RESEARCH ON ELABORATION AND TESTING OF AN INTEGRATED SYSTEM BASED ON XML DATA ANALYSIS

Loredana MOCEAN
PhD, Business Information Systems Department,
Babes - Bolyai University, Cluj Napoca, Romania

E-mail: loredana.mocean@econ.ubcluj.ro, loredana_mocean@yahoo.com

Robert–Andrei BUCHMANN
PhD, Business Information Systems Department,
Babes - Bolyai University, Cluj Napoca, Romania

E-mail: robert.buchmann@econ.ubbcluj.ro, robimailunu@yahoo.com

Abstract: This paper approaches the importance of XML for better organizing and managing plain text data. This document provides the architecture and testing plan for a data model describing organization metadata as a collection of networked information. As an important result, we propose a new model of an integrated system based on XML, using the data analysis. It also provides some steps we must follow for this data model - using XML, the Extensible Markup Language.

Key words: XML; Integrated System; Database; Testing Models

Introduction

XML is a standard language for data description, widely used for sharing business information without taking in discussion the incompatible programs, computer networks, data structures or operating systems. XML provides structural and syntactical data interoperability. Because all the exchanged files contain the same XML mark-up, indexing, searching, combining, and re-using text-based information, they are easily accomplished. XML is text-based, thus allowing the information interchange between systems that are not natively compatible. Many companies have implemented XML as a standard for transactions involving extended systems and have invested in networks and Web services for offering the necessary support. However, employees are rarely using XML labelling office systems for the

structure and content of their documents. The opportunities to capture and further re-use such data are missed, or the companies have difficulties in migrating data to an enterprise level system.

Purpose of the study

This study gathers information in order to determine answers to important questions about ERP systems and XML databases, because currently there is no effective approach to test methods of an ERP system's prototype based on XML. The purpose of this study is to support businesses and software companies which desire to implement or upgrade their ERP base system, enhance competitive capacity and actively sustain a global strategy, especially regarding the current trend of communication and collaborative data discovery.

Materials and Methods

After a few years of work and studies in e-Commerce, Database Design, ERP Systems and Software Engineering, we arrived to an important stage of research. The prototype of an ERP system based on XML was proposed in previous researches and we reached the point of prototype testing.

- First of all, we have studied the theoretical concepts on Software Engineering and Programming Engineering;
- During the study of the ERP concepts we collaborated with two important software companies from Cluj-Napoca;
- The documentation base is being provided by multiple sources, one of them being the free articles and academic magazines offered by the Central University Library "Lucian Blaga"; at the same time, the Central University Library in collaboration with "Babes-Bolyai" University in Cluj-Napoca offers free access to on-line databases like PROQUEST, SPRINGERLINK, JSTOR.

Results

We define a new model of an integrated system based on XML and data analysis. The advantages of this model are the following [4][1]:

- The work-process is adaptable, it can be modified and adjusted to the projects' and customers' requirements;
- The capacity to process a big amount of data in a short period of time;
- The efficient communication with the customer and the quick reaction to his requirements, obtaining them with a superior flexibility;
- The externalization of data processing services;
- We need to design different levels of gave through which the processes can assure these levels.

