

# Developing an XML Gateway for Business-to-Business Commerce

Shi-Ming Huang Irene Kwan\* David C. Yen\*\* Shang-Yuen Hsueh  
 Department of Information Management, National Chung Cheng University, Taiwan  
 Department of Information systems and Computing, Brunel University, London, England\*  
 Dept. of Decision Sciences and Management Information Systems, Miami University, USA\*\*  
 Email: smhuang@mis.ccu.edu.tw

## Abstract

*Business-to-business (B2B) Electronic Business (EC) is speedily getting popular and companies are increasingly using this notion and technology for business transaction and data exchange among each other. Traditionally, using Electronic Data Interchange (EDI) on Value Added Network (VAN) is common for data exchange on B2B EC, however, drawback are the trade-off between flexibility and efficiency among partners in the whole supply chain; cost is often an issue due to the lack of standard in information exchange. Now, the popularity of XML (Extensible Markup Language) helps companies to solve the problem because of its economical cost and flexibility. In this paper, a model for data exchange is investigated so that the cost for information flow could be reduced and the flexibility of the supply chain could be improved.*

## 1. Introduction

EC means all kind of transactions or other business activities on network. It is clear that the application for EC must be built on the existing technology infrastructure (i.e. large amount of computers with communication software on network). Table 1 shows the framework and some applications for EC [1].

Generally, There are two kinds of EC: B2B EC and Business-to-Customer (B2C) EC. B2C EC, also called Electronic Business, means the promotions from enterprises to customers. Because the point is "promotion", it can be work on Internet now because most of customers using Internet but not VAN. On the other hand, B2B EC, also called Quick Response (QR) or Efficient Customer Response, means the integration of information among partners in whole supply chain, so using EDI on VAN is the main way now for data exchange with each other [2][3][4].

Domain Issue	Application Issue
EC Application	<ul style="list-style-type: none"> <li>Supply Chain Management (SCM)</li> <li>Video on Demand (VOD)</li> <li>On Line Marketing and Advertising</li> <li>Home Shopping</li> </ul>
Common Business Service Infrastructure	<ul style="list-style-type: none"> <li>Security and Authentication</li> <li>Electronic Payment</li> <li>Directories and Catalogs</li> </ul>
The Information Superhighway Infrastructure	<ul style="list-style-type: none"> <li>Telecom, Cable TV, Wireless Network and Internet</li> </ul>

Table 1: The framework and some applications for EC [1].

QR is a version of Just-in-Time purchasing tailored for retailing. To avoid being over stock in the warehouse and under stock on orders, retailers are implementing QR System because of its flexible response in producing orders and lower inventory cost. In retail space, QR retailing focuses on market responsiveness while maintaining optimize level of stocks. It can then create a supply chain between retailer, vendor, and consumer, thus exchanging information through network such as ordering information [1]. Figure 1 shows the framework for QR chain.

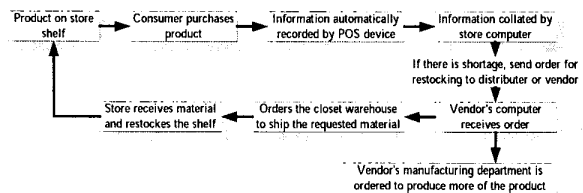


Figure 1: The Quick Response Chain [1].

Communication in the QR chain has always been an issue. Indeed EDI has improved communications among companies but its high cost and inflexibility were an insurmountable barrier for many small businesses. These new business strategies also create side effects. For instance, the vast investments need to implement EDI has lead to manufacturer/retailer tended to reduce the number of its supplier and move toward single sourcing. This is an undesirable in business sense. In result, many academic and industry organizations such as the Supply-Chain Council are seeking a new technology to improve the EDI technology.

This paper proposes a model for the information flow of the B2B EC. The model advocates the use of XML (Extensible Markup Language) as a mechanism to represent messages. It allows each partner of a supply chain to has its own data exchange format by adopting an XML meta-data management system in local side. In the next section, we first give a brief introduction of the information flow technology for B2B EC. In section 3, we discuss how XML is justified as a solution for the information flow of B2B EC. Our novel model is then described in section 4. A prototype system is presented in Section 5 to demonstrate and evaluate the feasibility of our research. Finally, conclusion and future directions are presented in Section 6.

