

World Maritime University

The Maritime Commons: Digital Repository of the World Maritime University

World Maritime University Dissertations

Dissertations

2002

Application of Extensive Markup Language (XML) for shipping companies

Bala Vithoba Meshram
World Maritime University

Follow this and additional works at: https://commons.wmu.se/all_dissertations



Part of the [Databases and Information Systems Commons](#)

This Dissertation is brought to you courtesy of Maritime Commons. Open Access items may be downloaded for non-commercial, fair use academic purposes. No items may be hosted on another server or web site without express written permission from the World Maritime University. For more information, please contact library@wmu.se.

WORLD MARITIME UNIVERSITY

Malmö, Sweden

**APPLICATION OF EXTENSIVE MARKUP
LANGUAGE (XML) FOR SHIPPING COMPANIES**

By

BALA VITHOBA MESHARAM

India

A dissertation submitted to the World Maritime University in partial fulfillment of
the requirement for the award of the degree of

MASTER OF SCIENCE

In

SHIPPING MANAGEMENT

2002

DECLARATION

I certify that all the material in this dissertation that is not my own work has been identified, and that no material is included for which a degree has previously been conferred on me.

The contents of this dissertation reflect my own personal views, and are not necessarily endorsed by the University.

Bala Vithoba Meshram

September 2, 2002.

Supervised by: Associate Professor Patrick Donner

World Maritime University

Internal assessor: Professor Shuo Ma

World Maritime University

External assessor: Mr. Jan Månsson, IT Manager

BIMCO, Copenhagen

ACKNOWLEDGMENTS

I am very much thankful to IMO and Norwegian government for funding my studies at World Maritime University (WMU). I would also like to thank to WMU for giving me the opportunity to write this dissertation.

Special thanks to Prof. Shoe Ma for suggesting the interesting and thought provoking topic of XML and also for giving some useful ideas.

I appreciate the guidance, encouragement and supervision of Prof. Patrick Donner. He took painstaking efforts in studying each chapter thoroughly and gave his valuable suggestions. I am very much thankful to him.

Ms. Susan Wangeci-Eklow took efforts in borrowing books from other libraries in Malmo and Lund. Ms. Cecilia was regularly sending me the electronic version of DigitalShip. Mr. David Moulder added in the library a book on XML on my request. Ms. Inger Battista checked the manuscript. I am thankful to them.

Thanks to Prof. P.K. Mukherjee and Prof. Francou for suggesting some of the XML application areas in shipping companies. Many thanks to Mr. Leif Lund from Freightman AB, Mr. Patrick Hillenius from Gothenburg Chartering AB, Mr. Richard Poulsen from Lehmann Junior AS, Copenhagen, Mr. Michael Steiness from BIMCO, and also thanks to the executives of various shipping companies for sharing their views on IT requirements during my field study visits to Piraeus, Oslo and London.

The wonderful and stimulating atmosphere of Henrik Smith Hostel and the beautiful green Malmo City has helped in my studies.

The Internet is an excellent source for getting a lot of information on XML. I am thankful to all those websites publishers for sharing information.

My employer, The Shipping Corporation of India Ltd, allowed me to study for nine months with paid leave. I am indebted to it.

Finally, I am deeply grateful and thankful to my wife, Sarita, for bearing my separation otherwise I could not have written this dissertation. She had to take care for our two daughters, Ishita and Saniya and their studies in my absence.

Title of Dissertation: **Application of Extensive Markup Language (XML) for Shipping Companies**

Degree **MSc**

This dissertation enables shipping companies to understand the XML specifications and the implementation of XML based applications.

XML specifications consist of XML documents, the validation procedure, document presentation and the other related standards and technologies. The content of an XML document describes the business information one wishes to present. Elements and attributes are defined to identify contents by following the rules and syntax of XML specifications. The validation procedure of XML documents is created separately in a DTD (Document Type Definition) or an XML Schema document.

EDI message standards could be replaced with XML documents for exchanging transactions between business partners. XML has advantages over EDI messages in terms of cost saving, implementation time and support.

Shipping companies can implement the XML based systems for various categories of applications such as document management, e-business and web services. The XML system architecture includes XML processor, application server, database server and Internet connectivity. Shipping companies can implement XML based applications for container tracking, on-board systems management and data exchange.

It is concluded that shipping companies can easily implement XML based systems and can effectively improve business performance. The main recommendations are education and training to employees and implementation of XML based systems in a phased manner on the existing platforms.

KEYWORDS: XML, E-Business, Internet, Computerization, EDI, Web Services.

TABLE OF CONTENTS

Declaration	ii
Acknowledgements	iii
Abstract	v
Table of Contents	vi
List of Tables	ix
List of Figures	x
List of Abbreviations	xii
1 Introduction	1
1.1 Background	1
1.2 Objective of the dissertation	2
1.3 Methodology used	3
1.4 Limitations of the dissertation	3
1.5 Intended readers	4
1.6 Outline of chapters	4
2 Extensible Markup Language (XML) and its specifications	6
2.1 Introduction	6
2.2 History of XML	9
2.3 Comparison of HTML and XML	10
2.4 Specification of XML	13
2.4.1 Preparation of XML documents	14
2.4.1.1 Authoring tools for preparing XML documents	19
2.4.1.2 Explanation of XML document	19
2.4.1.3 Elements tags and parts	23

	2.4.1.4	Attributes	24
2.4.2		XML Document validation	25
	2.4.2.1	Document Type Definition (DTD)	25
	2.4.2.2	XML Schema	37
2.4.3		XML document presentation	44
	2.4.3.1	Cascading Style Sheets (CSS)	45
	2.4.3.2	Extensible Style Sheets (XSL)	45
	2.4.3.3	XML Linking Language (XLink)	48
	2.4.3.4	XML Pointer Language (XPointer)	48
	2.4.3.5	XML Path Language (XPath)	48
2.4.4		Related standards and technologies	49
	2.4.4.1	Namespaces in XML	49
	2.4.4.2	Document Object Model (DOM)	49
3		EDI and XML	51
	3.1	Introduction to EDI	51
	3.2	EDI definition	51
	3.3	Key elements of EDI	51
	3.4	Translation software	54
	3.5	Message standard	55
	3.6	Comparison between EDI and XML	56
4		Implementation of XML based systems	64
	4.1	Features of XML	64
	4.2	Type of applications	67
	4.2.1	Document management	67
	4.2.2	Business-to-business data exchange	68
	4.2.3	Channel Definition Format (CDF)	68
	4.2.4	Synchronized Multimedia Integration Language	69
	4.2.5	User defined markup language	69
	4.2.6	Web services	70

4.3	Areas in shipping companies	70
4.3.1	E-Business	70
4.3.2	Building of content management system and IETM	71
4.3.3	Ship-to-shore communication	73
4.4	Implementation of XML based systems	73
4.4.1	Preparation of XML document	75
4.4.2	Selection of application server	76
4.4.3	Selection of database server	76
4.4.4	Selection of platform	77
4.4.5	Choosing the mode of transmission	77
4.4.6	Decision on security policy	77
4.5	Case study on implementation of XML based systems	77
4.5.1	Objective	78
4.5.2	Description	78
4.5.3	Information flow	78
4.5.4	System architecture	79
4.5.5	XML documents used	81
4.5.6	Computer programs required	88
4.6	Benefits and limitations of XML	89
4.6.1	Benefits of XML	89
4.6.2	Limitations of XML	91
5	Conclusions and recommendations	94
5.1	Conclusions	94
5.2	Recommendations	97
	References	100
	Selected bibliography	106
	Appendix A Questionnaire on Application of XML for shipping companies	110

LIST OF TABLES

Table 2.1	Explanation of user-defined code	21
Table 2.2	The parts of an Element Tag	23
Table 2.3	The parts of an Element markup	23
Table 2.4	Specifying an Attribute	24
Table 2.5	Type of contents found in Elements	27
Table 2.6	Occurrence indicators in DTD	30
Table 2.7	Predefined entities	34
Table 4.1	Cross reference table of XML documents	81
Table 4.2	Explanation of Container delivery notice document	82
Table 4.3	Explanation of Ship arrival notice document	85
Table 4.4	Explanation of Ship berthing and container location document	86

LIST OF FIGURES

Figure 2.1	Example of Vessel information in XML document	6
Figure 2.2	XML for different outputs	7
Figure 2.3	Using HTTP to serve XML documents from web server	8
Figure 2.4	Using XML on the server but sending HTML to the browser	8
Figure 2.5	HTML code to display vessel information	11
Figure 2.6	XML code to display vessel information	12
Figure 2.7	Example of Vessel Information	14
Figure 2.8	Tree diagram of vessel information	15
Figure 2.9	Example of XML document for vessel information	16
Figure 2.10	Display of XML document on XML Web browser, Amaya	17
Figure 2.11	Display of XML document on HTML Web Browser, IE5.5	18
Figure 2.12	DTD of Vessel Information	25
Figure 2.13	XML Document for Vessel Information	40
Figure 2.14	XML Schema for Vessel Information	43
Figure 2.15	XML Schema declaration in XML document	44
Figure 2.16	CSS code for Vessel Information	45
Figure 2.17	XSL code for Vessel Information	47
Figure 3.1	EDI system	52
Figure 3.2	Communication Network Methods in EDI	53
Figure 3.3	Data structure of an EDI transmission.	54
Figure 3.4	EDIFCAT message to purchase engine spares	59
Figure 3.5	List of purchase order in XML	60
Figure 3.6	XML supports both users on a workstation and a server	61
Figure 4.1	E-business applications	71
Figure 4.2	Architectures and Technical Solutions for F117 Fighter	72
Figure 4.3	Common XML Architecture	74
Figure 4.4	Complete XML Solution	75
Figure 4.5	Information Flow Diagram	79

Figure 4.6	System architecture of container delivery system	80
Figure 4.7	XML Document for Container Delivery Notice	83
Figure 4.8	XML DTD for Container Delivery Notice	84
Figure 4.9	Arrival Notice XML Document	85
Figure 4.10	Arrival Notice XML Document DTD	86
Figure 4.11	Berthing and Container Location XML Document	87
Figure 4.12	Berthing and Container Location XML Document DTD	88

