structured. In a cybernetic system, when DTD are provided, some parsers, also called validating parsers, are able to read the DTD and check the XML document structure against it. If the XML document conforms to the DTD, then the XML document is valid. When parsers cannot check for document conformity and as a result ignored the DTD, they are called non-validating parsers. When an XML parser successfully processes an XML document that does not contain a DTD, the document is considered syntactically correct or well formed. See (Figure 2) as XML documents and their corresponding DTDs are parsed and sent to an application.

Figure 2. XML Document and Its Corresponding DTDs

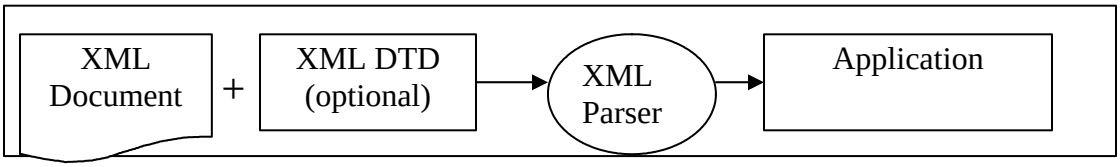
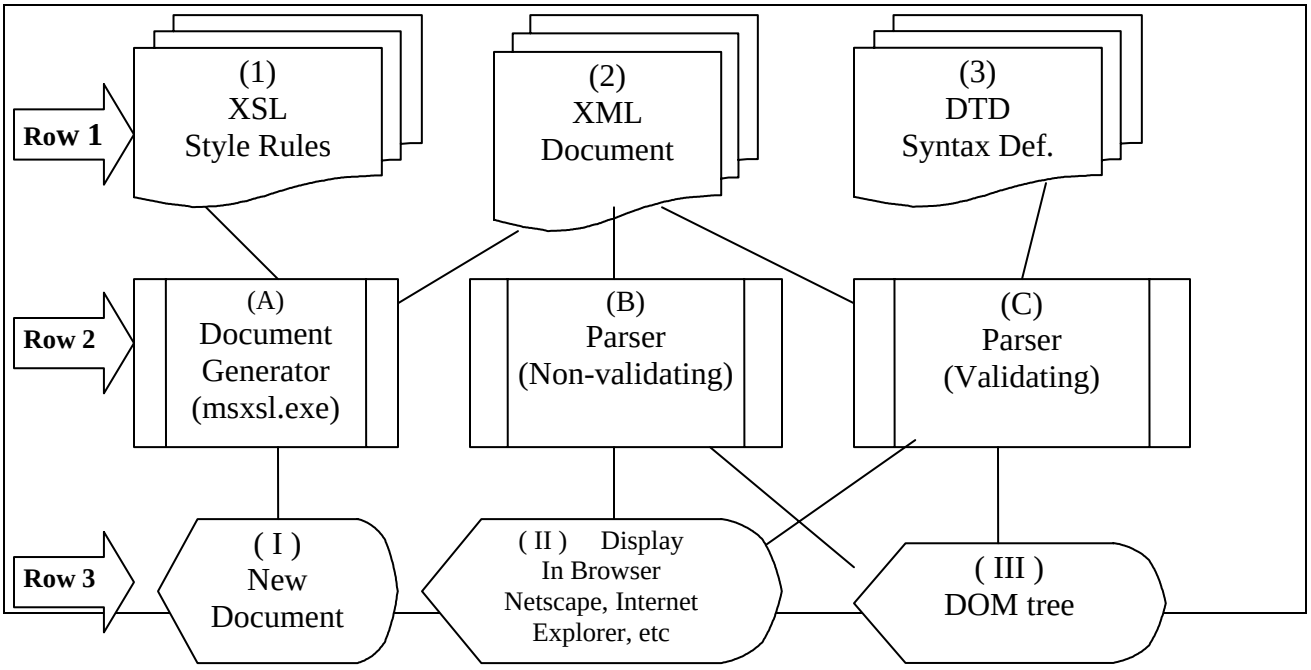


Figure 3 indicates some of the characteristics that may be present in an application that uses XML technology. The three categories represent features that could be present in applications that use XML. These features are the Multi Document Type's layer (Row 1 of figure 3), the Predefines Application Process layer (Row 2 of figure 3) and Output Phases layer (Row 3 of figure 3).

