Annales UMCS Informatica AI 2 (2004) 357-367

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

XML in eBusiness tasks realization

Dariusz Bober*

Department of Computer Science, Lublin University of Technology,
Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

There is no doubt that XML is rapidly becoming one of the most important data formats. An example of such data are the electronic business documents round via Internet between the partners of B2B business model. In this article the author will describe his experience from the initiation of electronic data interchange with the XML technology, in one of Polish companies.

1. Aspects of electronic data interchange

The initiation (put into practice) of electronic data interchange (EDI) is a very important process in companies nowadays, which is connected with a measurable profit [1]. The EDI (Electronic Data Interchange) standard is well known and highly valued, but also a very expensive solution. A company which is looking for economies could not be interested in the investment of such money as the EDI needs. An alternative could be the XML technology.

XML (eXtensible Markup Language) is relatively new, but very fast and still developing technology. The first time XML was presented at the SGML 96 Conference in Boston (November 1996). The 1st XML Conference was held by the Graphic Communications Association (San Diego, March 97). In October 97 the W3C came with a note on "W3C Data Formats on XML, SGML, HTML, and RDF" [2]. Until now the XML specification is backed by SoftQuad, Adobe, IBM, HP, Microsoft, Netscape, Lockheed Martin, NCSA, Novell, Sun, Boston University, Oxford University, and the Universities of Illinois and Waterloo [3].
The XML is a meta-markup language that provides a format for describing structured data. XML has much in common with the data description languages, such as DDS and SQL, but there is a major conceptual difference. An XML document is completely self-contained and includes both data and data description. Other languages only

* *E-mail address*: borowik@pluton.pol.lublin.pl

describe data (field names, data types, and sometimes-valid field values). The data are sorted in the relational database independent of the data description. With XML, a programmer should be able to extract the meaning of the data from a document without additional information.

On the surface, XML looks like HTML (Hyper Text Markup Language). Both are derived from the Standard Generalised Markup Language (SGML)[4]. The tools that generate HTML can often be reused to generate XML [5-7]. Both HTML and XML use marks <, > and & to create element and attribute structures. The main difference is that XML has not got a limitation of pre-set library of tags. When the document is marked up in XML, it is possible to choose the tag name that is the best way to describe the contents of the element. This facilitates more precise declarations of content and more meaningful search results across multiple platforms. It is flexible enough to handle an incredibly wide variety of information, and also allows for such information to be self-describing, so that it may be manipulated by software that has not been previously exposed to description of the underlying meaning behind the data. With its powerful expressiveness and flexibility, the XML promises to give the EDI's functionality without excessive costs increase.

Looking at so many advantages of the XML it is possible to think "why is somebody still paying for the EDI systems?" The answer is simple: "because of its completeness". The EDI standard has been developed since the 60th and it offers all, from hardware (highly efficient servers and so-named „communicators") through software (where it is sometimes necessary to change the company's ERP system) to its logical, standardized structure of the document format. This is an integrated solution, where two partners of business have to have the EDI standard system.