## 2. Technology for XML Gateway

In 90's, many enterprises have invested heavily on B2B EC for automatic business data exchange between their partners. It is targeting the share of information within a heterogeneous environment. Without uniform standards of EDI, it will have  $N * N$  data exchange ways among  $N$  systems. The system will become very complexes and difficult to manage and maintain. As such, a well-structured EDI is indeed an important topic for the information flow in B2B EC.

### 2.1 Traditional EDI

The EDI standard means to develop uniform standards for inter-industry electronic interchange of business transactions. The standard of EDI, uses a uniform format to exchange the business information, is to represent the items of data (such as the name and price) and the business information (such as order form, request form). The feature of the EDI system is used for exchanging business information between enterprises by the electronic communication. We therefore apply the definition of EDI as "A protocol that facilitates the transfer of business information in a consistent electronic standard format" in our research [1].

EDI must use a common protocol to implement the connection among each system. A common protocol is

indispensable when communicating among each organization. Figure 2 shows the connection using EDI.

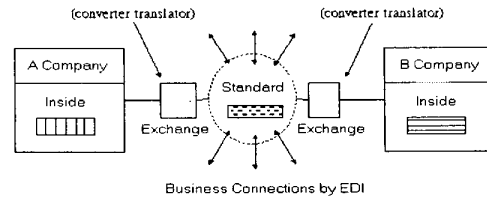


Figure 2: Connection using EDI [5].

### 2.2 Web-based EDI

A traditional EDI usually works in the VAN. In order to reduce the cost of the expensive service of VAN, many vendors bring up the idea of web-based EDI. The web-based EDI needs to provide the functionalities, such as security, transactional integrity, connection stability, authentication, and non-repudiation of messages etc. These functionalities, however only partially solve the drawbacks of the traditional EDI implementation, such as the high cost involved in message transport [6]. To completely solve the drawbacks of EDI, many enterprises and standards organizations are investigating a universal data format that can minimize the need of customization when electronic business document is exchanged between trading partners.

### 2.3 Extensible Markup Language (XML)

The XML initiative emerged under the auspices of the World Wide Web Consortium (W3C) in 1996. It is a kind of meta-languages to define the syntax of other languages, such as HTML (Hyper-Text Markup Language). The main difference is HTML defines the way showed on the WWW Browser, but XML define the markup syntax. [7][8]

XML indeed have some advantages, such as strai-ghtforwardly usable over the internet, support a wide variety of applications, easy to write programs which process XML documents, prepared quickly, formal and concise design and easy to create document. As such, XML is apparently a good candidate for data exchange.

A XML message is consisting of two parts: the message content and the message definition. The message content contains the information carried in the message. The message definition specifies the structure of the message content as well as provides meanings to different parts of the message. Formally, the message definition is referred to as a Data Type Definition (DTD). Figure 3 below shows an incomplete DTD. It specifies that the message content can contain at least one ("+") of "book" element. Each "book" element in turn contains the "isbn" and "bookname" elements. The "isbn" element

is specified as being a string (“#PCDATA”). Two attributes are given to the “bookname” element to specified as being a string (“#PCDATA”).

The message content is represented using elements that may be nested. Each element begins with a start tag and finishes with an end tag with information in between the two tags. Figure 4 below shows the content of a message, which carries information about a list of two books. The isbn of the first book is captured in the “isbn” element, which begins with the start tag “<isbn>” and finishes with the end tag “</isbn>” with the isbn of the book “957-8239-39-4” in between the two tags.

```
<!ELEMENT books (book)+>
<!ELEMENT book (isbn,bookname)>
<!ELEMENT isbn (#PCDATA)>
<!ELEMENT bookname (#PCDATA)>
```

Figure 3: An Example DTD.

```
<?xml version="1.0"?>
<!DOCTYPE Books SYSTEM "Xbook.dtd">
<books>
  <book><isbn>957-8239-39-4</isbn>
  <bookname>XML in Action</bookname> </book>
  <book><isbn>957-717-3389-4</isbn>
  <bookname>VB5</bookname> </book>
</books>
```

Figure 4: An Example of XML message.