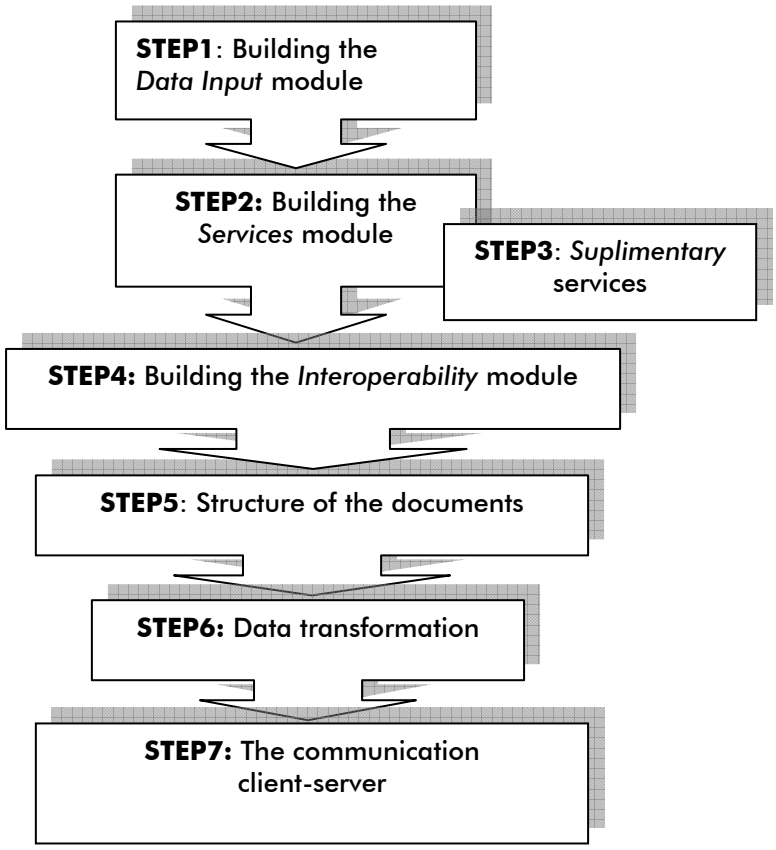

Figure 1. The steps followed in the elaboration of the model

The system's testing activities are [2]:

1. *The Modular testing* - is testing a module to see if it achieves the associated function of the module. The test is executed through some dedicated scripts (test drivers) which must have predictable behaviour, or manually. All discrepancies between the estimated results and the obtained results are collected in a test summary and are being tracked as errors;
2. *The Integration testing* - is a test phase involving two or more modules linked by the internal logic of the application. It assumes that the modular testing has already been executed and the reported errors were debugged. A great emphasis falls on testing the module interface;
3. *The Application testing* regards the application as a whole. It assumes that previous testing phases are completed and it's based on a test stub that automatically runs the application, feeds input and records output. Automated stubs are implemented with the AutoIt language and include random testing that simulates random user behaviour;
4. *The Integration application testing* - its goal is to check the application's functioning along its modules and through compatibility involving use cases;
5. *The Acceptance testing* - the goal is to validate the system in relation to the user's requirements.

In the following paragraphs, we will describe the steps in building the model and the proposed testing activities, for each building step.

Step 1: Data Input

When we design the module of *Data Input*, we have to include the following build activities:

a. Specialized teams on different formats of data and specialized teams in different programming environnements must build the data input. The data must be delivered in the following formats: XML/HTML (*.xml, *.htm), Comma Separated Values (*.csv), Word (*.doc, *.rtf, *.txt), Excel (*.xls), Portable Document File (*.pdf), Access (*.mdb).[12]

b. The data input specifications will be related to the online or offline Help, data indexes, e-books, databases, market studies, etc; of these services may benefit commercial banks, financial institutions, insurance companies, educational instititutions, universities and other institutions which work with a big volume of data.

c. We will implement some data validation procedures, including validation in relation to the XML Scheme, as to assure data quality and data structures integrity.

d. The data input will also be provided by alternative data capture methods, paper documents, microfilms or scanned images via OCR software able to convert scanned data in plain text markable data.

As testing activities, we propose the activities shown in table 1.

Table 1. The testing activities implemented in the first step: *Data Input*

Testing activities/Building activities	a	b	c	d
Modular testing	√	√		
Integration testing		√	√	√
Application testing				
Integration of application testing				
Acceptance testing			√	

Step 2: Services for the model

Our model must contain several modules which offer the following:

a. Verifying and adjusting the data.
b. Data analysis.
c. Data processing in structured or unstructured format as XML.
d. Creation and modification of XML Scheme.
e. Description of converting specifications.
f. Processsing of data input through OCR.
g. Structure and conversion of data formats.
h. By an efficient use of the conversion tools based on the DOM technology, our model must offer data conversion in XML, SGML, HTML and another structured formats, departing from different data formats: Word, Notepad, Textline, QuarkXPress, PageMaker, Mediaview etc.
i. Automatic data conversion.