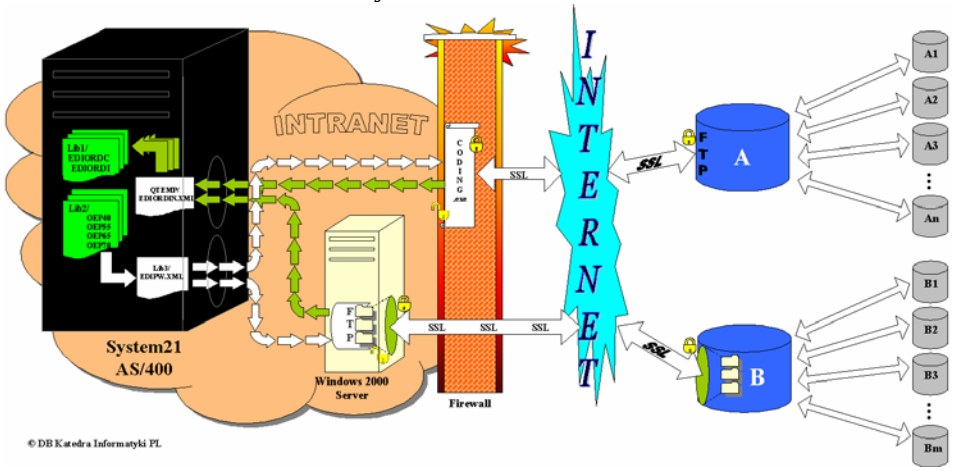

Fig. 1. Model of EDI/XML solution in the presented company

In the XML it is necessary to complement its functionality for almost everything: from a way of importing and exporting of XML documents from your ERP system, through logical system of documents circulation, to the physical and secure way of documents exchange. All this work is done by the company computer scientists.

Fig. 2. The logical structure of the orders' XML file example - provided by firm A

In this article the author will describe his experience from the initiation of electronic data interchange with the XML technology, in one of Polish companies specialized in the pasta production. At about 90% of company' clients there are wholesalers and hypermarket chains. The most frequently exchanged types of financial documents are: an order, dispatch advice and an invoice.

The company Enterprise Resource Planning (ERP) system in the System 21 is provided by Geac [10]. System 21 is implemented into AS/400 machine with OS/400 operating system and integrated IBM DB2 date base [11]. System21 is the fully integrated distribution and manufacturing solution, which enables the company to streamline customer service, production, distribution and financial procedures [9].

The necessity of electronic data interchange, as a result of trade contracts, became a fact last year, and it is necessary to point out that: there were only two months for putting it into practice.

There are two different computer programming firms which provide electronic data interchange solutions, and which are preferred by company clients. Firm A provide service for one group of clients and firm B for the other (see Fig. 1). Both firms, A and B, offer few formats of electronic files of data exchange: structured TXT, XML and others. Moreover, each firm offers another

kind of logical way of data delivery. Firm A keeps the FTP server, where its clients (e.g. our company) may upload and download their data as physical files. The data interchange must be served through a secured line and here a coding program provides the security. This program just codes and encodes the ports, which are used during the connection. The connection is secured by the coding program in Secure Socket Layer (SSL) [2,9] standard (see Fig. 1 “solution A”). An example of logical structure of XML files delivered by this firm is presented in Fig. 2. One physical XML file may include many documents from many customers of the company.