LIST OF ABBREVIATIONS

ANSI	American National Standard Institute
ASC	Accredited Standards Committee
ASCII	American Standard Code for Information Interchange
B2B	Business-to-Business
CDATA	Character Data
CDF	Channel Definition Format
CML	Chemical Markup Language
CORBA	Common Object Request Broker Architecture
CRM	Customer Relationship Management
CSS	Cascading Style Sheet
DOM	Document Object Model
DTD	Document Type Definition
ebXML	Electronic Business Extensive Markup Language
EDI	Electronic Data Interchange
EDIFACT	Electronic Data Interchange For Administration Commerce and Transport
ETSF	Electronic Trading Standard Format
FTP	File Transfer Protocol
GeoML	Geological Markup Language
HRMML	Human Resource Management Markup Language
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IETF	Internet Engineering Task Force
IIS	Internet Information Server

IMPA	International Marine Purchasing Associations
ISM	International Safety Management
ISO	International Standard Organization
IT	Information Technology
J2EE	Java 2 Platform Enterprise Edition
JCALs	Joint Computer Aided Logistics and Support
LDAP	Lightweight Directory Access Protocol
MathML	Mathematical Markup Language
MTML	Marine Trading Markup Language
MTO	Multi-modal Transport Operator
NSF	Notes Format
OASIS	Organization for the Advancement of Structured Information Systems
PC	Personnel Computer
PCDATA	Parseable Character Data
PDF	Portable Document Format
RDBMS	Relational Database Management System
RELML	Real Estate Listings Markup Language
RPC	Remote Procedure Call
RTF	Rich Text Format
SGML	Standard Generalized Markup Language
SMIL	Synchronized Multimedia Integration Language
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Application Protocol
SQL	Standard Query Language
STCW	Standards of Training, Certification and Watch keeping
SWIFT	Society for Worldwide Inter-bank Financial Telecommunication
TCP/IP	Transmission Control Protocol/Internet Protocol
UN/CEFACT	United Nations Center for Trade Facilitation and Electronic Business
UNCTAD	United Nations Conference on Trade and Development
URI	Universal Resource Identifier

URL	Uniform Resource Locator
VAN	Value Added Network
VoxML	Voice Markup Language
W3C	World Wide Web Consortiums
WML	Wireless Markup Language
WMU	World Maritime University
WSDL	Web Service Data Language
WWW	World Wide Web
XML	Extensive Markup Language
XSD	XML Schema Definition
XSL FO	XSL Formatting objects
XSL	Extensible Style Language
XSLT	Extensible Style Language Transformation

CHAPTER 1

INTRODUCTION

“By 2004, XML-related technologies will play a role in virtually every aspect of modern e-business service oriented architectures”

- Gartner, October 2001.

(as cited by Shmeitzer, 2002).

1.1 Background

Extensive Markup Language (XML) is the buzzword of the present Internet era. No Information Technology (IT) user can ignore this new exciting technology, which is just over four years old. The entire IT user community, which is specifically dependent on Internet, has found new hopes in current issues related to data exchange and content management. Large standard bodies, industry groups, vendors and the users around the world have come together to develop XML standards. Major IT companies like Microsoft, IBM, Oracle, Sun Microsystems, VeriSign, and Fujitsu and many institutions like UN/CEFACT (United Nations Center for Trade Facilitation and Electronic Business, <http://www.unece.org/cefact/>), Rosettanet and Bolero.Net (<http://www.bolero.net/>) have dedicated resources for the development and standardization of XML. These organizations among others are integrating XML based technologies in their own implementations and products as well as supporting XML (Kayl, 2001). A Google (<http://www.google.com>) search on XML shows hits of about 14 million web pages on the Internet. It shows the growing importance of XML in the present Internet technologies.

According to Marshall (2001), “XML is like the air that we breathe”. Levitt (2000) writes, “XML is the language of business”. While comparing with EDI, Pierce (2002) opinions “XML spells the end for EDI”. According to the Gartner Group, 70% of the business transactions will use XML in 2003 (Bruzek, 2002). American Shipper has published a special article (Forsyth, 2000) on XML mentioning that XML is breaking down IT barriers in logistics chain management. However, Vollmer (2001) cautions us not to believe the hype that EDI will end; and he writes that EDI and XML are the just perfect match for e-business. Kearns (2002) writes that XML is not a magic bullet for identity management, Web services or any other technology initiative. These are some articles, which highlight that XML is a new era in present Internet world. But questions arise, what is an XML? And why is it so important? Can XML help in solving certain IT problems of shipping companies?

Some of these problems are:

- Shipping companies are facing the difficulty of exchanging data with their business partners due to the proprietary format.
- Shipping companies are spending a lot of money in supporting EDI (Electronic Data Interchange) and maintaining the costly EDI network.
- Small shipping companies cannot invest in EDI since EDI is expensive and takes months for installation.
- Shipping companies cannot integrate their applications with the applications of their business partners due to heterogeneous platforms.
- Shipping companies cannot send information in many languages to computer applications.

XML may not be the solution to all the IT problems of shipping companies but XML has certain features, which can solve many IT problems of shipping companies.

1.2 Objective of the dissertation

The objective of this dissertation is to enable shipping companies to understand the implementation of XML based systems. The dissertation has detailed the basic

specifications for an XML document. Brief information is given on the XML related technologies so that shipping companies are familiar with those technologies for further exploration. Moreover, shipping companies using EDI can appreciate the benefits of using XML.

1.3 Methodology used

This dissertation is based on the source material available in books on XML, various periodicals and journals, and the information available in various online magazines and websites on XML. The author's 15 years experience in the IT field is also used for preparing this dissertation. The author has discussed XML technologies with shipping companies during field study visits to Piraeus, Oslo and London. The author has also visited BIMCO and Messers. Lehmann Junior A/S in Copenhagen, Freightman AB and Gothenburg Chartering AB in Gothenburg. During these visits, the author has tried to find out how these companies manage data exchange and what kinds of IT problems they face. The author has used a questionnaire to find out this information (Appendix A).

1.4 Limitations of the dissertation

The author has given various examples of XML documents in this dissertation. These XML documents are prepared by the author and they are well-formed and valid. However, since the author does not have experience in writing software in the Internet programming languages like Java or Visual basic, it has not been possible to develop the software to test the actual transmission of XML documents.

As this dissertation had to be completed within four months as required by the World Maritime University (WMU), the author could not allocate sufficient time to develop the required software. Any technical mistakes found in this dissertation may be due to the misunderstanding of the author.

1.5 Intended readers

This dissertation is prepared as a part of academic study. Readers with general experience in using a Web browser and e-mail should be able to understand this dissertation. Readers do not need to be computer programmers or IT experts to understand the relevance, use and applicability of XML. The author has tried to avoid technical terms and, wherever possible, when such terms have been necessary they are explained.

1.6 Outline of chapters

As the main objective of this dissertation is to help shipping companies to understand the implementation of XML, the author has assumed that readers do not have the previous knowledge of XML. **Chapter 2** deals with the detailed specification required for preparing an XML document and DTD (Document Type Definition). After reading this chapter, readers should be able to prepare an XML document and understand an XML document. A brief history of XML and a comparison of XML with HTML are also included so that readers know the reasons for developing the new markup language, XML. Since XML consists of many technologies like XML Schema, XLink, XPointer, XSL and DOM, a brief explanation of these technologies is also included in this chapter.

Chapter 3 specifically explains the difference between EDI and XML so that the present EDI users can understand the importance of XML in the e-business.

Shipping companies will appreciate the importance of XML, if process of implementations of XML and its benefits are proven. **Chapter 4** mentions the kind of application areas in which XML can be used by shipping companies. The process of implementation of XML based systems and the resources required for implementations are included in this chapter. At the end of this chapter, benefits and limitations of implementing XML based systems are explained.

Chapter 5 includes the conclusions based on the previous three chapters. It also includes recommendations based on the understanding of the author of the XML technologies and the author's own experience in the IT field.

CHAPTER 2

EXTENSIVE MARKUP LANGUAGE (XML) AND ITS SPECIFICATIONS

2.1 Introduction

Extensive Markup Language (XML) is a file format to store and transfer the information from computer-to-computer over the Internet. The format of XML is universal to transfer data on the web. XML allows easy description and delivery of rich, structured data from any application in a standard, consistent way. It is not a software program but it is a computer language for describing information. The word markup in XML means the extra information required to control the layout and appearance of the words themselves. The vessel information, for instance can be described in XML as shown in Figure 2.1.

```
<?XML version 1.0>
<Vessel-Information>
  <VesselCode Age="10"> ABC123 </VesselCode>
  <VesselName> Himalaya </VesselName>
  <DWT Unit="Tons"> 50000 </DWT>
</Vessel-Information>
```

Figure 2.1: Example of Vessel information in XML document

XML describes the information in simple and easy to understand computer language and it allows the users to define their own elements and attributes within an XML document. Thus, XML gives the users freedom to represent the information as per

their choice rather than to depend on some computer software. XML provides a standard method of exchanging documents and data to various output formats (see Figure 2.2). XML documents can be viewed on the present web browsers like Internet Explorer 5.5 or Netscape Navigator 6.2 similar to HTML (Hyper Text Markup Language) Web pages (see Figure 2.3 and Figure 2.4).