The Multi Document Type layer (row 1) represent documents formatted according to the XML and related standards. The pre-defined application process layer (Row 2) represents software that is provided by the application to process the documents. This software will contain an XML engine to process the XML document and other application related codes. The Output layer (Row 3) represents the result of processing.

Figure 3. Applications using XML Technologies



MULTI-DOCUMENT TYPES (ROW 1)

The Multi Document row shows the actual document types that are used in XML. These include an optional XSL style sheet, the XML document and an optional description of a valid syntax called document type definition (DTD). When implementing an XML technology, the documents may be located on several sites and are linked together using the XLink feature. An XLink format looks like a standard uniform resource locator or URL ("href://'). With the XLink features toggled on, it becomes easier for the XML document to provide links to the DTD, XSL or other XML embedded documents.

APPLICATIONS (ROW 2)

Row 2 contains the predefined application process cells "A," "B" and "C". This row contains some of the application types that may be used to process XML and its optional related documents. The connecting lines indicate the relationships between document types and the final results of an operation.

In cell A, the application type is a file conversion that inputs both an XML document and XSL Style Sheet and the outputs of a new document. The original document is usually written in a HTML code that is designated for use in a browser, although any sort of document could be created. Cells (B and C) are embedded programs that are used to convert XML document into a format that can be used by another program. The parser (non-validating) and validating cells are applications or entrenched code that makes use of a library parser to interpret the XML file into a tree like structure. The cycle is complete when the system performs the processing to transformation phase of data for the output. The parsers have the capability to check the structure of an XML document if a syntax description (a Document Type Definition) is provided. Developers usually build a parser into the Internet Explorer 5.0 and Netscape 7.1 browsers. The Microsoft Browser makes use of "data islands" that contain XML documents and the scripting languages that extend the Document Object Model (DOM) to reference the XML data as part of the overall document. Several programs' libraries are available for incorporation into the user written programs.

OUTPUTS (ROW 3)

Row 3, cell I, II and III contains some examples of the type of output that results from the applications of a cybernetic system involving XML. The programmer might choose to store the output in a file, send the result to a data warehouse or a database, display output in a browser, or keep the result in memory for future use. These examples reveal the power of XML technologies. Management or the end-users can decide on how to use the data stored in the XML document. A programmer can also use the same data for dissimilar or future purposes.

With the use of XML, the user is not limited to having the display of data controlled by the browser or required to send requests back to the server to redisplay information that the user is interested in. Instead, the creation of applications to use XML files provide potential competitive advantage for document sharing among businesses that would otherwise not be available from proprietary file format.

XML TECHNOLOGIES

XML platform enables global businesses to develop application-specific languages that better describe their business data. XML makes it possible for a Web site to be voice sensitive so that customers and business clients can make phone calls to retrieve important information over any type of phone line, regardless of it digital, analog, or IP capabilities. The platform provides the necessary structure, such as grammar in an English language format, to enable systems and people to communicate effectively and efficiently.

Since the main focus of XML is data, everything else such as formatting, search and display, etc., requires a member of the XML family. These XML technologies and their basic purposes are illustrated in table 1. These components mirror the modularity of the web.

Table 1. Components of XML Technologies

XML TECHNOLOGIES	
Cascading Style Sheets (CSS) **Extensible Hypertext Markup Language (XHTML)** **Extensible Style Sheet Language Formatting (XSL)** **VxMl (Voice XML)**	**This group are used for presentation purposes**
Document Type Definitions (DTD) **XML Schema**	**This group is used for data type description purposes**
XSL Transformation (XSLT) **XPath** **Xlink** **Xquery**	**This group are used for data type descriptions purposes**

The integration of computing architecture and increasing need for enterprise integration of data demand a structure of components that is easily assembled, can communicate with multiple programming languages, is interchangeable and re-usable at all times. As noted by Goldfarb and Prescod (2000), just as interchangeable parts controlled the industrial age, so will re-usable information drive the information age.

IMPLICATION FOR MANAGERS AND PRACTITIONERS

In the introduction phase, XML was described as a concept and toolset that will allow the creation of applications, which can be used by heterogeneous organizations, to analyze, transfer, and communicate information over the Web. The technology development cycle for XML is now in the "critical phase." The challenge for developers will be the impact of XML on organizational profit margin as businesses engulf the world of e-business.

