X-portal: Web-portal integrated with native XML database 2

Syed Murtuza Baker Dept of CSE
Manzur Ashraf Faculty, Dept of CSE
Muhammad Suhel Hussain Dept of CSE
A.K.Mohammad Sharif-Ul-Alam Razib Dept of CSE
{galib36,manzur_bd,suhel2k,razib_004}@hotmail.com
East West University

Abstract:

Our main motive in this paper is to establish XML as a data-source integrated with an e-portal. We have developed a prototype for that purpose, named 'X-portal' . Our challenge is to develop a native XML database, in which XML data is stored directly, retaining its natural tree structure (DOM architecture). At the same time, we would like to obtain all the benefits of relational database management, such as declarative querying and set-at-a-time processing. To obtain efficient processing on large databases, XML is rapidly emerging as standard for exchanging business data on the World Wide Web. Using XML technology on-the-fly stage (i.e., in transitional stage) is facing harsh difficulties[8,12,13,19,20] at present moment . Omitting these constraints and being motivated by [3], we focus our whole database in XML format behind the proposed e-portal. We tacitly propose the DBMS functionalities embedded in the XML data-source.

Keywords: Database, XML, Web services, e-commerce.

1. Introduction:

XML stands for eXtensible Markup Language .XML is not limited to publishing. It has been used successfully with business and commercial documents[4]. XML is rapidly emerging as standard for exchanging business data on the World Wide Web. Its nested, self-describing structure provides simple yet flexible means for applications to exchange data [22]. 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:

- Built in support for internationalization due to the fact that it utilizes Unicode. Human readable format makes it easier for developers to locate and fix errors than with previous data storage formats.
- 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. Large numbers of off-the-shelf tools for processing XML documents already exist.

With the ever increasing number of different media and output formats, it undoubtedly has become a necessity to separate the content from the structure and format of source documents and to use universally applicable formats to interchange data between applications and storage entities. [22,4,5] In particular it is possible to completely automate the flow of information between companies using XML data-bindings [4]. XML provides information more than data. So we can easily render it in different technologies for e.g. WAP. That is why it is extensible.

In this paper we focus on using XML as native database in behind of an e-portal. XML is used as a native database so that XML is portable in any technology(for example WAP). We use XML as native database so that we could adopt the legacy database very quickly and very easily with the new up-event software methodologies[22].
Using server-sided script (PHP) we integrate out native database and perform DBMS major functionalities in XML environment. We name it as 'X-portal'.

2. Organization:

The organization of the paper is as follows: Section 3 indicates related work of using XML in different aspects of secondary implementations. Section 4 covers the idea of web-application preliminaries, section 5 suggests features of using XML as data-source, section 6 propose the reason to choose PHP as server-sided script in implementing DB-related parsers, and finally section 7 describes detailed features and working procedures of proposed *X-portal*.

3. Related Work:

Lots of work have been done to store Semi-Structured data (XML document) into persistent storage mainly in three ways – a) special purpose DBMS b) OODBMS and c) RDBMS. In regard to special purpose database system, Rufus [11], Lore [1,9] and Strudel [7] report the development of research prototype, while LOTUS notes was developed as a commercial product [15]. These are tailored to store & retrieve XML documents using special purpose indices and techniques of query optimization. In perspective of object oriented database management system, the rich capability of such DBMS permits the use of O_2 [2] or Object store for the storage of XML documents (e.g., the MOSNET project [14]).For RDBMS, Two techniques can be considered. First, Schema are extracted from XML documents based on Semi-Structured data [23,6,10]. Second, considering workload of a target application, efficient Schema can be found [8,12,13,18,19,20]. Besides using XML as entity utilized *on-the-fly* , different works have also been done to consider XML as native database and consistency native database manager like TIMBER [3]. In this paper we propose xml as native database in X-portal project to incorporate it in ecommerce application / web-portal.

4. Web application architecture:

The architecture for a website is rather straight forward. It contains three principal components: A web server, a network connection and one or more client browsers. The web server distributes pages of formatted information to clients that request

it. The request is made over a network connection and uses the HTTP protocol. In some situations the content of a page is not necessarily stored inside the file. It can be assembled at run time from information stored in database (or other information repository) and formatting instructions in a file. Alternatively, it can come from the output of a loadable module (CGI or ISAPI). The web server uses a page filter to interpret and execute the scripts in the page. Web sites employ this strategy are called dynamic sites.