### 3. XML as An Universal Data Format for Data Exchange

Much research has focused on XML/EDI to provide a new standard EDI lately [9][10]. In our literature survey, XML involves the following characteristics that can solve the problem encountered in the information exchange of B2B EC [11][12]:

- **Readability:** XML can express both data (i.e. XML message) and syntax (i.e. DTD). People and systems are easy to read and understand its data semantic and meaning.
- **Popularity:** The transmission of XML data is based on Internet. Internet technology has been widely used and adopted in most companies.
- **Flexibility:** XML has a characteristic of self-defined and self-description while EDI must have a special data format. The semantic of a XML message is depended on its DTD file. Each company can have their own DTD files to represent the same semantic. For example, different companies may have different order forms.
- **Heterogeneity:** XML is straightforwardly usable over the Internet. It has also been widely supported by a variety of applications.
- **Hyper-textability:** XML can transmit not only text

document but also multimedia data while traditional EDI can only transmit simple and formatted documents.

- **Cheapness:** A XML document can be transmit directly on Internet. It is not necessary to have additional VAN and software/hardware cost.

### 4. Model for Data Exchanging Using XML Gateway

#### 4.1 A Novel Model

The structure for data exchange using XML Gateway as shown in Figure 5 presents the novel substance in this paper. In the model, the XML document includes a DTD file and a XML-Message file. A DTD file contains the schema of the XML document. A XML-Message file contains the data of the XML document. There are five parts in the XML Gateway:

- **DTD Repository:** The part is used to store metadata from different DTD definition files.
- **Equivalence Repository:** Equivalence Repository is a sub-database of DTD Repository. It is used to store information of equivalence rules among different DTD File.
- **Local Database:** The part stores data coming from XML-Message file, such as the title or author of a book.
- **DTD Translation Module:** The module translates the source DTD file into individual metadata and stored into the DTD Repository.
- **Data Conversion Module:** The module has two functions. One is to discover the equivalence rules between the external DTD and local DTD. The function will store the equivalence rules into Equivalence Repository. The other is to convert the XML-Message file individual data by following its equivalence rules and store them into the local database.

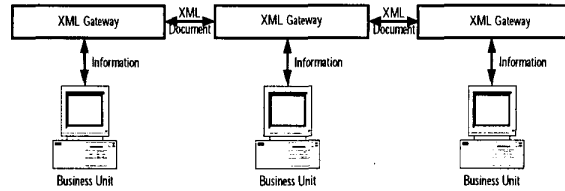


Figure 5: Structure for Data exchange using XML Gateway.

The model focuses on the data exchange of two partners - a sender and a receiver. This is basic environment of B2B EC. Figure 6 illustrates the process flow of our model and described as the following:

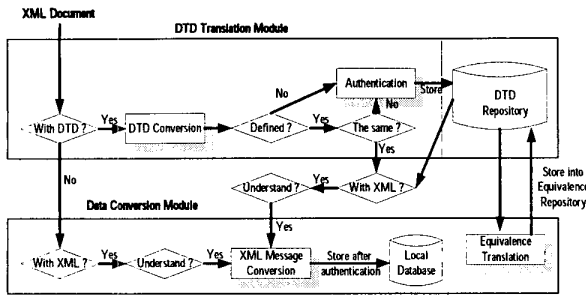


Figure 6: Process for the novel model.

When receiver received a document, it will firstly check the document contents. If the document doesn't contain any XML-Message or DTD files, the receiver will just give up it. If there is a DTD file in the document, the DTD Translation Module will parse the DTD file into metadata. It will then check if the DTD has already existed in our DTD Repository. If it has existed, then the system will compare the two metadata. If they are different, the authentication to the sender for this metadata is required to make sure if the sender will update the metadata. On the other hand, if the DTD does not exist in the local DTD Repository, authentication is also needed to make sure that the sender sends a new DTD. Finally the metadata will be stored into the DTD Repository for the DTD Translation.

After the new DTD has been stored into the DTD Repository, the model will process the Equivalence Translation function. The function will generate the equivalence rules and store these into Equivalence Repository. During the process, the interaction between the sender and the receiver is needed to confirm the equivalence rule is correct.

If the document contains a XML-Message file, our model will check if its DTD has existed in the DTD Repository. If answer is true, the system will retrieve its equivalence rule from Equivalence Repository and process the data conversion module. The data conversion module will convert the XML-Message file into the local database by following the equivalence rule.