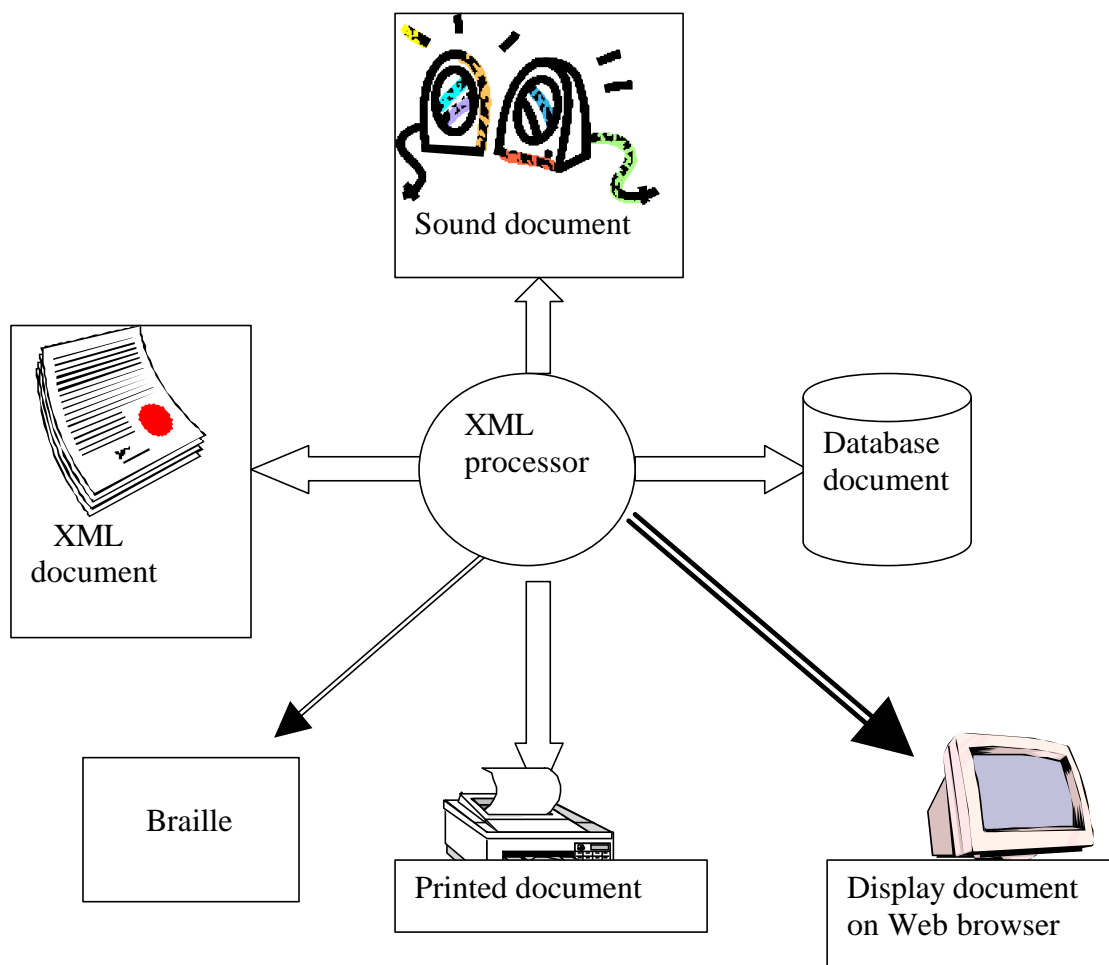


Figure 2.2: XML for different outputs

Source adapted: Tittel & Boumphrey, 2000, p.21

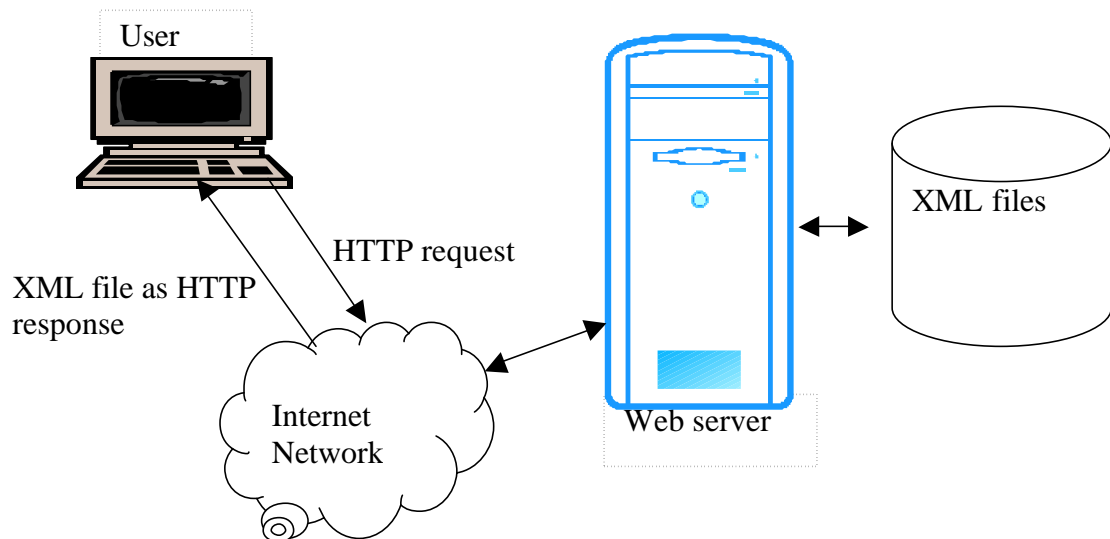


Figure 2.3: Using HTTP to serve XML documents from web server

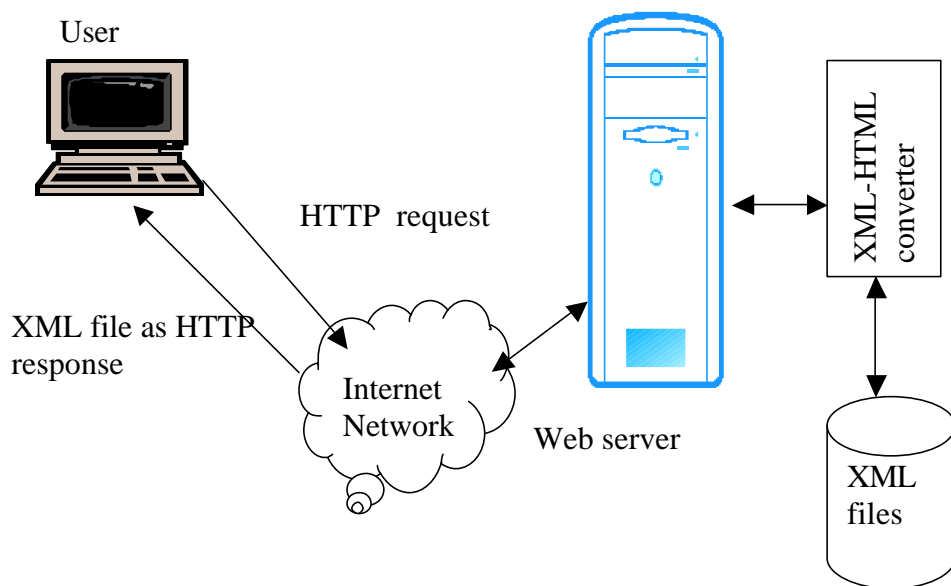


Figure 2.4: Using XML on the server but sending HTML to the browser

Source: Laurent, 1999, pp.28-31

This chapter explains:

- A brief history of XML (section 2.2).
- A comparison between HTML and XML since XML is developed to overcome certain deficiencies found in HTML. Moreover, HTML and XML are developed from the common parent language, SGML (Standard Generalized Markup Language) (section 2.3).
- Specifications of XML to prepare valid XML documents. Further, other developments in XML are also briefly mentioned (section 2.4).

2.2 History of XML

XML was developed by the World Wide Web Consortium (W3C) (www.w3.org/XML), which includes researchers and experts from the computer industry and the publishing industry. The W3C XML working group designed this language from the Standard Generalized Markup Language (SGML) to the web (Tittel & Boumphrey, 2000, p.12). SGML was created to provide a means of formatting and maintaining legal documents, printed books, reports and other documents within IBM. Although SGML is extremely powerful, it is very complex and a significant amount of software overhead is required to process it. In 1989, Tim Berners-Lee and Anders Berglund, two researchers at the European Laboratory for Particle Physics (CERN), Switzerland, created a tag-based language for marking up technical documents to be shared over the Internet. This language was eventually expanded into a simplified application called HTML, which has become the standard information format for the Web. HTML proved to be extremely successful and everyone rushed to get online over the Internet. Some interactive technologies like Java were developed to improve the features of the Web. More features were added to HTML, which is a good thing but it was being stretched beyond its real function. However, members of the W3C realized the need for a much more extensible alternative to HTML to meet the future demands of the Web. In 1996, the W3C set out to find a way to introduce the power and flexibility of SGML into the Web domain with three significant benefits – extensibility, structure, and validation. The

specification of a new markup language called XML 1.0 was released in February 1998 (Morrison, 2000, p.10).

2.3 Comparison of HTML and XML

Hyper Text Markup Language (HTML) is a widely used nonproprietary modern markup language of the Internet. It is mainly used for describing the information on the website, which is supported by the Web browsers like Internet Explorer and Netscape Navigator. HTML is extremely successful and popular due to its simplicity, navigability and freedom. Although HTML is one of the best Internet languages, there are certain limitations of HTML, which are explained below:

- **HTML lacks syntactic checking**

The HTML code is not validated by the Web browsers, which are designed to accept almost anything that looks even slightly like HTML. This runs the risk that the display will look nothing like what is expected. Although validating tools are readily available for checking the syntax of HTML documents, web developers try to bypass them. XML documents are validated by XML parsers for rules and syntax.

- **HTML lacks structure**

Web browsers do not check the structure of the HTML code and accepts the information in any order. XML documents are well structured and follow a well defined structure, otherwise the XML Web browser will display errors. It is difficult to use HTML language for expressing the text structure found in specialized areas like medical records, electronic commerce, academic bibliographies, mathematics, chemistry, and transportation.

- **HTML is not content-aware**

Another problem of HTML documents is the searching for specific information. Web search engines will simply retrieve too many documents and it is difficult to get the right information (Pfaffenberger, 1999, p.45). As the XML is content based and is

well aware of the content of the XML document, it is easier to search the information on a specific content.

- **HTML is not international**

The HTML code does not identify the language tag, which is identified in an XML document by the encoding and language tags.

- **HTML is not suitable for data interchange**

HTML tags do not identify the information within the document but XML can identify contents of an XML document. Hence, HTML cannot be used for data interchange. For example, to describe vessel details consisting of

1. Vessel code: A1234
2. Vessel name: Himalaya
3. Port of arrival: Malmo
4. Arrival date: 2002/06/11
5. Departure date: 2002/06/13

In HTML the above information will be described as shown in Figure 2.5.

```
<html>
<p> A1234</p>
<p> Himalaya</p>
<p> Malmo</p>
<p>2002/06/11</p>
<p> 2002/06/13</p>
</html>
```

Figure 2.5: HTML code to display vessel information

The tags shown in HTML code (see Figure 2.5) are fixed and a computer cannot process them. XML can display the vessel information (Figure 2.6), which is easy to understand, and a computer can directly process it.

```
<?xml version="1.0"?>
<Ves_detail>
  <ves_code>A1234</ves_code>
  <Ves_name>Himalaya</Ves_name >
  <Arrival_port> Malmo</Arrival_port>
  <Arrival_date Date_format="YYYYMMDD">2002/06/11</Arrival_date>
  <Departure_date>2002/06/13</Departure_date>
</Ves_detail>
```

Figure 2.6: XML code to display vessel information

- **HTML is not object-oriented**

Modern programming languages support object-oriented techniques, which are not supported by HTML. XML supports object-oriented features.