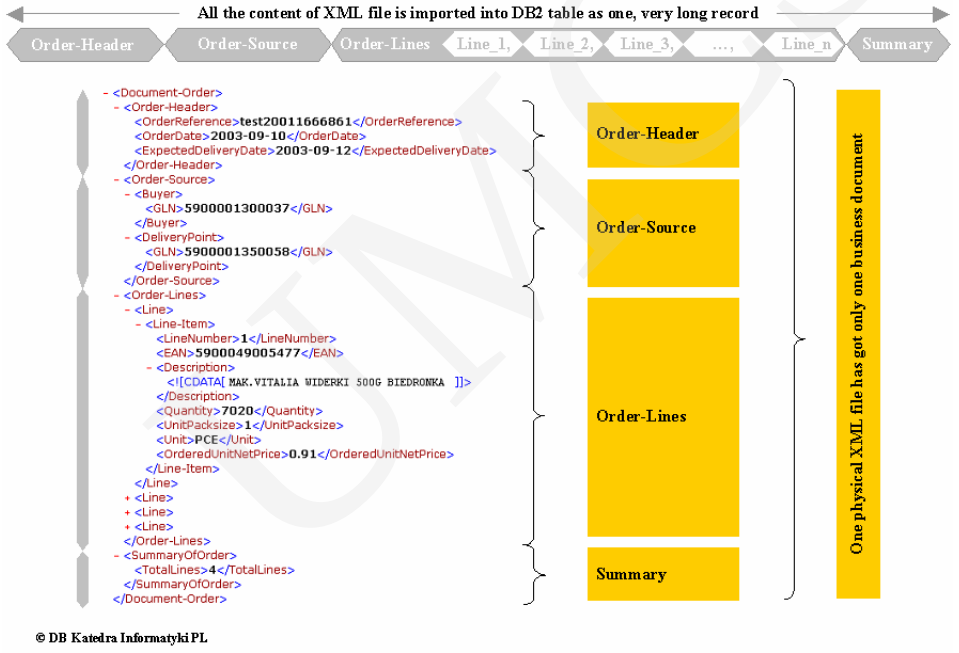

Fig. 3. The logical structure of the XML file example - provided by firm B

Firm B delivered its own program, which is responsible for documents exchange. This program has to be installed on the Windows server (see Fig. 1, “solution B”), and the delivered documents are sorted into separate directories on the local drive. This way there is one more point “in the data delivery chain” where the possibility of some faults may appear. But this program codes and encodes documents in the SSL standard itself and it offers the possibility of data backup and events log. The word “document” is used here, because, in the firm B’ solution, the logical structure of physical XML file includes only one business document (see Fig. 3). So into the directories there are delivered as many files as company customers have sent. There is another difference between the electronic data interchange solutions, and the company IT department

performing it. Of course each firm has its own names of tags and even more, firm A uses attributes of tags for the data addressed in XML file. This is in contrast to firm B where only tags are used.

This is one of the paradoxes of XML technology. The flexibility being one of XML advantages is also its disadvantage, because of many local separate solutions. The facility of data addressing and ability to transpose between the operating systems makes XML very attractive for business partners to implement their e-business. The partners usually interchange the electronic data. They implement their own structure of XML files where names of tags, quality and quantity of data exchanged are adjusted to their own ERP systems. This customization and personalization of information makes easier the business to business process. But it works only if there are two partners of e-business. It becomes the problem when a new partners comes with his own electronic data interchange solutions. This problem is called “The Tower of Babel” effect [12] because of many local customized implementations of XML.

2. Assumption

Looking for the best solution for our company we make the following assumption: ‘No external program, only System 21 takes control of import and export of XML business documents’. The main advantage of this decision is reliability of transactions; a document is imported successfully, or is not imported at all. Over that we reduced points of possible damages.

```
…
OVRDBF FILE(INPUT) TOFILE(QTEMP/FTP_CMD) MBR(FTP_GET)
CLRPFM FILE(LIBDB/FTPSRC) MBR(EDIOR_ILOG)
OVRDBF FILE(OUTPUT) TOFILE(LIBDB/FTPSRC) MBR(EDIOR_ILOG)
FTP &SERWER
…
```

Listing 1. Overwrite the standard input and output of FTP commands

```
SELECT CONO, LP, RECOD, RECDO,
CAST(TRIM(SUBSTR(EDIORDXML,
(POSITION(    '<tag>'       IN UPPER(EDIORDXML))
+ LENGTH(     '<tag>'       )), (LOCATE('</tag>', EDIORDXML,
(POSITION(    '<tag>'       IN UPPER(EDIORDXML))
+ LENGTH(     '<tag>'       ))) - 1)))
AS CHAR(3)) AS nr_lin
FROM "QTEMP"/&COND1
WHERE (POSITION('<tag>'    UPPER(EDIORDXML)) <> 0)
```

Listing 2. The example of SQL command locates and parses the contents of tag names “tag”

Of course there are some disadvantages of this decision too. Main disadvantage is that OS/400 has no implemented XML parser, a program which is responsible for handling XML syntax and, if desired, checking the contents of the document against constraints established in a document type definition (DTD) or schema [13]. Of course IBM offers a tool named DB2 XML Extender

[14], which provides new data types to store XML documents in DB2 databases and new functions to work with these structured documents. Managed by DB2, these documents are stored as character data or external files. But this tool is accessible for OS/400 platform only from the R5.02 version, where we have version R4.52 in the company. But owing to short time to "death line" and being experienced in the system upgrades – we decided to make a parser ourselves; so all the work of recognition remains to be done.

The first problem was: "How to import XML documents into relative data base?"

We started up an FTP service onto the Windows 2000 Server and shared the directories used by the B firm program. This way we reduced the problem into the model: 'FTP server – DB2 table', which from the OS/400's point of view is common for both solutions. The additional argument for this move was that using ftp command '***get***' with '***(replace***' parameter (see Fig. 4) it was possible to download the XML documents into a single column DB2 table. To run FTP commands in a batch program it is necessary to change the standard input to a file (see listing 1). The dynamic exchange of the file contents allows us to chose another source (server, file) of business documents.