#### 4.2 A DTD Repository

To parse the DTD files correctly, a schema for the DTD file is required. Figure 7 describes a DTD meta-data schema by using EER Model. The schema contains 19 meta-database-entities and 10 meta-relationships. It can be classified into four parts - Processing Instructions, Comments, Entities with its Notations, and Elements.

- (1). **Processing Instructions:** The Entities PROCESSING\_INSTRUCTION and ATTRIBUTE\_OF\_PI represent the processing instructions and its attributes in DTD File.

- (2). **Comments:** The Entity COMMENT represents the comments in DTD File.
- (3). **Entities with their Notations:** An Document Type may have Entities and NOTATIONS. In our EER Model, entities ENTITY and NOTATION represent these. We have two kinds of entities here: REF\_ENTITY for outside-entity and DEF\_ENTITY for inside-entity and parameter-entity.
- (4). **Elements:** Another part in a document type is Element. Here we represent it by entity ELEMENT. ATTRIBUTE\_ELEMENT represents normal element and CLASS\_ELEMENT represents the attribute list of an ATTRIBUTE\_ELEMENT.
- (5). Of course, in order to **identify** the relationship of an DTD File, we also have defined the entities: COMPANY and DTDFILE to store the DTD File Name and its owner (Original Company define and send the DTD File).

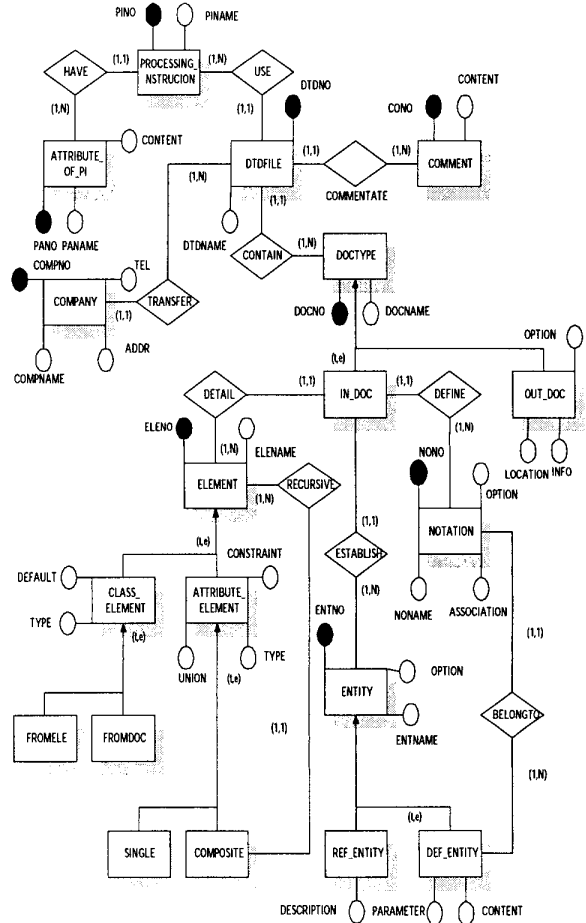


Figure 7: The EER Model for DTD File of XML.

### 4.3 DTD Translation

We introduce an algorithm to parse the XML Document and store items into DTD Repository (i.e. DTD Translation) in this section.

#### Pre-STEP: Discover the basic Information

First step for parsing DTD file is to get the information of sender, such as the sender's name, address, and telephone number, etc, then parses and stores them into local database if not in yet. Second, the information of the DTD file, such as file name, will be parsed and stored into database. Third, the name and content will be simply parsed and stored into database COMMENT.

In each following step, the DTD file will be scanned and the information needed in each step will be parsed and stored into corresponding database.

#### STEP 1: Parsing The Processing Instruction

When a processing instruction is found during scanning, first put its name into database PROCESSING\_INSTRUCTION. Then its attributes will be parsed and stored into database ATTRIBUTE\_OF\_PI. Algorithm of the process is shown as below:

#### STEP 2: Parsing The Document Type

There are two kinds of document types in XML format: Inside and outside one. So when a tag for document type declaration is found during scanning, the basic and common attributes, such as its name, will be stored into table DOCTYPE. But if the document type is an outside one, other attributes will be stored into database OUT\_DOC. On the other hand, the other attributes of inside document type will be stored into database IN\_DOC.