- **HTML lacks a robust linking mechanism**

HTML links are one-to-one and hard-coded in the source HTML files. If the location of the target link changes, HTML files have to be updated in all web pages wherever the link is incorporated. XML has the robust linking mechanism.

- **HTML is not reusable**

HTML codes are specifically tailored to their place in the web of associated pages and the HTML code cannot be reused by another web page. XML allows re-use of its code. HTML browsers like Internet Explorer or Netscape Navigator have hard-coded the features of HTML. Every time a new feature is added to the Web browser, it has to be upgraded to support the new feature otherwise the web browser will either show errors or ignore the new feature.

- **HTML is not extensible**

HTML codes are not extensible since the entire code is hard coded in the web browser. In XML, the user can define elements and tags (North & Hermas, 2002, p.9).

Though XML is better than HTML in various features as mentioned in this section, HTML will continue to be the most popular Markup Language to disseminate information on the web. In the web community, HTML is treated as another application of XML. Hence, XML does not replace HTML rather it is a complementary format.

2.4 Specification of XML

The specification of XML (version 1.0) was released in February 1998 by World Wide Web Consortium (W3C). The XML specifications (<http://www.w3.org/XML>) include a number of documents to create the XML based application. It can be divided in four parts for the understanding as follows:

1. Preparation of XML documents
2. XML document validation
3. XML document presentation
4. Related standards and technologies

Out of these four parts, the preparation of XML documents and their validation are the most important from the XML users' point of view. W3C has designed the XML specification for non-programmers and therefore a developer of XML document does not require any programming background. With this consideration, the rules, syntax and terminology used in preparing an XML document are explained. The explanation of XML document presentation and the related standards and technologies are kept minimal since software developers can handle them. However, the knowledge of

these developments can help the users in exploiting the XML features while preparing XML documents. Since the W3C has tried to develop XML technologies from non-programmers' point of view, these can also be handled by XML users. The detailed specification of XML and its related development are available on W3C's website, <http://www.w3.org/XML>. This website also contains the latest developments in XML

2.4.1 Preparation of XML documents

Extensible Markup Language is the core of the XML specification, which defines the format and rules for writing XML documents. An XML document consists of elements to describe the document and attributes of elements. Each element is a piece of data in an XML document and a group of elements forms an XML document. For example, vessel information can be described as shown in Figure 2.7.

Vessel owner: The Shipping Corporation of India Ltd
Vessel Name: Taj Mahal
Vessel Code: TM786
Type of vessel: CONTAINER
Built Date : 2002-01-26
Tonnage : DWT, GRT, NRT: 65000, 60000, 55000
Container capacity (TEU): 5000

Figure 2.7: Example of Vessel Information

Now, the above information can be shown in a logical structure in the form of a tree diagram (Figure 2.8).

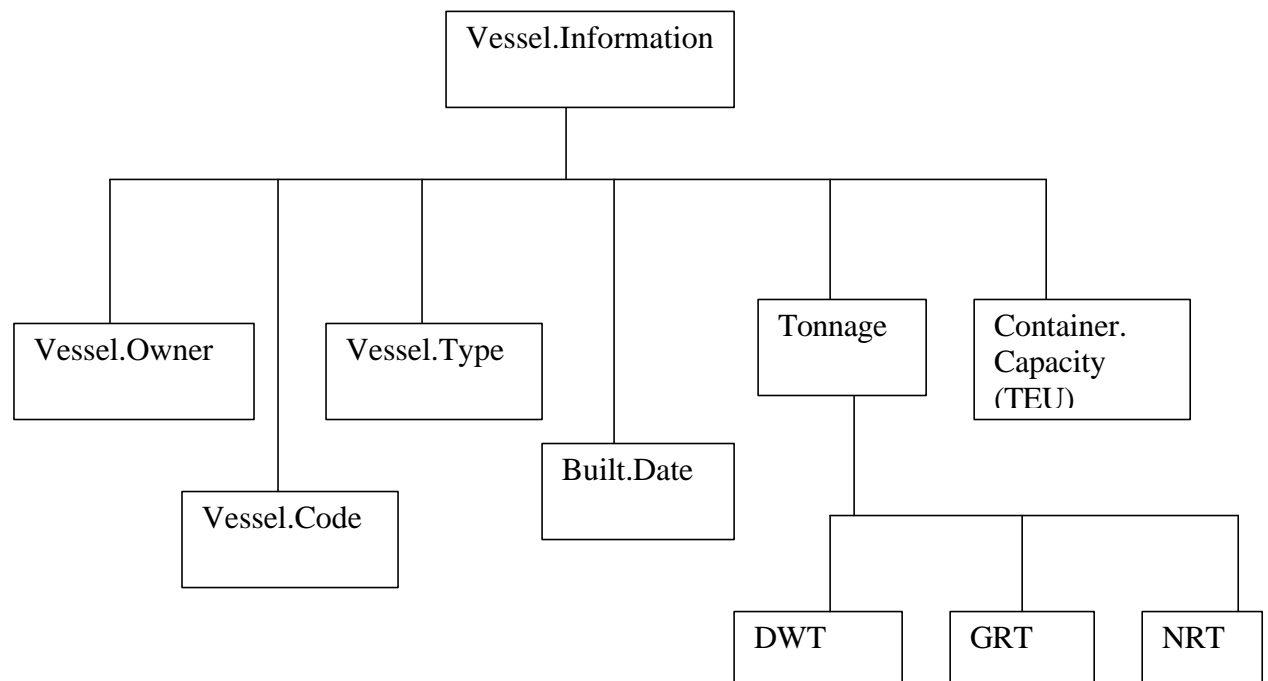


Figure 2.8: Tree diagram of vessel information

Each box shown above is an element for the XML document. The top most element i.e. Vessel.Information, is the root element since all the remaining elements are subordinate to or contained within the root element. The remaining elements i.e. Vessel.Owner, Vessel.Code, Vessel.Type, Built.Date, Tonnage, Container. Capacity, DWT, GRT and NRT, are the child elements of the root element. Further, the element Tonnage is the parent of the elements DWT, GRT and NRT. In XML terminology, elements on the same level of hierarchy are siblings of each other. In an XML document, all the elements are represented in a tree structure and hence, it is better first to prepare a tree diagram to describe the information. The root element is the most important element and each XML document should have only one root element. All the other elements are treated as subordinate elements to the root

element. This is a very important rule to be followed while preparing an XML document. An XML document (see Figure 2.9) can be prepared from the tree diagram of vessel information. The display of this XML document on an XML Web Browser like Amaya (<http://www.w3.org/Amaya/>) is shown in Figure 2.10. The Web browser displays only the content of the XML document without the XML tags.