As testing activities, we propose the activities shown in table 2.

Table 2. The testing activities implemented in the second step: *Services*

Testing activities/Building activities	a	b	c	d	e	f	g	h	i
Modular testing	√	√						√	√
Integration testing		√	√	√	√		√		√
Application testing					√	√		√	
Integration of application testing						√	√		
Acceptance testing			√		√	√		√	√

Step 3: Suplimentary services

We want to implement other modalities of data processing such as:

a. Images processing.

b. Creation and processing of the forms.

c. Design and support of databases.

d. The assurance of Total Quality Management.

As testing activities, we propose the activities shown in table 3.

Table 3. The testing activities implemented in the third step

Testing activities/Building activities	a	b	c	d
Modular testing	√	√	√	√
Integration testing		√	√	√
Application testing		√		
Integration of application testing			√	√
Acceptance testing	√			√

Step 4: The Interoperability enlargement between different platforms

We want to implement modalities of communication between different platforms. We follow these steps:

a. The analysis and possibility of communication within a heterogenous infrastructure: Windows platform, Linux platform etc.

b. The analysis and possibility of communication between different clients/servers databases: SQL server database, Oracle server database, SQL client database, Oracle server database etc.

c. We must assure communication of all clients with different servers based on different protocols of communication such as TCP/IP, SOAP etc.

d. We must build the XML common standards of communication.

e. We must ensure the transformation of the results of queries in XML format; XML offers a high abstraction level of the platform.

f. The implementation on each computer needs a XML client parser.

Table 4. The testing activities implemented in the fourth step: *Interoperability*

Testing activities/Building activities	a	b	C	d	e	f
Modular testing	√	√				
Integration testing		√	√	√	√	
Application testing						√
Integration of application testing						√
Acceptance testing			√			√

Step 5: Structure of XML documents

In [3] we presented a methodology for mapping relational data structure to XML vocabularies, which can be grouped in three phases:

a. Creation of the XML Scheme represented by the XSD document.

b. Definition of structures with a clear semantic specification.

With the help of XML Schema or DTDs we can define the contractual basis for data interoperability within the database system.

Table 5. The testing activities implemented in the fifth step: *Structure*

Testing activities/Building activities	a	b	c
Modular testing	√	√	
Integration testing		√	√
Application testing		√	
Integration of application testing		√	
Acceptance testing			√

Step 6: Data transformation

The XML offers some important services of data transformations (Data Transformation Services [13]).

We must go through the next steps:

a. Identify the data types, which sometimes do not have a clear equivalent in the translation step.

b. The enforcement by XML vocabularies of a set of common data.

c. The adoption of a standard where a common vocabulary convention is difficult to establish.

d. Use of advanced standards for changing the XML documents (XSLT).

Table 6. The testing activities implemented in the sixth step: *Data Transformation*

Testing activities/Building activities	a	b	c	d
Modular testing	√	√		
Integration testing		√	√	√
Application testing	√	√		√
Integration of application testing			√	
Acceptance testing			√	

Step 7: The client-server communication

a. We choose from a wide variety of Java-based XML and HTTP tools.

b. We use a prepacked set of XML-RPC tools.

c. With the help of XML-RPC tools we debug and establish connection between systems located in different environments.

d. We make the installation of XML-RPC servers.

e. We make the installation of ADO or ADO.net client/server communication technology.