The downloading process of XML files runs in three steps (see Fig. 4). In the first step the names of files are read. In the second step the files are downloaded and processed one by one. After the correctly ended process the name of file is passed to the third step, where during one connection the correctly imported files are removed to backup directory. This way the correct documents are imported successfully and incorrect ones remain on the server for future actions (to correct errors and import again).

3. The process of parsing an XML business document

The process of parsing an XML business document is more complex because of differences between the structures of files dispatched by the EDI providers. In the process of parsing incoming XML files depend on their structure. The files provided by the firm A are pre-parsed into separated tables with a single business document in it. This process runs in two steps: at first the downloaded file is decomposed into the parts of data of single company customer, and in the second step each part is decomposed into single business documents (see Fig. 4). This way we have some separated documents with the data of customers' orders (one order per one document/table), which is similar to the solution of the B firm. At that point the documents are parsed and the data are recognized into the flat multi column table. Now the data are validated and information about possible errors is written into a logs table. The correct data (only if the whole document is correct) are copied into the relative table of the working data base.

All the process until the coping is run in the QTEMP library, so there is no problem with the name of file duplication, because the library is cleared at the

end of task. That allows import of different structured XML files – it is impossible to import external file into a static OS/400 table, because if length of lines inside the file is longer than the size of column of that table – the data will be cut off. For each new file a new table has to be created to download the file, and the ftp command ***get*** takes care of that.

To parse a XML document the SQL build-in functions are used (see listing 2). The algorithm allows to locate and sub string contents of current tag.

In the example there is parsed some table from *QTEMP* library. The table with the name presented by variable *&COND1* has got only one-column name "*EDIORDXML"*. There is located and parsed a tag named "tag".

Table 1. Example of data in the configuration table

Solution	Name of tag/attribute	Tag/attribute	Name of column in System 21	Type of data	Size
A	OrderReference	T	CUSO40	VARCHAR	26
B	lnr	A	CUSO40	VARCHAR	26
A	OrderDate	T	DTCO40	DATE	10
B	data_zam	A	DTCO40	DATE	10
A	ExpectedDeliveryDate	T	DTTR40	DATE	10
B	termin_real	A	DTTR40	DATE	10
A	Quantity	T	EQTY55	NUMBER	15.3
B	il_jedn_zam	A	EQTY55	NUMBER	15.3

The physical structure of temporary tables, with the documents generated by a system of the B firm, allows us to recognize the whole business data during one turn of the parsing process. They have the data described by the tag attributes in this way that the whole data of concrete line are imported as one record. So there is no problem with the correct data combination, the whole data of the same RCN (record number) concern the same order line number. The physical structure of temporary table, with the document generated by a system of the A firm, needs more complex process. The whole document is imported as one long record, so at the first turn of process only the tag name "*TotalLines"* is recognized. After that the process of parsing is repeated as many times as many lines of document are.

For mapping the data to the concrete column of the target flat table and for valuating them, there is defined a configuration table (see Table 1). It allows us for data mapping over the differences between each solution.