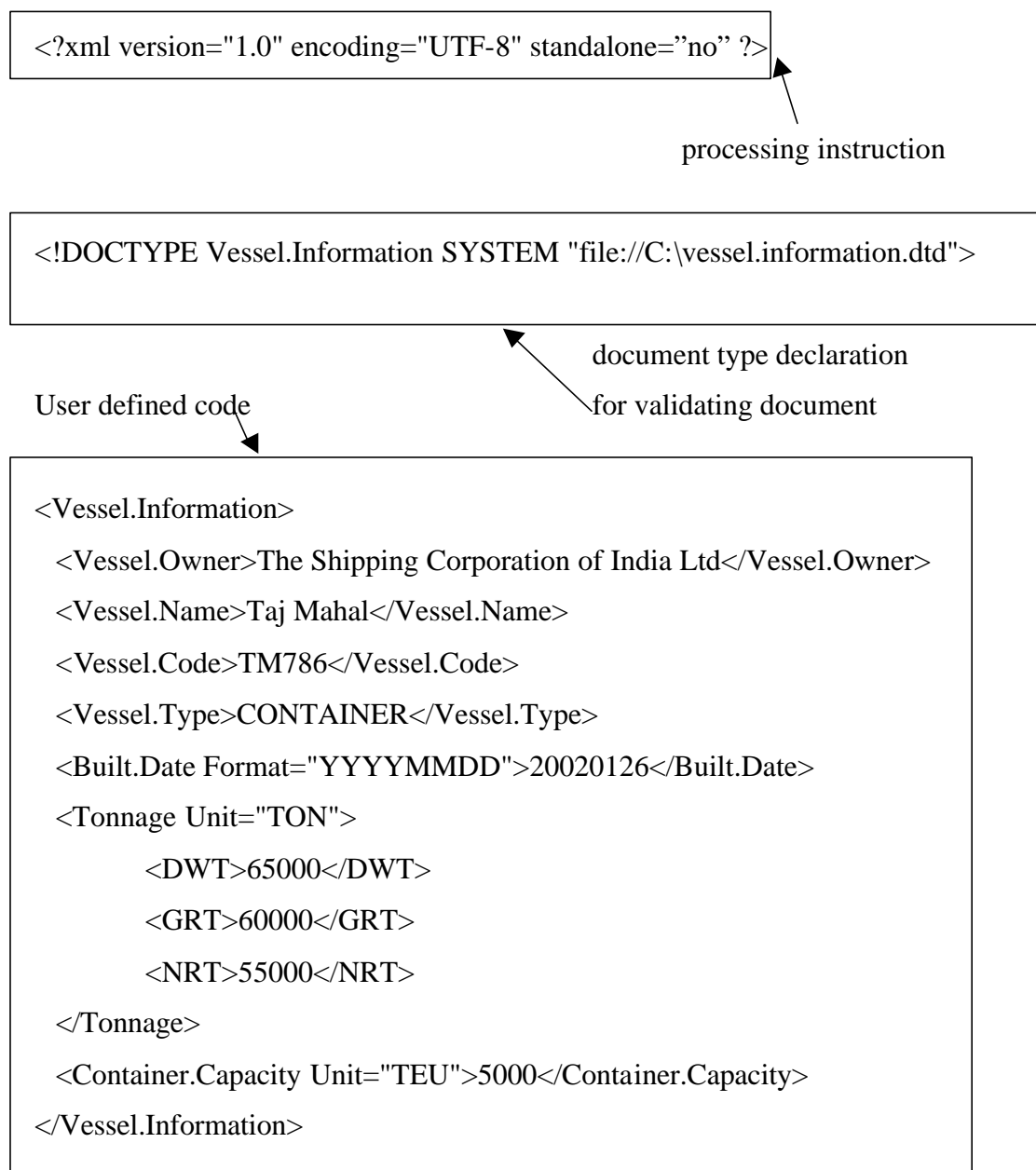


Figure 2.9: Example of XML document for vessel information

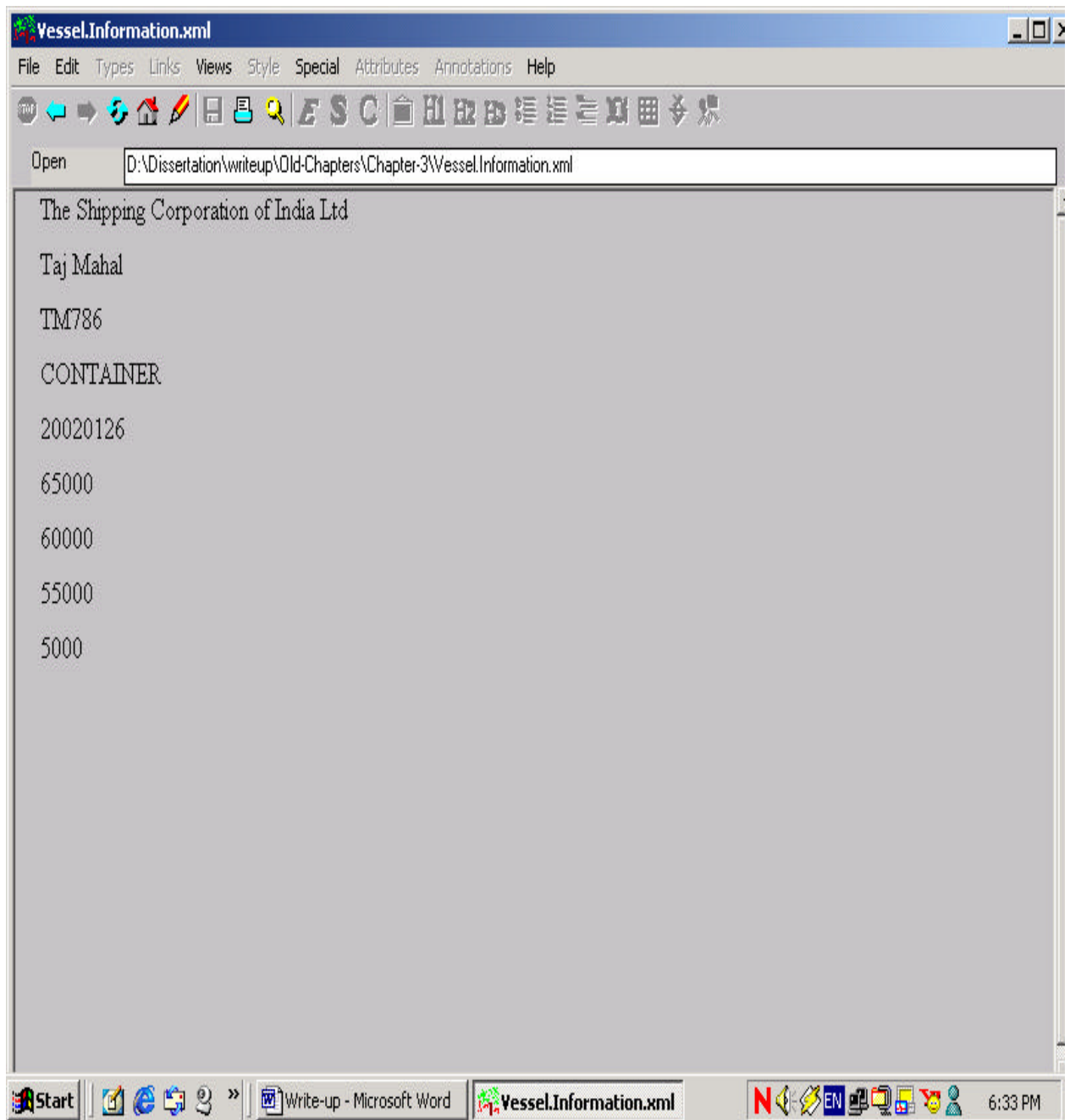


Figure 2.10: Display of XML document on XML Web browser, Amaya

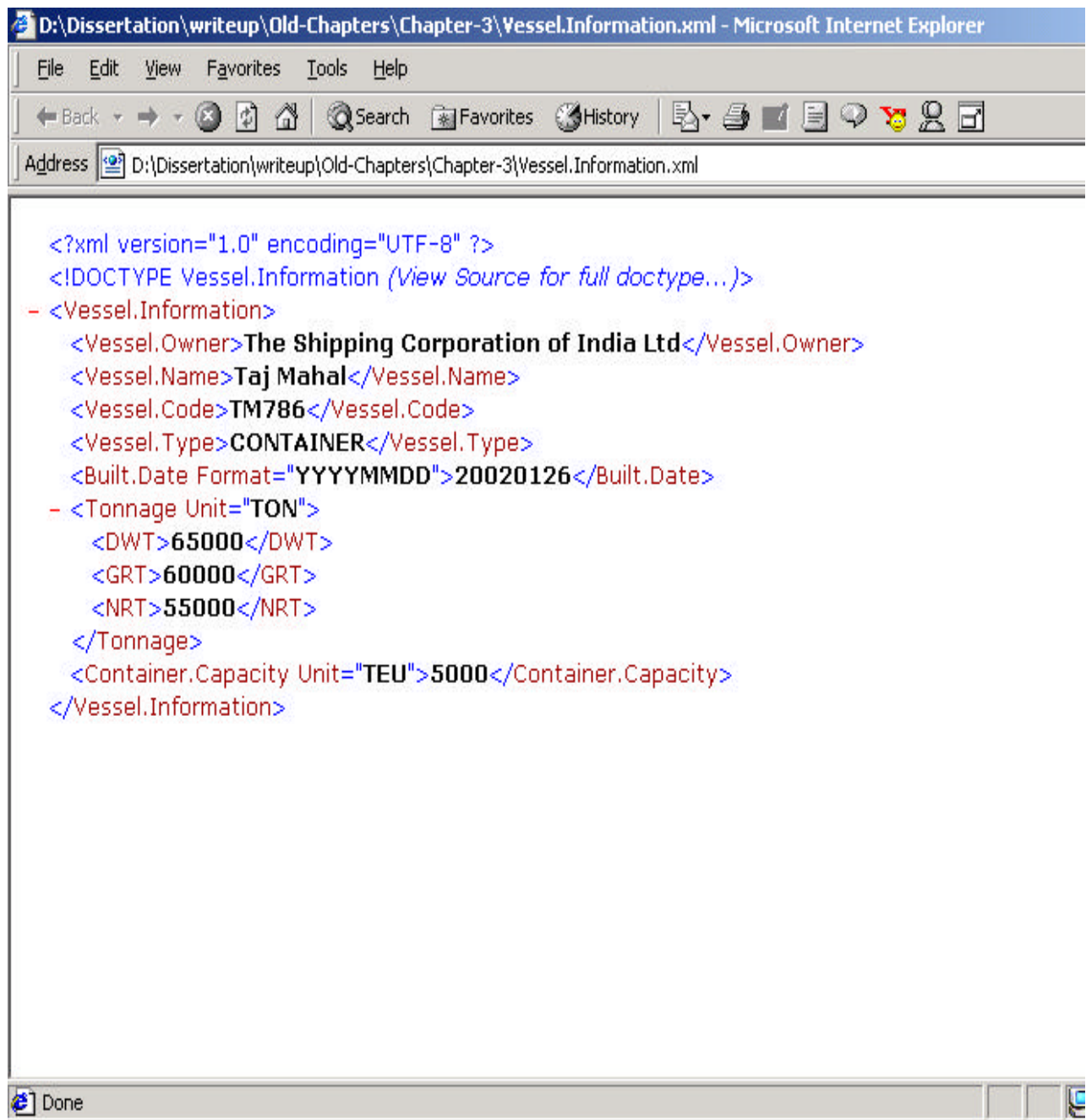


Figure 2.11: Display of XML document on HTML Web Browser, IE5.5

An XML document can also be displayed on an HTML web browser like Internet Explorer 5.5 (see Figure 2.11). HTML Web Browser displays the entire XML document with XML tags and content.

It is easier to create the XML document by an XML editor by listing all the elements and their hierarchy.

2.4.1.1 Authoring tools for preparing XML document

Some of the XML document authoring tools are as follows:

1. Epic Editor (http://www.arbortext.com/html/epic_editor_overview.html)
2. FrameMaker (<http://www.adobe.com/>)
3. Xmetal (SoftQuad, 2002)
4. XML Pro (<http://www.vervet.com/>)
5. XML Spy (<http://www.xmlspy.com/>)
6. Stylus Studio (<http://www.exln.com/products/stylusstudio/>)

(Tittel & Boumphrey, 2000, p.35).