Many companies will find that while XML enriches organizational capabilities, it also results in the need for major systems upgrades. As a text-based format, XML carries a lot of baggage. Most developers call it the "lingua franca" of standards. This standard can also be termed "lingua verbose". "XML is very verbose in nature because lot of bits can be put together to represent a very small amount of information. For developers to avoid these problems, there will be a need to increase the capacity on networks servers for memory, storage and CPU processing, because XML consumes large amount of horsepower.

For software developers that design Web applications, XML provide a powerful and flexible format for expressing data. This format could serve as conduit for sending data between clients and servers. It can also serve as a transfer format for sharing data between applications, or as a storage format for different storage devices. XML provide flexibility in the manner information is stored. Developers will no longer remain locked into a fixed schema of predefined relational database tables. Instead, they can extend their capabilities to manage both structured and unstructured information by adding new XML tags to their data sets, despite the dynamic nature of the e-commerce environment.

Since XML offers a mechanism for adding meta-content to HTML and the Web, end-user managers will be provided rich set of Web applications for browsing, communication, and collaboration (Boumphrey 2000).

IMPLICATION FOR RESEARCHERS

For researchers, the technology development cycle for XML is now the "excitement phase." What does the future hold? Addition research will be done to answer that question. However, the next few years will show the impact of XML on e-commerce businesses as new information retrieval and data exchange capabilities becomes critical to enterprise competitive strategies. Also, with the increased capabilities of businesses to store and process data, the role of the browser will change from that of presentation to being an application tool. This change will result from the combination of XML based data and HTML embedded scripting to customized presentation format which will be beneficial to the end-users. However, further research should confirm all the mentioned possibilities.

CHALLENGES

Although XML is now the world standard platform for e-business transactions, it is still not the solution for complete web transaction. XML is only a standard foundation on which answers can be built. XML faces the challenge of speed, scalability, and memory and storage capability as the technology interface with peripheries of computing environment.

According to Dyer (2003), most organizations find that while XML enriched their information capabilities, it also results in the need for major systems upgrades. Dornan (2003) stated that XML is the 'lingua verbose' of standards because as a text-based format, it has the capability of carrying a lot of excess baggage. The reasons could be that a whole lot of bits have to be configured to represent a very small amount of information. As a result, the organization will need to increase the capacity of its network memory, storage and CPU processing capabilities, because XML consumes a large amount of horsepower. Since XML is self-describing, programmers will need to do more work in the designing phase.

Walsh and Sang (2002), Dyer (2003) among others noted that the human experience is far beyond text, with audio, video and images becoming more and more a part of our computing environment. The effect of this is that organizational system consumption of bandwidth will be dramatically larger than any amount of markup inserted inside of text.

The impact of XML on the database environment will eventually mean more specialization for database specialist. According to Travis (2000), there is a demand for infrastructure products to be standardized. Some developers might not develop a product that is unique because of the difficulty of integration of such product into the IT platform.

Another challenge will come from the use of XML in applications and as a source of data for search engines. With the systems increased workload to store and process data, the role of browsers will change from one of presentation only format to that of being an application tool. This change will result from the combination of XML based data and HTML embedded scripting or Java to customized presentation schema and to the user's needs.

SOLUTIONS

Several recommendations have been made on resolving the potential challenges of XML technologies. These include processing XML on a separate server. The reason for this recommendation is that as the volume of XML traffic among various organizations increase, the tasks of routing, parsing, and acting on traffic will overtax the general-purpose processors thereby sub-optimizing the task and functionalities of the processor. Firms should develop a new generation of XML processing appliances and specific switches that will eventually offload complex XML processing thereby easing the clogging of the system.

Another recommendation is that IP network routers could handle the XML challenges by taking on more of the functions of the application data routers. By implementing the "content-aware Web routers" the processors will be able to filter, route, queue, and cache SOAP.XML content.

The design and construction of industrial web-based services could eliminate the clogging of data traffic. These will provide four pillars of support in the areas of messaging, transactions, security and identity. At the moment, these capabilities are being provided from two different entities. One is from Microsoft's.NET and the other is J2EE (a Java-centric specification). The Micorsoft's.Net is a Window-centric framework designed for extending Window-based networks into the extended space of the Web. The J2EE, is a java-centric specification that is being implemented by technological oriented organizations such as Oracle, Hewlett Packer, IBM. Sun Micro Computers, BEA, etc.