Table 7. The testing activities implemented in the seventh step: *Communication Client-Server*

Testing activities/Building activities	a	b	c	d	e
Modular testing	√	√			
Integration testing		√	√	√	√
Application testing		√		√	
Integration of application testing					
Acceptance testing			√		

Some important things we must rely on, when we build and test the system

We must remember that XML, although a very strong technology, can lead to certain risks, if it not used properly, as follows:

- XML can bring a penalization for the performance, partially because of

generation/parsing/validation, especially in the case of an abusive DOM parsing (SAX being the alternative use of a smaller memory window but with disadvantages regarding the element reference);

- XSLT transformations also affect performance.
- XML is not fit for the representation of amorphous systems of unstructured data because it contains only a limited set of characters.
- We can transport XML binary data, but only after a previous conversion in a text file format; this means an important restriction, especially for a type of applications like the wireless applications.
- XML data fills a greater space than in binary format, therefore we must think about the efficiency transportation of the documents XML through the network, especially because of the possible emergence of strangleholds.
- XML format can be canonicalized and compressed with the classic algorithms for superior performance.

The test cases used in our testing phases are following the recommendations of standards such as IEEE 829 regarding testing documentation. Each test case is documented on three levels [2]:

- The test case specification, a general description and an identification of each test case
- The test case design, a more detailed description, including instruments, input partitioning and the expected result.
- The test case procedure, of an algorithmic nature, with an m-n relationship to test cases (the same test case may use more than one testing procedure).

Conclusions

Our system has a performant database for the post-relational era, it is based on a new generation of technologies, combining a multidimensional data server with a versatile applications server. We use an advanced object technique, fast development for Internet, an advanced programming language, a unique data stock technology, etc. Our system must support all traditional methods for Web pages development, a unique technology named *Cache Server Pages* (CSP), optimized for improved access to the database system.

References

1. Birbek, M. **Professional XML- Programmer to Programmer,** Wrox, 2004
2. Bradley, N. **The XML Schema Companion,** Addison Wesley Professional, 2003
3. Buchmann, R.-A. **Modeling relational data with XML,** Infobusiness 2006 – in „Proceedings of the International Conference on Business Information Systems", Iassy, 2006
4. Buchmann, R.-A. **Testarea Produselor Software, suport de curs,** 2008
5. Buraga, S. **Semantic Web * fundamente si aplicatii,** Matrix Rom, Bucharest, 2004
6. Laudon, K. and Traver, C. G. **E-Commerce: business. technology. Society, Updated Edition,** Addison Wesley Professional, 2002
7. Mindruta, C. **Arhitecturi, tehnologii si programare WEB,** Matrix Rom, Bucharest, 2005
8. http://www.afaceri-online.net
9. http://www.devx.com/Java/Article/16407/1763
10. http://www.forensicinnovations.com/formats-mime.html

11. http://www.informit.com/articles/article.aspx?p=25446
12. http://www.pcworld.ro
13. http://www.softpageinternet.ro/manual_html

[1] Codification of references:

[1]	Buraga, S. **Semantic Web * fundamente si aplicatii,** Matrix Rom, Bucharest, 2004
[2]	Buchmann, R.-A. **Testarea Produselor Software, suport de curs,** 2008
[3]	Buchmann, R.-A. **Modeling relational data with XML,** Infobusiness 2006 – in „Proceedings of the International Conference on Business Information Systems", Iassy, 2006
[4]	Mindruta, C. **Arhitecturi, tehnologii si programare WEB,** Matrix Rom, Bucharest, 2005
[5]	Bradley, N. **The XML Schema Companion,** Addison Wesley Professional, 2003
[6]	Laudon, K. and Traver, C. G. **E-Commerce: business. technology. Society, Updated Edition,** Addison Wesley Professional, 2002
[7]	Birbek, M. **Professional XML- Programmer to Programmer,** Wrox, 2004
[8]	http://www.pcworld.ro
[9]	http://www.devx.com/Java/Article/16407/1763
[10]	http://www.softpageinternet.ro/manual_html
[11]	http://www.afaceri-online.net
[12]	http://www.forensicinnovations.com/formats-mime.html
[13]	http://www.informit.com/articles/article.aspx?p=25446