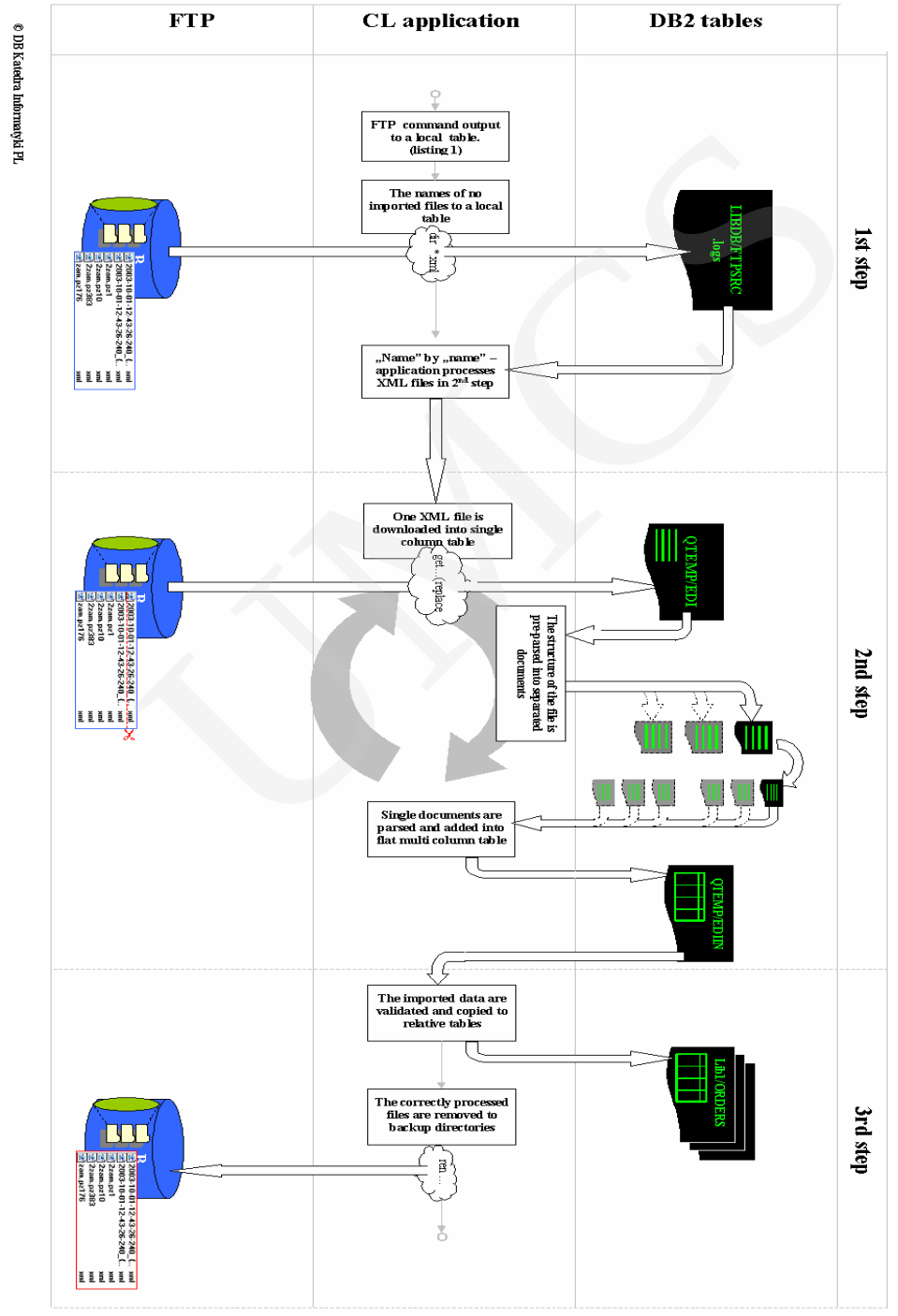

Fig. 4. The process of import of XML business documents to System 21

4. The advantages and disadvantages

The advantages of presented implementation of electronic data interchange are:

- XML is used as a new technology of data description,
- whole process of data downloading and importing to the company's ERP system runs as a butch job of that system – in this way it is controlled and reported by the same machine,
- the data imported from the XML files are accessible at the same moment – users of System 21 views no differences between working with the system tables and those with the external data,
- the quality of imported data is very high – only correct business information is imported; incorrect files remain on servers and after the correction could be imported again.

Main disadvantage of that implementation is close dependence of algorithm on logical and physical structures of XML file sent by the EDI providers. The next disadvantage is a still existing human factor, but until now there is no way to pass over that point of the downloading data algorithm. Still the role of controller belongs to an end-user and he has to take care of data and to correct the errors.

The other one is aforementioned "The Tower of Babel" problem. But this problem is unavoidable in so diverse trade of computer science solutions.