#### STEP 3: Parsing The Notation

When a tag of notation declaration is found during scanning, the whole declaration will just be parsed and stored into database NOTATION. Algorithm for the process is shown as below:

#### STEP 4: Parsing The Entity

There are three kinds of entities in XML format: Inside, parameter, and outside one. So when a tag for entity declaration is found during scanning, the basic and common attributes, such as its name, will be stored into table ENTITY. First, if the entity is an inside one, just put the content into database DEF\_ENTITY and the parameter field of the row will be set with EMPTY because inside entity is a definition without parameter. In similar way, other attributes, including the parameter attribute, of parameter entity will be stored into database DEF\_ENTITY. But if outside entity is found, other attributes of the entity will be stored into database REF\_ENTITY because outside entity is actually a reference entity to other DTDs. The algorithm is presented in next page.

#### STEP 5: Parsing The Element List

There are several kinds of element in XML format, and

would need to be discussed separately. When an element was found during scanning, the name of the element will be stored into database ELEMENT first. Attribute element means the declarations with tag <ELEMENT>. If it is found, first the basic attributes will be parsed and stored into database ATTRIBUTE\_ELEMENT. If there are sub-elements in the element (i.e. <!ATTLIST>), its name will be stored into database COMPOSITE as well. If not, its name will be stored into database SINGLE. Class element is another kind of element. There are two kinds: contents and attribute lists. If it is found, first the basic attributes will be parsed and stored into database ATTRIBUTE\_ELEMENT. If it is content element, the basic attributes of it will be stored into database FROM\_DOC. On the other way, the name of attribute list will be stored into database FROM\_ELEMENT because it is a sub-element of an element. The algorithm will be shown in next page.

### 4.4 Data Conversion

Due to the heterogeneity of the data between different XML documents, the conflict resolution is required for our model. In this research, the data heterogeneity is solved by use equivalence rules. An Equivalence Repository is defined to store the equivalence rules between different XML documents. The database for Equivalence Repository is shown in Figure 8. There are four types of equivalence:

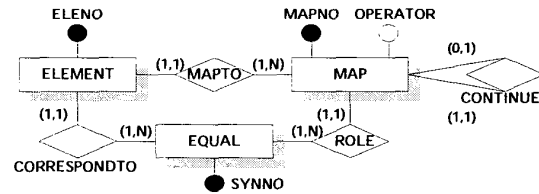


Figure 8: EER Schema for Equivalence Repository.

- (1). **One-To-One:** For example, the meaning of <BOOK> in DTD of company A is the same as <BOOKNAME> in local DTD repository. In this case, DTD repository should only directly map <BOOK> to <BOOKNAME>. When Company A sends XML data with tag <BOOK>, just stored into corresponding table of <BOOKNAME> in local database. Figure 9 shows the metadata for this example.

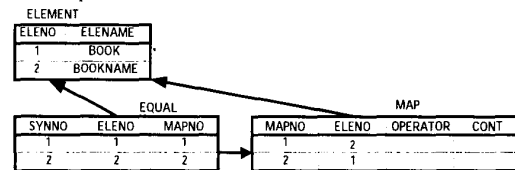


Figure 9: One-To-One Equivalence metadata.

- (2). **One-To-MultiObject:** For example, the content of <BOOK> in DTD of company A is the combination of <BOOKNAME> and <ISBN> in local repository. In this case, the process for separating is required (In next section, there will be a more detail example) and the relationship can be expressed with the following equation:

$$A.BOOK = this.BOOKNAME + this.ISBN$$

So when company A sends XML data with tag <BOOK>, the content must be parsed into two parts: bookname and ISBN part and stored into corresponding tables of <BOOKNAME> and <ISBN> in local database. Figure 10 shows the metadata for this example.

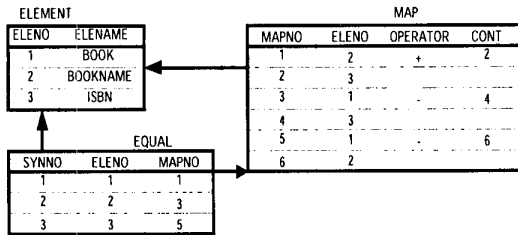


Figure 10: One-To-MultiObject / MultiObject-To-One Equivalence metadata.