SUMMARY AND CONCLUSION

The study has presented an overview of XML technological contributions to the Web infrastructure. It further presented how XML technologies will enable businesses to attain a higher Plato in the future. XML is an industry-wide recognized language for developing representations of semi-structured data that could be shared in an enterprise Web integrated system. It is known as the "Lingue Franca" of standards.

XML technology is a standard that empowers the enterprise to reinvent new electronic partnerships that leverage the availability and ubiquity of the internet to exchange and share data.

Due to XML, desktop computers are now used as a bank loan officer's user interface to conduct banking business on the Internet. It is also now possible for two legacy computing systems to speak with each other without expensive software changes to their coding systems. Enterprise systems such as healthcare, educational, governmental and corporation etc., have been impacted by XML technologies because it is the glue that binds together the Internet, legacy computing systems, browsers and "inputs," just as the human voice.

XML will improve Web-browsing applications for viewing, filtering, and manipulating information on the Internet. XML, if properly implemented will lead to cost savings, interoperability and new opportunities for businesses. As collaboration on the Web spreads to more businesses, customer services will eventually migrate from phone lines and storefronts to Web sites.

The study concludes by revealing that technologies, such as XML will enhance business prospects for maintaining and sustaining an above average returns on enterprise investments thereby maintaining leadership in its market place.

REFERENCES

A. Fabio Arciniegas, "What is XLink?" .(retrieved November 25, 2001), 2001.

Agital. (2001, August). *XML e-business solution available for free 30-day test*. (retrieved November 25, 2001), http://www.agital.net/company/press.

Angeles, R., Corritore, C.L., Basu, S.C., and Nath, R. (2001). Success factors for domestic and international electronic data interchange (EDI) implementation for US firms, *International Journal of Information Management*, 21(5), 329-347.

Arbortext Epic E-Content Engine and Empolis Sigmalink integrate for XML content management and publishing solution. (2000). *Arbortext News and Events.* http://www.arbortext.com/News_and Events/recent_news_pages. (retrieved November 25, 2001), 2000.

Arbortext. (2001). XML for managers. Arbortext Think Tank. (retrieved September 10, 2001),http://www.arbortext.com/Think_Tank/XML_for_Managers/xml_for_managers.html.

Boumphrey, F., & Tittel, E. *XML for Dummies.* Foster City, CA: IDG Books Worldwide, 2000.

Boumphrey, F.; Direnzo, O.; and et. Al., *XML Applications.* USA: Wrox Press, 1998.

Boumphrey, F.; Greer, C.; and et. Al., *Beginning XHTML*. USA: Wrox Press, 2000.

Byous Jon, Web Services and J2EEvTechnology: One Platform," Java.sun.com, July 8, 2002

Ceponkus, A., and Hoodbhoy, F., *Applied XML: A Toolkit for Programmers.* United States: John Wiley & Sons, 1999.

Chang, D., and Harkey, D., *Client/Server Data Access with Java and XML*. United States: John Wiley & Sons, 1998.

Chen, A., "*Getting to XML: The new B2B lingua Franca will need to live with EDI. Here's how to mix the two.*" eWeek (retrieved 7 August 2000), http://www.zdnet.com/eweek/stories/general/0,11011,2610043,00.html (2000.

Coyle, Frank, P. "XML, Web Services and the Data Revolution." Boston. Addison-Wesley Information Technology Series, 2002

Dick, K., *XML A Manager's Guide.* Reading, MA: Addison-Wesley, 2000.

Didie, M. "*Getting into i-Mode*." (retrieved 20 September 2000), http://www.xml.com/pub/2000/09/20/wireless/imode. HTML 2000.

Dodds, L., "*Gentrifying the Web: Taking a look at XHTML, the XML-Deviant finds that although the W3C HTML Activity is moving forward, the rest of the web is still lagging behind.*" (retrieved 13 September, 2000), http://www.xml.com/pub/2000/09/13/xhtml/index.html 2000.