2.4.1.2 Explanation of XML document

- The first box¹ (Figure 2.9) is the processing instruction to the XML processor², which will handle an XML document. It is an XML declaration, which is optional (but highly recommended). It can be up to five parts as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
```

1. `<?xml` : XML declaration begins.
2. `version="1.0"` : version of XML in use. The current version is 1.0
3. `encoding="UTF-8"` : specifies the character encoding in the XML document. UTF-8 means 8-bit ASCII (American Standard Code Information Interchange) characters. The ISO standard allows definition of various character sets. By default, ISO 10646 UTF-8 encoding is assumed

¹ Ignore the box as it is drawn for clarity and explanation only.

² The XML processor is a parser, working on behalf of an application. Its task is to extract the information from an XML document and deliver it to the application. The XML processor should return errors or warnings if the XML document does not follow proper XML syntax.

in the XML document. If the user desires some other encoding, it must be declared in the encoding statement at the beginning.

4. standalone="no" : specifies that external markup declaration may exist. By default, the value is "no".
 5. ?> is end of the XML declaration.
- The second box is the document type declaration, which corresponds to the location of the DTD (Document Type Definition) of the XML document. It tells the XML processor where the DTD is located (see section 2.4.2.1 for DTD). The document type declaration is the XML statement that declares what type of XML document follows and identifies the document type definition (DTD), which contains the description of the allowed structure of this type of document.

`<!DOCTYPE Vessel.Information SYSTEM "file://C:\vessel.information.dtd" >`

The document type declaration statement can be divided in five parts:

1. <!DOCTYPE : begins the document type declaration
2. Vessel.Information : name of the root element. In this case, vessel.information is the root element of the XML document.
3. SYSTEM means the DTD is located on the user's PC.
4. "file://C:\vessel.information.dtd " is the physical file name of the DTD.
5. > indicates end of the document type declaration.

The general syntax of the document type declaration statement is:

`<!DOCTYPE root-element-name external-pointer [internal-subset] >`

In the above syntax, the external-pointer can be SYSTEM or PUBLIC. SYSTEM means the DTD is located on the user's system and PUBLIC means the DTD is

located on the public network, which can be located by identifying the URI³ (Universal Resource Identifier). The internal-subset means the DTD is defined within the XML document. The user can define the DTD as an external file or within an XML document and can use XML Schema in lieu of DTD. In that case, the user has to use XML Schema declaration (refer section 2.4.2.2 for details).

- The third box is the user-defined code. The explanation of the user-defined code is briefly given in Table 2.1.

Table 2.1: Explanation of user-defined code

Line number	XML statement	Explanation
1	<Vessel.Information>	Root element – start tag
2	<Vessel.Owner>The Shipping Corporation of India Ltd</Vessel.Owner>	Child element- Vessel.Owner with start tag & end tag
3	<Vessel.Name>Taj Mahal</Vessel.Name>	Child element- Vessel.Name with start tag & end tag
4	<Vessel.Code>TM786</Vessel.Code>	Child element- Vessel.Code with start tag & end tag
5	<Vessel.Type>CONTAINER</Vessel.Type>	Child element- Vessel.Type with start tag & end tag
6	<Built.Date Format="YYYYMMDD">20020126</Built.Date>	Child element- Built.Date with start tag & end tag. Attribute 'Format' is defined for Built.Date
7	<Tonnage Unit="TON">	Child element- Tonnage

³ URI (Universal Resource Identifier) is used to retrieve a file from a location. While browsing a Web page or downloading a file from an FTP site, the URL (Uniform Resource Locator) is displayed. It is one of the forms of URI. For a majority of uses in XML specification, URI and URL are virtually interchangeable terms (North & Hermas, 2002, p.103).

		with start tag. It is a group element. Attribute 'Unit' is defined.
8	<DWT>65000</DWT>	Child element- DWT with start tag & end tag. Defined within group element, Tonnage.
9	<GRT>60000</GRT>	Child element- GRT with start tag & end tag. Defined within group element, Tonnage.
10	<NRT>55000</NRT>	Child element- NRT with start tag & end tag. Defined within group element, Tonnage.
11	</Tonnage>	Child element- Tonnage with end tag.
12	<Container.Capacity Unit="TEU">5000</Container.Capacity>	Child element- Container.Capacity with start tag and end tag. Attribute 'Unit' is defined.
13	</Vessel.Information>	Root element – end tag

In the user-defined code, the users can define the name of an element of their choice and need not select a predefined name of an element. All names of elements in XML standard are case sensitive. The element name should follow rules as follows:

- A name should consist of at least one character: a to z, or A to Z
- If the name consists of more than one character, it may start with an underscore (_) or a colon (:). The name should not consist of <_>.

- The initial letter can be followed by one or more letters, digits, hyphens, underscores, and full stops.
- Spaces and tabs are not allowed in element names.

(North & Hermas, 2002, p.42).

2.4.1.3 Element tags and parts

An XML document consists of elements and each element consists of tags and content (Table 2.2).

Table 2.2: The parts of an Element Tag

Symbol	Description
<	Start tag open delimiter
</	End tag open delimiter
foo	Element name
>	Tag close delimiter
/>	Empty tag close delimiter

Source: North & Hermas, 2002, p.38

As an XML document is concerned with element markups, each element markup consists of three parts: a start tag, the contents and the end tag (see Table 2.3).

Table 2.3: The parts of an Element markup

Symbol	Name	Description
<foo>	Start tag	At the start of an element, the opening tag
Text	Content	In the middle of an element, its content
</foo>	End tag	At the end of an element; the closing tag

Source: North & Hermas, 2002, p.38

When an element has no content, empty-element tags are used. There is a special empty tag close delimiter, />, as <empty-element/>. This tells the XML processor that it should not be looking for a corresponding end-tag.

2.4.1.4 Attributes

An element can include one or more attributes which give further information about the element they limit. In the example (Figure 2.9), elements such as “Built.Date” has an attribute “Format”, “Tonnage” has an attribute “Unit”, and “Container.Capacity” has an attribute “Unit”. The rest of the elements do not have attributes. The user can define attribute names as per the naming rules applicable for element-name and can specify an attribute only in the element start tag. The syntax for specifying an attribute is

```
<element-name attribute-name="attribute-value">
```

The technical description of the attribute specification is given in Table 2.4.

Table 2.4: Specifying an Attribute

Symbol	Description
<	Start tag open delimiter
Element-name	Element name
Property	Attribute name
=	Value indicator
“ ‘	Literal string delimiter (single quote or double quote)
Value	Value of the attribute
>	Start tag close delimiter

Source: North & Hermas, 2002, p.39

As explained in this section, preparation of an XML document consists of

- XML declaration statement
- Document type declaration statement to identify DTD or XML Schema definition
- Description of root element
- Description of child elements and their attributes.

2.4.2 XML document validation

Once an XML document is prepared, it can be validated by using a DTD or an XML Schema, which defines the validation procedure of the elements and attributes. It is not mandatory to write a DTD or an XML Schema but it is recommended.

2.4.2.1 Document type definition (DTD)

DTD describes the validation procedure for an XML document. It can be written within the XML document or an external file. The DTD of an XML document (see Figure 2.9) is shown in Figure 2.12.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!ELEMENT Vessel.Owner ANY>
<!ELEMENT Vessel.Name ANY>
<!ELEMENT Vessel.Code ANY>
<!ELEMENT Vessel.Type (#PCDATA)>
<!ELEMENT Built.Date (#PCDATA)>
<!ATTLIST Built.Date Format CDATA #REQUIRED>
<!ELEMENT DWT (#PCDATA)>
<!ELEMENT GRT (#PCDATA)>
<!ELEMENT NRT (#PCDATA)>
<!ELEMENT Tonnage (DWT,GRT,NRT)>
<!ATTLIST Tonnage Unit CDATA #REQUIRED>
<!ELEMENT Container.Capacity ANY>
<!ATTLIST Container.Capacity Unit CDATA #REQUIRED>
<!ELEMENT Vessel.Information
(Vessel.Owner,Vessel.Name,Vessel.Code,Vessel.Type,Built.Date,Tonnage,C
ontainer.Capacity)>
```

Figure 2.12: DTD of Vessel Information (“vessel.information.dtd”)

Explanation of DTD

- The first line in the DTD (Figure 2.12) tells the XML processor which version of XML to use, which character set is used to write the XML document and whether any external mark-up is used within the DTD (refer section 2.4.1.2 for more explanation).

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
```

- Second line onwards in the DTD describe the elements and their attributes.

Element declaration in DTD

Each element of an XML document must be declared in the DTD and only one element must be declared at a time. The element in DTD is declared in the following form:

```
<!ELEMENT element-name content-model>
```

For example, Vessel.Owner in the DTD is declared as

```
<!ELEMENT Vessel.Owner ANY>
```

Other elements such as Tonnage and NRT are declared as

```
<!ELEMENT Tonnage (DWT,GRT,NRT)>
```

```
<!ELEMENT NRT (#PCDATA)>
```

The above syntax can be divided in four parts:

1. `<!ELEMENT` : this is the start tag to declare an element in the DTD. It is an XML keyword.
2. `element-name` : the name of the element described in the XML document. For example, Vessel.Owner is the name of the element.

3. content-model : The content of an element can be EMPTY, ANY or describes the sequence and repetition of child elements within the element

This is described in detail in Table 2.5.

4. > : symbol to describe the end tag of declaration.

Table 2.5: Type of contents found in Elements

Content type	Example	Meaning
ANY	<!ELEMENT Name ANY>	Allows any type of element content, either data or element markup
EMPTY	<!ELEMENT Name EMPTY>	Element must contain no data
Mixed content	<!ELEMENT Name (#PCDATA ChildName)> e.g. Vessel.Type	An element contains character data or a combination of sub-elements and character data
Element content	<!ELEMENT Name (Child1, Child2)> e.g. Vessel.Name,Vessel.Code are children of Vessel.Information	An element contains only sub-elements or children

Source: Tittel & Boumphrey, 2000, p.68

Empty elements

Empty elements have no content (they are forbidden to have any) and are marked up as `<empty.element/>`. They are used for things like external graphics or external applications.

Unrestricted elements

The opposite of an empty element is an unrestricted element, which can contain any element. An unrestricted element's content is declared as:

```
<!ELEMENT element.name ANY>
```

Mixed elements

Empty elements and unrestricted elements are special types of elements which either allow nothing or everything. XML's element declaration precisely specifies which other elements are allowed inside an element, how many times they can occur and in what order or sequence.

When only text is allowed inside an element, it is identified by the keyword `PCDATA` (parseable character data) in the content. The element is declared as:

```
<!ELEMENT element.name (#PCDATA)>
```

For example, the element `Vessel.Type` is declared as

```
<!ELEMENT Vessel.Type (#PCDATA)>
```

An element may consists of other declared elements which can be shown as:

```
<!ELEMENT element.name (child1, child2, child3)>
```

For example, the element Tonnage is declared as

```
<!ELEMENT Tonnage (DWT,GRT,NRT)>
```

It means Tonnage consist of all the elements DWT, GRT and NRT.

An element may consist of any one of the other elements. It can be shown as,

```
<!ELEMENT element.name (child1 | child2 | child3)>
```

For example, element Tonnage may be declared as