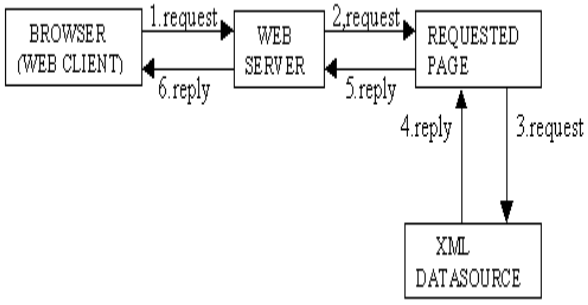


Figure 1 Dynamic Web Site Architecture

Dynamic web sites (shown in figure 1) offer certain advantages to website designers. They make it easy to keep the contents fresh and synchronized with data in a database. The overall look and feel of the web site is defined by a set of pages that contain code executed by the web server during a request for this page. In this context, the file can either be a plain text file with scripts interpreted by the web server or a compiled binary file that is executed by the web server. In either case the code in the "page" references and utilizes server resources which include databases, e-mail services, file services etc [24].

5. Using XML as native database in e-applications:

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 craft of HTML became increasingly necessary and XML was born.

The primary usage of XML is in data-centric model [6]. In a data-centric model, XML is used as storage or interchange format for data that is structured, appears in a regular order and is most likely to be machine processed instead of read by a human. In a data-centric model, the fact that the data is stored or transferred as XML is typically incidental since it could be stored or transferred in a number of other formats which may or may not be better suited for the task depending on the data and how it is used[6].

When we consider XML as self sufficient database that is kernel functionalities, Create, Read, Update, Delete (CRUD) activities, Data dictionaries (which will store data about the structure of data, relationships between data items, integrity constraints expressed on data, the name and authorization privileges associated with users) and interface functionalities (implementation of DDL, DML, DIL, DCL) have to be implemented and we have done that in proposed X-portal with embedded XML-parsers implemented by PHP.

6. Using PHP as server-sided scripting language:

We have made our online shopping store using PHP. It is an HTML-embedded scripting language. Much of its syntax is borrowed from C, java and perl with a couple of unique PHP-specific features thrown in. the goal of the language is to allow web developers to write dynamically generated pages quickly and effectively[21,5]

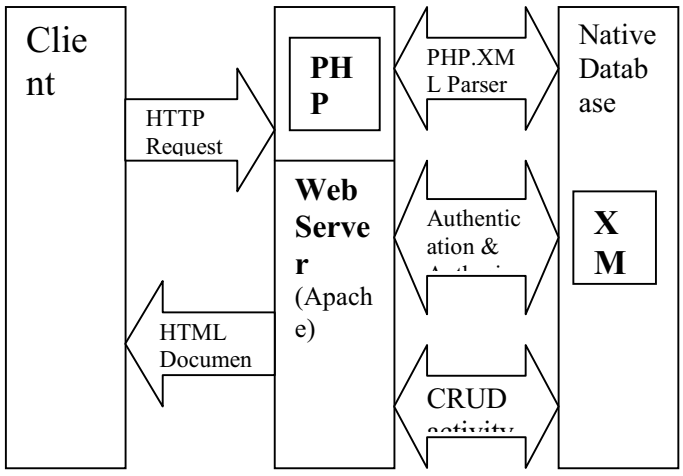
PHP was created with these particular needs in mind. Moreover, PHP code was developed for embedment within HTML. In doing so, it was hoped that benefits such as quicker response time, improved security, and transparency to the end user would be achieved, considering that almost a million and a half sites are currently running PHP (at the time of this article's publication), it would appear that these developers were right.[21]

6.1 Advantages of PHP over other server-sided scripting languages:

We are interested to give a comparison of PHP with other prevalent technologies to focus on the reason of our choosing PHP as embedded script of X-portal. The biggest drawback of ASP is that it's proprietary system that is natively used only on Microsoft Internet Information Server (IIS). This limits it's availability to Win32 based servers. ASP is said to be a slower and more cumbersome language than PHP, less stable as well [5] . Cold Fusion has better error handling, database abstraction and date parsing although database abstraction is addressed in PHP 4. PHP runs on almost every platform there is; Cold Fusion is only available on Win32, Solaris, Linux and HP/UX [5].

7. Working principle of proposed X-portal:

Our main motive is to develop a native database using XML which we will use in e-portal. Also we have implemented a security model in this project to validate 'authenticity' and 'protection of e-data'. We have taken an store as our example and we developed e-portal based on that example. A complete database has to follow the CRUD[8] (create, read, update, delete)scenario. Figure 3 shows the complete suite of X-portal.