5. The results and what remains to be done

The results show that the XML technology was adapted as the data describe the format of business documents (orders, invoices). The process of electronic data interchange was implemented with no modification necessity of the company ERP system – that reduced very much the costs of the EDI venture. There was worked out the algorithm of the XML files decomposition into relative tables of DB2, with no external parsers, only SQL and CL programming is used. The best result is that it works and the number of EDI serviced documents still grows (see Fig. 5).

The future research will go to standardization of structure of exchanged XML documents. At now the greatest interest is turned into the UN/EDIFACT semantics which is promoted as the standard by the EAN International [15].

There are also works on the algorithm independent of physical structure of the exchanged XML file. Over that there are works on implementation of kind of business communicates as Orders In and Dispatch Advice another.

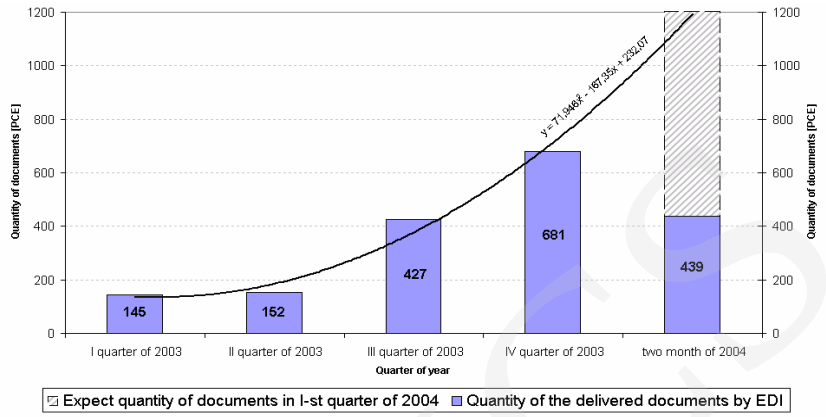

Fig. 5. The graphical representation of the number of the delivered documents by EDI

6. Conclusions

The XML is quite new but very expansive technology. It is already used in almost each kind of data description and exchange activity, from simple games (e.g. Quake Arena), thorough – those discussed in the article – the electronic business documents interchange, up to scientific data (e.g., DNA sequences). And the same the quantity of available data in the XML format grows and will grow. The necessity of querying, sorting and storing those data comes and as it is common knowledge that the relative databases do those jobs the best, and therefore a barrier between the relative data bases and the XML data should be reduced. Especially the process of data mapping from a XML file into relative tables of data base could be invisible (transparent) for the users.

XML file-importing method presented by the author unfortunately still expects from a user his knowledge about the logical and physical structure of that file. It is prepared for customized configuration of DB2, System21 and the specification of the XML business documents. It will be good to find out if other firms (e.g. Oracle's XML Developer's Kit [16] or Tamino [17]) solve that problem and next – how.

Acknowledgement

The author expresses thanks to his wife Katherine for patience and understanding of all his research.

References

[1] http://www.esupplychain.pl/
[2] Diedrich R., Patterson L.M., *Solving the XML Busines Data Exchange Dilema*, iSeries NEWS, (2001) 43.
[3] Hoffman S.L., *XML Blueprint*, iSeries NEWS, February (2003) 20.
[4] http://www.w3.org/MarkUp/SGML/

[5] http://xml.gov/
[6] Holzner S., *XML Vademecum profesjonalisty*, Helion-Poland, (2001) 5, in Polish.
[7] Harold E.R., *XML Bible*, Helion-Poland, (2000) 46, 613, in Polish.
[9] Bober D., Muryjas P., *Aspects of XML Technology in eBUSINESS transactions* in *4th International conference of PHD students, Engineering Sciences I*, University of Miskolc, Hungary, ISBN 963 661 590 X, (2003).
[10] http://www.geac.com.pl/
[11] http://www306.ibm.com/software/data/db2/
[12] http://www.microsoft.com/presspass/features/2000/jul00/07-11.NETframework.asp
[13] http://msdn.microsoft.com/xml/
[14] http://www306.ibm.com/software/data/db2/extenders/xmlext/
[15] http://www.ean.pl/
[16] http://otn.oracle.com/tech/xml/xdk/xdkprod.html
[17] http://www.softwareag.com/tamino/