- (3). **MultiObject-To-One:** For example, the content of <BOOK> in local repository is the combination of <BOOKNAME> and <ISBN> in DTD of company A. In this case, the process for combination is required (In next section, there will be a more detail example). So when company A sends XML data with tag <BOOKNAME> and <ISBN>, just combine the contents and stored into corresponding table of <BOOKNAME> in local database.
- (4). **MultiObject-To-MultiObject:** The way for solving the problem is the combination of 2 and 3. For example, if the combination of <BOOKNAME> and <ISBN> in DTD of company A is equal to the combination of <BOOK> and <NUM> in local repository, the metadata will be like Figure 11.

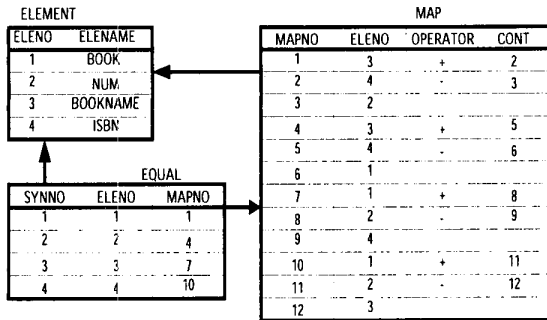


Figure 11: MultiObject-To-MultiObject Equivalence metadata.

## 4.5 A Case Study

The section represents a simple case study for information flow among bookstores for this research. There are two business units in the case, PUBLISHER as the publisher and STORE as the bookstore. Now the STORE will send its book order to PUBLISHER by using XML document. Figure 12 shows the DTD files for PUBLISHER and STORE.

```

<!-- Book.DTD by PUBLISHER -->
<!ELEMENT ISBN (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT INITPAGE (#PCDATA)>
<!ELEMENT ENDPAGE (#PCDATA)>
<!ELEMENT AUTHORS (MAN)*>
<!ELEMENT MAN (#PCDATA)>
<!-- ATTLIST POSITION CDATA #IMPLIED -->

```

```

<!-- Book1.DTD by STORE -->
<!ELEMENT BOOK (#PCDATA)>
<!ELEMENT AUTHORS (AUTHOR)*>
<!ELEMENT AUTHOR (#PCDATA)>
<!-- ATTLIST POSITION CDATA #IMPLIED -->

```

Figure 12: Two sample DTD Files.

In this case, publisher's DTD will be the local DTD and store's DTD will be the external DTD. Figure 13 shows the original data in the DTD Repository of the publisher.

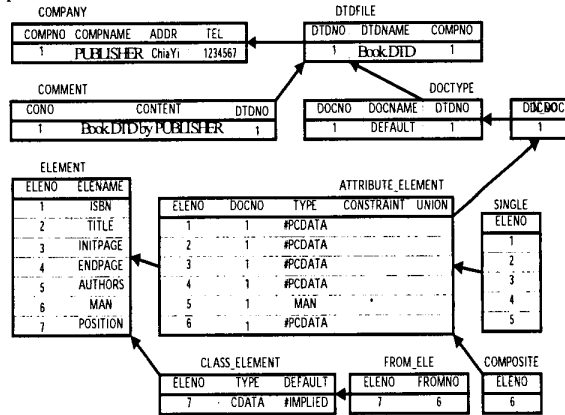


Figure 13: Original Database in the DTD repository of "PUBLISHER".

When the publisher's XML Gateway receives a XML document from the STORE, the XML Gateway will discover the basic information of the DTD file and found that it is a new DTD file.

The XML Gateway will then execute the schema translation module to parse the DTD metadata and store the metadata into the DTD repository (see Figure 14).

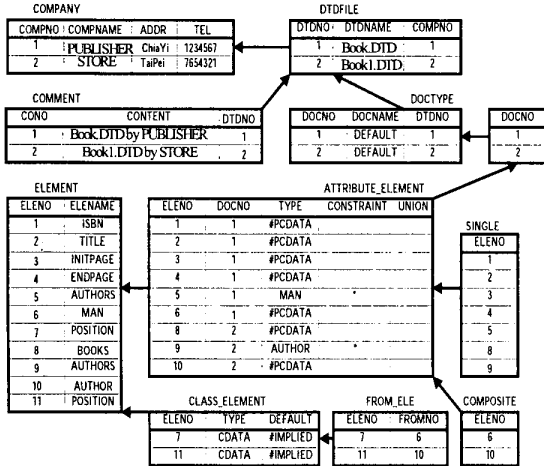


Figure 14: The metadata without equivalence rules in the DTD Repository of the publisher.