Customer-use in X-portal is limited to viewing the stock and ordering items. Figure 4 shows the welcome page.

Figure 3: Complete X-portal suite

We provide different functions in our web page. The main features of our project are as follows:

7.1 Customer-use:

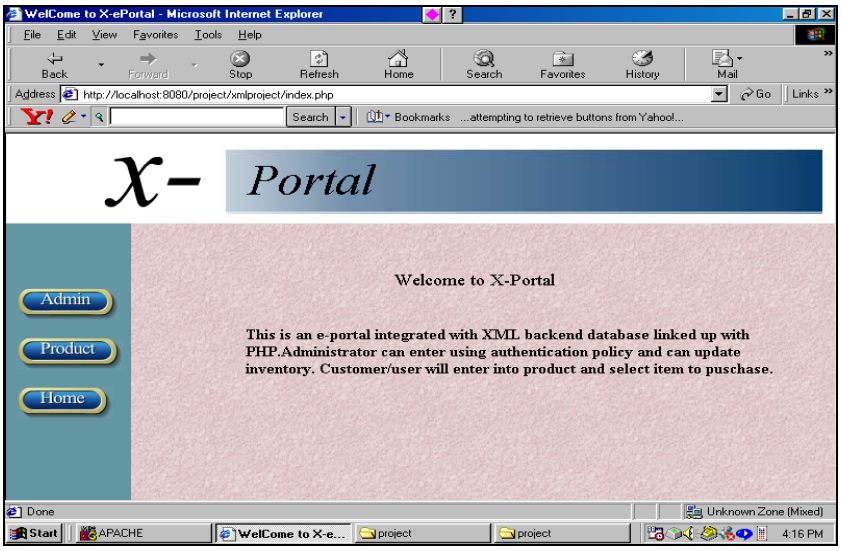


Figure 4: Index page of X-portal

7.1.1 Viewing the stock/inventory and ordering:

At the welcome page (*index.php*) customer is directed either to view the product information or to order a product. When the customer wants to order a product the portal redirect them to *customer_order.php* file. In this part we have parsed an XML file named '*product.xml*' (considered as a table in any relational database [10]) and viewed all the products that our store provides. We have used XML parser to do so. Upon ordering items by customer, portal works on two XML files at a time. It stores customer name, his address and date of purchase in a log file named *customer_log.xml*(consider a log table). We have done it, since we may need to undo some orderings. The other file on which we have worked on, is *product.xml*. When the customer orders a product the portal updates (which may be considered as update query in any relational database) *product.xml* file. In relational database we may consider this file as the product table .In Figure 5 and listing 1,we provide a snapshot of customer-page and *product.xml* file respectively.