```
<!ELEMENT Tonnage (DWT | GRT |NRT)>
```

It means Tonnage consists of either DWT or GRT or NRT.

Element occurrence indicators

By using an occurrence indicator, the user can specify how often an element or group of elements may appear in an element (Table 2.6).

Table 2.6: Occurrence indicators in DTD

Symbol	Example	Meaning
,	<!ELEMENT Vessel.Information (Vessel.Owner,Vessel.Name,Vessel.Code)>	All listed child elements must be used in the order shown
	<!ELEMENT Vessel.Information (#PCDATA Vessel.Name)	Either one or the other may be used
	<!ELEMENT Vessel.Information (Vessel.Name)	This non-symbol indicates a required occurrence exactly once
+	<!ELEMENT Vessel.Information (Vessel.Name +)	This child element must be used at least once
*	<!ELEMENT Vessel.Information (Vessel.Name *)	This child element may be used any number of times or not at all
?	<!ELEMENT Vessel.Information (Vessel.Name?)	This child element must be used exactly once or not at all

Source: Tittel & Boumphrey, 2000, p.71

It may be observed that occurrence indicators specify how often an element or group of elements can occur- not at all, once, or an unlimited number of times. It does not allow the users to specify an exact number of occurrences only 0, 1, or infinite. XML's content form does not allow the users to choose elements on a selective basis, for example, "if element A and B, choose between C and D; otherwise choose between E and F." (North & Hermas, 2002, p.63). These limitations of DTDs are overcome in the XML Schema definition, which was released in May 2001 by W3C.

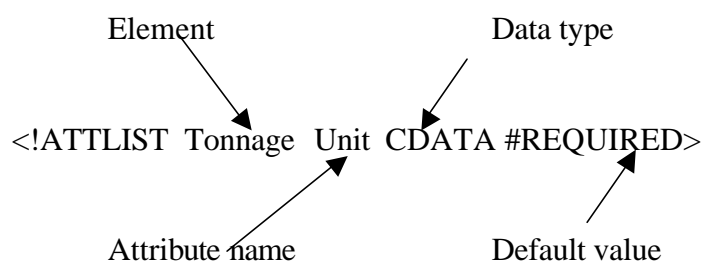
Attribute declarations

Each element may have one or more attributes which are declared in the DTD in the following format:

`<!ATTLIST element-name attribute-name data-type default-value>`

For example, attribute of element, Tonnage is declared as,

`<!ATTLIST Tonnage Unit CDATA #REQUIRED>`



The attribute declaration syntax is explained as follows:

1. `<!ATTLIST` - start tag of attribute declaration and it is an XML keyword.
2. `element-name` – name of element for which attribute to be declared.
3. `attribute-name` - the name of attribute corresponding to the element-name.
4. `data-type`
5. `default-value`
6. `>` - end tag of attribute declaration.

Attribute data types

There are three data types of attributes:

- **CDATA** or character data, a string attribute, whose value consists of any amount of character data. This is the default data type. For example,
`<!ATTLIST Tonnage Unit CDATA #REQUIRED>`

- An enumerated attribute, whose value is taken from a list of declared possible values, separated with vertical bars (|). For example, Unit can take two values - TON or CUBIC-METER.

`<!ATTLIST Tonnage Unit (TON | CUBIC-METER) #REQUIRED>`

- A tokenized attribute, whose value consists of one or more tokens that are significant to XML. These are classified according to what their value, or values can be:

1. ID - unique identifier of the attribute
2. IDREF , IDREFS – identifier reference
3. ENTITY – allows the attribute to take a declared value
4. ENTITIES – allows multiple values
5. NMTOKEN – name tokens restrict how strings are used in attributes
6. NMTOKENS – multiple name tokens
7. NOTATION – name of a particular file type or non-text data

Attribute default values

It specifies the action the XML processor should take if the value of an attribute is missing in the XML document. There are three possible default values:

1. #REQUIRED means that the attribute is required and should have been there. If it missing, it makes the document invalid. For example, value of attribute Unit is required for the element, Tonnage

`<!ATTLIST Tonnage Unit CDATA #REQUIRED>`

2. #IMPLIED means that the XML processor must tell the application that no value was specified. It is then up to the application to decide what it is going to do. For

example, if the value of attribute Unit is missing, the application will decide the value,

```
<!ATTLIST Tonnage Unit CDATA #IMPLIED>
```

3. #FIXED means that if the default value is preceded by the keyword #FIXED, any value that is specified must match the default value or the document will be invalid. For example, value of attribute, Unit should be TON otherwise the document is invalid.

```
<!ATTLIST Tonnage Unit CDATA #FIXED "TON">
```

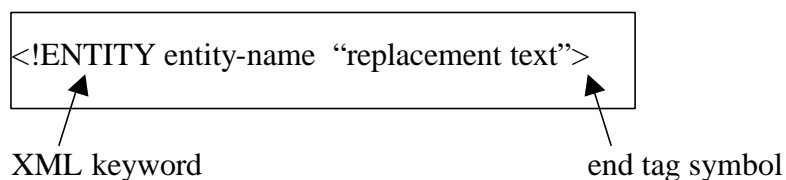
Entity references

An entity is generally an external object such as a graphic file or a block of text, which is referred to inside an XML document. For example, a company's name such as "The Shipping Corporation of India" can be referred to by the company inside an XML document. It saves the time of typing the company name repetitively. This can be declared in the DTD as,

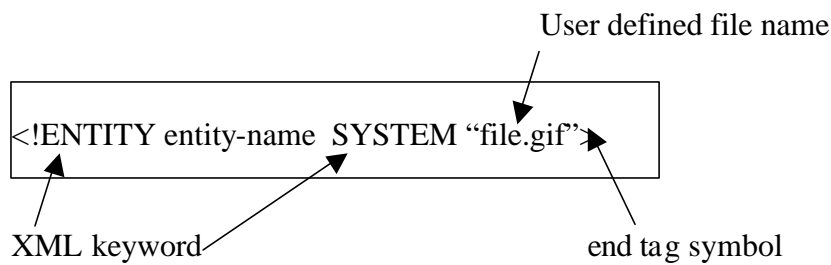
```
<!ENTITY company "The Shipping Corporation of India">
```

Now, every time the string "&company;" appears in an XML code, the XML processor will automatically replace it with The Shipping Corporation of India.

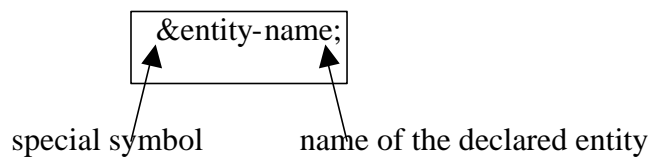
An entity can be declared internally or externally but the entity must be declared in the DTD before it is used. The declaration of an internal entity looks like this:



An external entity is declared as:



An entity is always referred to in an XML document as:



For example, entity company is referred to inside an XML document as:

```
< Vessel.Owner>&company; </Vessel.Owner>
```

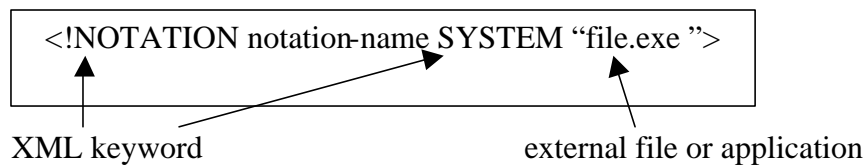
XML uses some special characters, which cannot be directly used in an XML document in the user-defined names or data. However, these characters can be used by declaring predefined entities (Table 2.7).

Table 2.7: Predefined entities

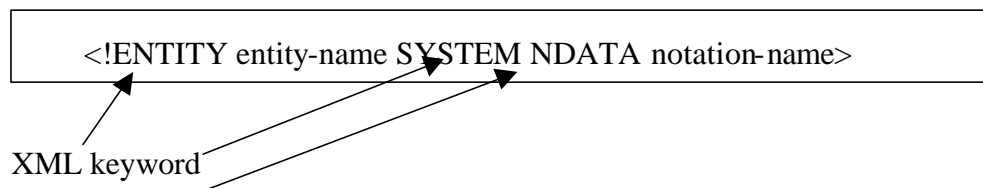
Character	Replacement
&	&
'	&apos
"	"e
<	<
>	>

Notation declaration

Notation declarations allow XML documents to refer to external information such as graphic files or an external application to which a processing instruction is addressed. A notation must be declared in the DTD before it is used. It can be declared as:



The keyword, `SYSTEM` indicates that an entity appears in a URI. It can be `PUBLIC`, which indicates that a standard name of an entity is referenced. After declaration of the notation, it is referred to by name in an entity declaration with `NDATA` (notation data) keyword as:



Well-formed XML document

Elements, attributes and entities are the three primary building blocks of XML documents. By using these three objects, 90% of the requirements of an XML document will be fulfilled. According to the XML standard, a data object is not officially an XML document until it is well formed. An XML document using elements, attributes, and entities is well formed under the following conditions (North & Hermas, 2002, p.68):

- It contains one or more elements.
- It has just one root element that contains all the other elements.
- Each element should have a start tag and end tag matching exactly.

- The names of attributes do not appear more than once in the same element start tag.
- The values of its attributes do not reference external entities, either directly or indirectly.
- The replacement text for any entity referenced in an attribute value does not contain a < character (it can contain <).
- Its entities are declared before they are used.
- None of its entity references contain the name of an unparsed entity.
- Its logical and physical structures are properly nested.
- All elements must have an explicit end tag (for example, <Vessel.Name>TajMahal</ Vessel.Name>)
- All tags must be nested properly (for example,

```
<Tonnage Unit="TON">
    <DWT>65000</DWT>
    <GRT>60000</GRT>
    <NRT>55000</NRT>
```