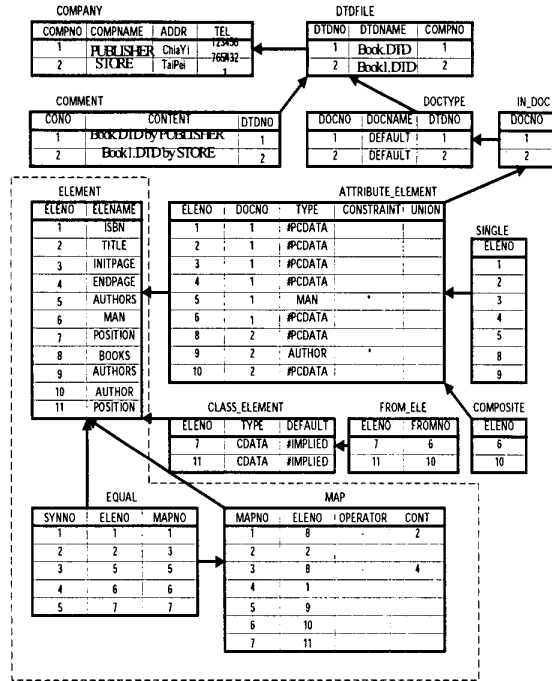


Figure 15: The metadata of "PUBLISHER".

The data conversion module will process the equivalence rules discovery function by interacting with STORE. For instance, if the equivalence rules are the following:

- ◆ The metadata <!BOOKS> of "STORE" is the combination of the two tags <!ISBN> and <!TITLE> in "PUBLISHER". In other words, we can write as:
  - This.TITLE = Book1.BOOKS - this.ISBN and*
  - This.ISBN = Book1.BOOKS - this.TITLE*
- ◆ The metadata <!AUTHORS> of "STORE" is the same as tag <!AUTHORS> in "PUBLISHER".
- ◆ The metadata <!MAN> of "STORE" is the same as tag <!AUTHOR> in "PUBLISHER".
- ◆ The metadata <!POSITION> of "STORE" is the same as tag <!POSITION> in "PUBLISHER".

The publisher's DTD Repository will be as Figure 15. The block part is the Equivalence Repository.

For instance, STORE sends the following XML-Message file as shown in Figure 16. Then the data conversion module will convert the above XML-Message file into PUBLISHER local database as Figure 17.

```

<BOOK>XML in Action 957-8239-39-4</BOOK>
<AUTHORS>
<AUTHOR POSITION="USA">William J. Pardi</AUTHOR>
</AUTHORS>
<BOOK>Electronic Commerce 957-97222-2-6</BOOK>
<AUTHORS>
<AUTHOR POSITION="Taiwan">Li, C.H.</AUTHOR>
</AUTHORS>
    
```

Figure 16: An example XML-Message file.

ISBN	TITLE	INI-PAGE	END-PAGE	AUTHORS	MAN	POSITION
957-8239-39-4	XML in Action			William J. Pardi	William J. Pardi	USA
957-97222-2-6	Electronic Commerce			Li, C.H.	Li, C.H.	Taiwan

Figure 17: Local database.

## 5. System Implementation

To prove our hypothesis, a prototype system is developed by authors. The system is designed by using Visual Basic and MS -SQL server. Figure 18 shows the

architecture of the prototype system. The interface of the system prototype will be like the interface shown in Figure 19.

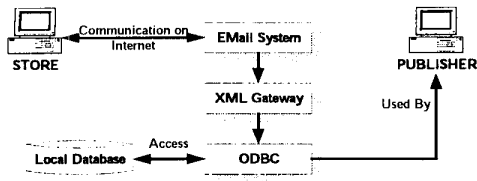


Figure 18: The process of the system prototype.