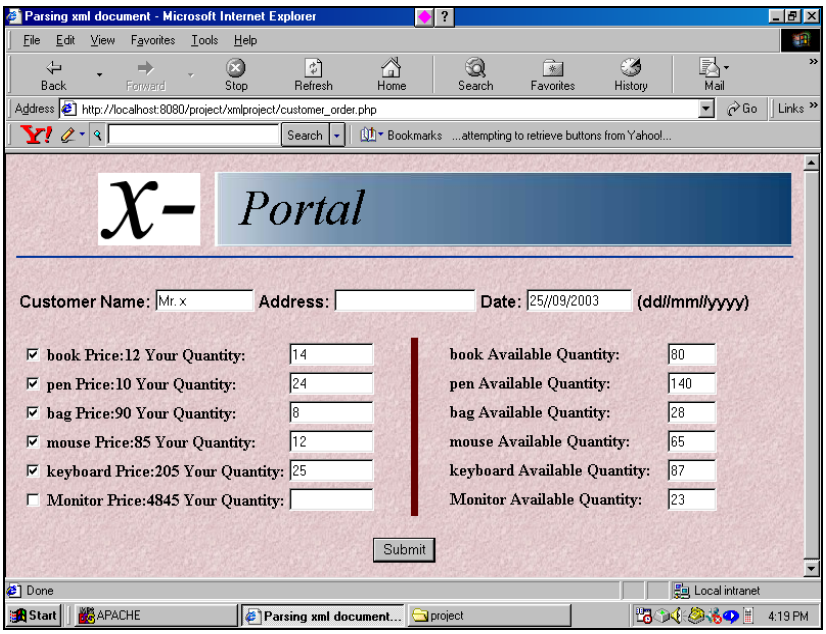


Figure 5: Customer- page of X-portal

</lotr_characters>

Listing 1: Product.xml content

```
<?xml version="1.0" encoding="UTF-8"
  standalone="yes" ?>
 <!DOCTYPE lotr_characters (View Source for full
  doctype...)>
- <lotr_characters>
  - <product>
      <item>book</item>
      <price>12</price>
      <quantity>26</quantity>
    </product>
  - <product>
      <item>pen</item>
      <price>8</price>
      <quantity>99</quantity>
    </product>
  - <product>
      <item>bag</item>
      <price>26</price>
      <quantity>7</quantity>
    </product>
  - <product>
      <item>mouse</item>
      <price>85</price>
      <quantity>18</quantity>
    </product>
  - <product>
      <item>keyboard</item>
      <price>210</price>
      <quantity>20</quantity>
    </product>
  - <product>
      <item>Monitor</item>
      <price>4850</price>
      <quantity>21</quantity>
  </product>
```

7.1.2 Supply-chain (B2B) :

When the customer orders a product the portal updates the quantity of his ordered product from *produt.xml* file on the *quantity* tag. So the file always contains updated data. When any of product quantity goes beyond threshold value (here zero(0))the portal does not show them on customer_order.php file. Again when any of my product goes to less than three(3) the portal immediately prints the product in *need_order.xml* file to facilitate the urgency of ordering the product(s) to the company. Company can arrange ordering of product from supplier (B2B).

7.2 Administrative use:

Now we will describe how the proposed e-store can take entry of a new product (which may be considered as insert query in any relational database) into XML data-source. We allow this option to be viewed only by the administrator. Security and authentication issues is implemented using appropriate scripts integrated with *password.xml* data-source. Administrator can enter new product or update any information of the product that is already in my product.xml file integrated with script *entry_product.php*. We provide session register so that this page cannot be accessed without this session registered value and the session is registered only when the admin enters the login name and password correctly. So there is no chance that the pages may be viewed from outside. In the *entry_product.php* file the page prompts for new item name, price and quantity. This file is added into the *product.xml* file (this may be considered as insert query in relational database). The *update_item.php* script provides updating issues of price or the quantity of any product. In figure 6, we provide snapshot of administrative page.

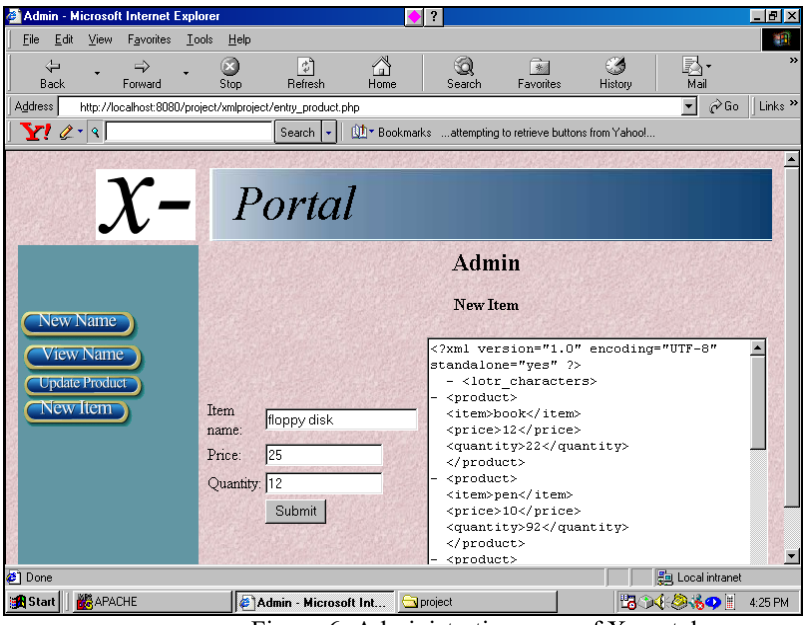


Figure 6: Administrative page of X-portal

7.3 Security issues:

X-portal have the support for a number of specialized access modes for browsing and authoring. It provides traditional user-on-demand mode for document release, a broadcast mode based on a combination of digital signature and encryption techniques. It supports 'push' or 'pull' [16] technology to provide user-specific {authorized} xml files.

8. Conclusion:

In this paper we focus native XML database behind of an e-portal. We have performed CRUD activities, security issues like authentication & authorization. It can be further extended with more features like 'transaction roll-backing', ' deadlock-preventive mechanism', etc upon the user requirements. Security policies must be restructured with proper encryption methodologies.