Dodds, L., "Schema Round-up." (retrieved 6 September, 2000), http://www.xml.com/pub/200/09/06/deviant.html .

Dornan, Andy. "XML: The End of Security Through Obscurity?" Network Magazine, April 2003 Issue.

Dyer, Russell. "The XML.com Interview: Liam Quin. " Oreilly XML.com, April 06, 2003

Foreschauer, B., Stary, C., Ellmer, M. Pilsl, T., Otner, W., & Totter, A. (2000, March). SCHOLION-scaleable technologies for tele-learning. *Proceedings of the association for computing machinery's symposium on applied computing, 1,* 137-140.

Goldfarb, C.F., & Prescod, P. *The XML Handbook.* Upper Saddle River, NJ: Prentice Hall, 2000.

Holander, Dave and Sperberg-McQueen, C.M. "Happy Birthday, XML. " W3C web site. www.w3.org

Holland, R., "*More and more, XML diving into the 'stream.'*" eWeek (27 July 2000) http://www.zdnet.com/eweek/stories/general/0,11011,2608541,00.html, 2000.

Holland, R., "*XML bridges e-apps*." eWeek (7 August 2000), 2000. http://www.zdnet.com/eweek/stories/general/0,11011,2585447,00.html

Hollander, J., "*Netscape Sees XML in Its E-Commerce Future*." E-Commerce Times (retrieval September 2000) http://www.ecommercetimes.com/news/articles/990309-2.shtml http://www.xml.com/pub/2000/09/xlink/index.html 18, 1999.

Jones, M.C., Ku, C., and Berry, R.L. (2000). "Electronic Data Interchange: A Cross-Cultural Comparison of Key Usage Aspects between U.S. and Taiwanese Firms" *Journal of Global Information Technology Management,* 3(4), October, 24-44.

Kobielu, James. "Web Services Work, But Will They Scale?" Business Communications Review, April 2003

Morgenthal, J. P. and La Forge, B. *Enterprise Application Integration with XML and Java.* Upper Saddle River, NJ: Prentice Hall, 2001.

Mirchandani, D.A. and Motwani, J (2001) "Understanding Small Business Electronic Commerce Adoption: An Empirical Analysis." *Journal of Computer Information Systems, Volume XXXI, Number 3, Spring 2001*

Oracle. *Using XML in Oracle database applications.* (retrieved November 25, 2001) http://technet.oracle.com/tech/xml/info/htdocs/otnwp., 2001.

Oxley, J.E. & Yeung, B. (2001). E-commerce readiness: Institutional environment and international competitiveness, "*Journal of International Business Studies*, 32(4), 705-723.

Roussinov, D. and J. Zhoa (2003), "Leon Automatic discovery of similarity relationships through Web mining." *Decision Support System, 35(1), 149-166.*

Sterling Commerce. *XML and e-Business.* (retrieved September 25, 2001) http://www.sterlingcommerce.com/solutions/products/ebi/wp/xmlebiz%5Fwp.html., 2001.

Sum Microsystems. *Sun Microsystems leads industry effort to simplify XML development.* (retrieved November 25, 2001), http://www.sun.com/smi/Press/sunflash/1999-12/sunflash.991207.2.html. 1998.

TIBCO. *TIBCO Extensibility. TIBCO Software, Inc.* (retrieved September 2, 2002), 2002.http://www.tibco.com/solutions/products/extensibility.

Travis, B., XML *and SOAP Programming for BizTalk Servers*, Redmond, Washington, Microsoft Press, 2000.

Walsh Kathy and Shin Sang, "Discover and Publish Web Services with JAXR." Java World, June 14, 2002

Walsh, N. "A Technical Introduction to XML." (retrieved 3 October 1998), http://www.xml.com/pub/98/10/guide0.html 1998.

Wang, H., Mylopoulos, J., and Liao, S. (2002). Intelligent Agents and Financial Risk Monitoring Systems. *Communications of the ACM*, March 2002, 83-88.

XML/EDI Group. *Guidelines for using XML for electronic data interchange.* (retrieved September 24, 2001)http://www.xmledi-group.org/xmledigroup/guide.htm., 1998.

Zviran, M. and Erlich, Z. (2003). Measuring IS User Satisfaction: Review and Implications. *Communications of the Association for Information Systems*, Vol. 12, 81-103.