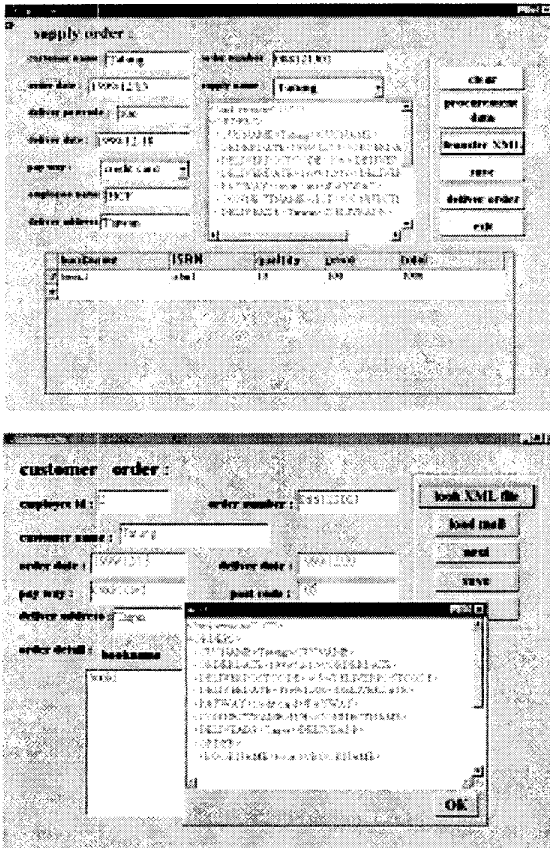


Figure 19: The Interface of system prototype

## 6. Conclusion

We can conclude that XML may be the trend of transmitting standard for information flow in B2B EC. Because of using XML, we can reduce the cost for establishing and training of EDI/VAN. The content of document can also be more flexible because of the user-defined data-type and the multimedia data, as such information can be easily shared and popular to each partner. In result, we use XML as the standard of the information transmitting to solve the problem of information flow in B2B EC. We vision that this kind of EC and transaction might be more feasible within B2B

EC and could becoming the standard approach in the near future.

## 7. Acknowledgement

The work presented in this paper has been supported by National Science Council, Taiwan, R.O.C, under Grant No. NSC89-2213-E-194-035. We deeply appreciate their financial support and encouragement.

## 8. Reference

- [1] Kalakota, R. and Whimston, A.B., Frontiers of Electronic Commerce, Addison-Wesley (pub.), 1996. ISBN: 0-201-84520-2.
- [2] Li, C.H., Electronic Commerce, BestWise (pub.), 1998. ISBN: 957-97222-2-6.
- [3] Wu, T.F. and Hsieh, C.C., Management Information Systems, BestWise (pub.), 1998. ISBN: 957-97226-6-8.
- [4] Kalakota, Ravi and Robinson, Marcia, e-Business Roadmap for Success, Addison-Wesley (pub.), 1999. ISBN: 0-201-60480-9.
- [5] Huang, S.M. et. al., A Digital Ordering System in MIME Protocol. Workshop of Commerce Automation, Taiwan, 1999.
- [6] Fu, Shiwa; Chung, J.Y.; Dietrich, Walter; Gottemmukkala, Vibby; Cohen, Mitchell; and Chen, Shyhkwei, A Practical Approach for Web-based Internet EDI, 19th IEEE International Conference on Distributed Computing Systems Workshops, 1999.
- [7] Extensible Markup Language (XML)1.0, <http://www.w3.org/XML/>
- [8] Laurent, Simon ST. and Cerami, Ethan, Building XML Applications. McGraw-Hill (pub.), 1999. ISBN: 0-07-134116-1.
- [9] Wong, P., The Role of XML in the Advancement of Business-to-Business Electronic Commerce. International Workshop of Electronic Commerce, Hong Kong, April 1999.
- [10] Murphy, P. and Daley, J., International Freight Forwarder Perspectives on Electronic Data Interchange and Information Management Issues. *Journal of Business Logistics*, 1996.
- [11] Huang, S.M., Li, S.H., and Hsueh, S.Y., Using XML to Solve the Problem of Information Flow in Supply Chain Management Systems. Conference of Western Decision Support Institute, 2000.
- [12] Pardi, William J., XML in Action, Microsoft Press, 1999.