References:

[1] S. Abiteboul, D. Quass, J. MeHugh, J.Widom, J.Wiener. *The Lorel Query Language for Semi-structured Dat*a, International Journal on Digital Libraries, 1(1), pp. 68-88, April 1997

[2] F. Bancihon, G. Barbedette, V. Benzaken, C. Delobel, S. Gamerman, C. Lecluse, P. Pfeffer, P.Richard, F. Velez. *The design and implementation of O2, an object-oriented database syste*m,

[3] Jagadish H., Al-Khalifa S., Chapman A., Lakshmanan L., Nierman A.,Paparizos A., Patel M.,Srivastava D,Wiwatwattana N.,Wu Y.,Yu C.,TIMBER: A native XML database
December 2002, The VLDB Journal — The International Journal on Very Large Data Bases, Volume 11 Issue 4

[4] P. Buneman, S. Davidson, M. Fernandez, and D. Suciu, Adding Structure to Unstructured Data, Proc. of The International Conference on Database Theory (ICDT), Delphi, Greece, 1997.

[5] Official php site ;www.php.net

[6] A. Deutsch, M. F. Fernandez, D. Suciu, *Storing Semi-structured Data with STORE*D, SIGMOD Conference 1999: 431-442

[7] M. Fernandez, D. F. Florescu, J. Kang, A. Levy and D. Suciu, Catching the Boat with Strudel: Experiences with a Web-Site Management System, Proc. of ACM SIGMOD Conference on Management of Data, WA, 1998

[8] D. Florescu, D. Kossman, *A Performance Evaluation of Alternative Mapping Schemes for Storing XML Data in a Relational Database,* Rapport de Recherche No. 3680 INRIA, Rocquencourt, France, May 1999

[9] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, J. Widom, *Lore: A Database Management System for Semi-structured Data,* SIGMOD Record 26(3): 54-66 (1997)

[10]S. Netorov, S. Abiteboul, and R. Motwani, Extracting Schema for Semistructured Data, Proc. of ACM SIGMOD Conference on Management of Data, Seattle, WA, 1998

[11] K. Shoens, A. Luniewski, P. Schwarz, J. Stamos, and J. Thomas, *The Rufus system: Information organization for semi-structured data,* Proc. of the Int. Conf. On VLDB, pages 97-107, Dublin, Ireland, 1993

[12] J. Shanmugasundaram, K. Tufte, C. Zhang, G. He, D. J. DeWitt, J. F. Naughton, *Relational Databases for Querying XML Documents: Limitations and Opportunitie*s, VLDB 1999: 302—214

[13] F. Tian, D. DeWitt, J. Chen, and C. Zhang, The Design and Performance Evaluation of Various XML Storage Strategies. *Submitted for publication ,Computer Science, University of Wisconsin, Madison*

[14] R.V. Zwol, P. Apers, and A. Wilschut, Modeling and Querying Semi-structured Data With MOA, Workshop on Query Processing for Semi-structured Data and Non-standard Data Formats, 1999.

[15] http://www.lotusnotes.com/, 1998.

[16] Bertino E., Castano S., Ferrari E., Mesiti M., Protection and administration of XML
data sources; Data and Knowledge Engineering 43(2002) 237-260, Elsevier,2002

[17] M. Yoshikawa, T. Amagasa, T. Shimura and S. Uemura, XRel: A Path-Based Approach to Storage and Retrieval of XML

Documents using Relational Databases, ACM Transactions on Internet Technology, Vol. 1, No. 1, June 2001.
[18] A. Schmidt, M. Kersten, M. Windhouwer, F. Waas, Efficient Relational Storage and Retrieval of XML Documents, Proceedings of WEBDB 2000.
[19]M. F. Fernandez and J Simeon and P. Wadler, An Algebra for {XML} Query",Foundations of Software Technology and Theoretical Computer Science", pp.11-45, 2000.
[20]M. J. Carey and D. Florescu and Z. G. Ives and Y. Lu and J. Shanmugasundaram, E. J. Shekita and S. N. Subramanian",XPERANTO: Publishing Object-Relational Data as (XML)",WebDB (2000), pp. 105-110.
[21]www.zend.com/zend/art/intro.php#heading5
[22] Jayavel S., Eugene S., Rimon B., Michael C., Bruce L., Hamid P., Berthold R.Efficiently publishing relational data as XML documents IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120, USA.
[23] “XML by example” by Benoit Marhcal (1st edition).
[24] J2SDK 1.4.0 on windows 2000 server according to the documentaion of http://java.sun.com/j2se/1.4/install-windows.htm.