```
</Tonnage>
```

is correct and the following description is incorrect,

```
<Tonnage Unit="TON">
    <DWT>65000</DWT>
    <GRT>60000</GRT>
    <NRT>55000</Tonnage></NRT>
```

- Empty tags use special XML syntax that open and close the tag in a single expression (for example, <vessel />).
- All attribute values should appear in quotation marks (for example, <Tonnage Unit="TON">).
- All entities in a document must be declared explicitly in the DOCTYPE declaration.
- All tags are case sensitive (for example, <Vessel> is no the same as <vessel>) (Tittel & Boumphrey, 2000, p.59).

2.4.2.2 XML schema

XML Schema is an XML based alternative to DTD for describing the structure of an XML document. The XML Schema language is also referred to as XML Schema Definition (XSD). XML schema is one of the recent developments in XML technology. The current version 1.0 of XML Schema was recommended on 2 May 2001 by W3C (World Wide Web Consortium). The XML Schema working group is working on the next version 1.1 to further improve on the present XML Schema. (<http://www.w3.org/XML/Schema>).

An XML Schema consists of the definitions of:

- elements
- attributes
- child elements
- order of child elements
- number of child elements
- whether an element is empty or can include text
- datatypes of elements and attributes
- default and fixed values of elements and attributes

An XML Schema is better than DTD due to some of the following reasons:

- XML Schema is extensible to future additions
- XML Schema is richer and more useful than DTD
- XML Schema is written in XML
- XML Schema supports 44 datatypes
- XML Schema supports namespaces

(W3Cschoools, 2002)

XML Schema datatypes

One of the greatest strength of XML Schema is the support for 44 datatypes which are useful for:

- describing the permissible document content
- validating the correctness of data
- working with data from database
- defining data facets (restrictions on data)
- defining data patterns (data formats), and
- converting data between different datatypes

Some of these **datatypes** are as follows:

- String: A collection of characters
- Decimal: A number that includes a decimal point and some number of decimal places.
- Date
- Time
- AnyURI: a URI or URL
- Integer: a number without a decimal point
- Boolean: true or false

The complete list of datatypes is available at <http://www.w3.org/TR/xmlschema-2>.

Each datatype has a list of constraints, which can be used to further define the data described within an element or attribute. These constraints or restrictions are useful in validating the data or attribute values. Some of these constraints are:

- Restriction on values such as minimum value, maximum value
- Restriction on a set of values such as list of values
- Restrictions on a series of values such as a range of values
- Restrictions on length of string such as minimum length, maximum length

Structure of XML Schema

The structure of XML Schema consists of

1. XML declaration

```
<?xml version="1.0" encoding="UTF-8?>
```

2. Schema element

```
<xsd:schema xmlns:xsd="namespace">
```

This identifies the document as XML Schema

3. Element declaration

```
<xsd:element name="element-name">
```

This defines the element-name described in the XML document

4. Attribute declaration

```
<xsd:attribute name="attribute-name" type="datatype">
```

This defines the attribute-name (described in the XML document) and its datatype.

Element declarations

An XML Schema document must have elements, which are defined in element declarations. The element declarations are of two categories:

1. Simple type definitions: These are used for elements, which cannot contain any other elements or attributes.
2. Complex type definitions: These are used for elements, which can contain other elements or attributes.

For example, an XML document (Figure 2.13) has four elements viz. VesselInformation, VesselCode, VesselName and DWT.

```

<?xml version="1.0"?>
<VesselInformation>
  <VesselCode ID="786"/>
  <VesselName> Himalaya </VesselName>
  <DWT Unit=Tons> 50000 </DWT>
</VesselInformation>

```

Figure 2.13: XML Document for Vessel Information

VesselName is a simple type definition and it can be declared as:

```

<xsd:element name=" VesselName" type="xsd:string"/>

```

VesselInformation, VesselCode and DWT are complex type definitions since they contain other elements or attributes.

VesselInformation is a complex definition and it can be described as:

```

<xsd:element name=" VesselInformation">
  <xsd:complextype>
    <xsd:sequence>
      <xsd:element ref=" VesselCode">
      <xsd:element ref=" VesselName" >
      <xsd:element ref=" DWT">
    </xsd:sequence>
  </xsd:complextype>
</xsd:element>

```

Content models

A content model defines what type of content an element has - text, other elements, or a combination of the two.

Text: The element can contain only text.

```
<xsd:element name=" VesselName" type="xsd:string"/>
```

Empty: The element cannot contain child elements or text.

```
<xsd:element name="VesselCode">
  <xsd:complexType>
    <xsd:attribute name="ID" type="xsd:integer" use="required"/>
  </xsd:complexType>
</xsd:element>
```

Mixed: The element consists of text and other elements.

Composite element

The element that contains child elements is a composite element which can be of three types:

1. **sequence** indicates child elements must occur in a specified order in the XML document.
2. **choice** indicates that any one of the elements may occur in the XML document.
3. **all** indicates that any or all of the elements may occur in the XML document

Attribute declaration

An attribute declaration consists of name of the attribute and datatype. The general syntax for attribute declaration is:

```
<xsd:attribute name="attribute-name" type="datatype">
```


For example, the attribute of element, DWT is UNIT. It can be declared as:

```
<xsd:element name="DWT">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:integer">
        <xsd:attribute name="Unit" type="xsd:string" use="required"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

An attribute may have default values or fixed values. The default value is automatically assigned to the attribute when no other value is specified. A fixed value is also automatically assigned to the attribute but the user cannot specify another value. Furthermore, all attributes are optional by default but the user can specify that the attribute value is required.

```
<xsd:attribute name="UNIT" type="xsd:string" default="TON"/>
<xsd:attribute name="UNIT" type="xsd:string" fixed="TON"/>
<xsd:attribute name="UNIT" type="xsd:string" use="optional"/>
<xsd:attribute name="UNIT" type="xsd:string" use="required"/>
```

The user can also define attribute groups if more than one element is using the same kind of attributes. In such case, the attribute groups must be declared in the beginning of the XML Schema document.

Example of XML Schema

The XML Schema of the XML document in Figure 2.13 can be written as shown in Figure 2.14.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="VesselCode">
    <xsd:complexType>
      <xsd:attribute name="ID" type="xsd:integer" use="required"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="VesselName" type="xsd:string"/>
  <xsd:element name="DWT">
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base="xsd:integer">
          <xsd:attribute name="Unit" type="xsd:string" use="required"/>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="VesselInformation">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="VesselCode"/>
        <xsd:element ref="VesselName"/>
        <xsd:element ref="DWT"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

Figure 2.14: XML Schema for Vessel Information (Vessel.xsd)

Once an XML Schema document is ready, it should be declared in an XML document. Figure 2.15 shows an example of declaring an XML Schema in an XML document.

```
<?XML version="1.0" encoding="UTF-8"?>
<VesselInformation
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file://c:\vessel.xsd">
  <VesselCode ID="786"/>
  <VesselName> Himalaya </VesselName>
  <DWT Unit=Tons> 50000 </DWT>
</VesselInformation>
```

Figure 2.15: XML Schema declaration in XML document

Choice the XML Schema or DTD

Since an XML document can be validated by using either DTD or XML schema, the choice of XML Schema or DTD depends upon the content of the XML document. If the XML document consists of large amount of data and the data should be validated, XML Schema is preferable. DTD is used for XML documents consisting of large blocks of text data rather than other types of data (Tittel, Pitts & Boumphrey, 2002, pp.121-122).

2.4.3 XML document presentation

XML is not intended to describe how content should be formatted but it is designed to describe the roles that content plays in a document. To display an XML document in a particular format on a computer screen or any output device, cascading style sheet (CSS) language and extensible style language (XSL) have been developed by W3C. XLink, XPath, and XPointer are further added to enhance the capabilities for searching and displaying an XML document.

2.4.3.1 Cascading Style Sheets (CSS)

Cascading Style Sheet (CSS) is basically designed to provide a means of styling an XML document on the Web. CSS defines style constructs like fonts, colors, and positioning of data. It is required to isolate the presentation properties from content (Tittel & Boumphrey, 2000, p.145). A sample of the CSS code created for vessel information is shown in Figure 2.16. More information on style sheets is available at website <http://www.w3.org/Style/CSS/>.

```
Vessel.Owner {  
  display: list-item;  
  color: black;  
  font-family: Helvetica, sans-serif;  
  font-size: 24pt;  
  font-weight: bold;  
}
```

```
Vessel.Name {  
  display: inline;  
  color: blue;  
  font-family: Courier, monospace;  
  font-size: 14pt;  
  text-indent: 12px;  
}
```

```
Vessel.Code {  
  display: block;  
  color: red;  
  font-family: Western, fantasy;  
  font-size: 12pt;  
  text-indent: 12px;  
}
```

Figure 2.16: CSS code for Vessel Information

2.4.3.2 Extensible Style Language (XSL)

XSL is the style sheet language similar to CSS to display an XML document. However, XSL has more functionality to manipulate XML documents. Two fundamental technologies XSL Transformation (XSLT) and XSL Formatting objects

(XSL FO) are implemented in XSL. XSLT is used to transform the XSL formatting language that is used to apply formatting style to XML documents for display purposes. XSL formatting objects contain all of the functionality of CSS and then more (Morrison, 2000, p.177). More information on XSL is available at website (<http://www.w3.org/Style/XSL/>). XSLT is actively being used to transform XML documents to HTML, because Web browsers lack consistent support for XML while HTML is compatible and widely used (Tittel & Boumphrey, 2000, p.183). (www.w3.org/TR/xslt).

A sample of the XSL code created for vessel information is shown in Figure 2.17.

```
<?xml version='1.0'?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="xml"
encoding="iso-8859-1" />
<xsl:template match="/">
<html>
<head>
<title>
<xsl:value-of select="Vessel.Information" />
</title>

</head>
<body>
<xsl:apply-templates />
</body>
</html>
</xsl:template>
<xsl:template match="Vessel.Owner">
<h1>
<xsl:apply-templates />
</h1>
</xsl:template>
<xsl:template match="Vessel.Name" >
<h2>
<xsl:apply-templates />
</h2>
</xsl:template>

<xsl:template match="Vessel.Code">
<li>
<xsl:apply-templates />
</li>
</xsl:template>
```

Figure 2.17: XSL code for Vessel Information

2.4.3.3 XML Linking Language (XLink)

XLink defines the hyperlink capabilities of XML. The users can create and describe the links between XML documents by using elements of those documents. To add an XLink to an XML document, XLink namespaces are used in the XML document and an XLink declaration is used in the DTD or XML Schema (<http://www.w3.org/XML/Linking>).

For example, a simple XLink is declared in an XML document as:

`<xlink:simple href="http://www.wmu.se" WMU</xlink:simple>`

2.4.3.4 XML Pointer Language (XPointer)

XPointer is used to get inside a document even though the users do not have editing privileges for that document. XPointer provides several mechanisms to create links into specific document locations. A large chunk of that document can be selected by using the XPointer mechanism (<http://www.w3.org/TR/xptr/>).

For example, the second child element i.e.Vessel.Name in the Vessel.Information document can be referred to as

`Vessel.Information.xml#/1/2`

Current browsers do not support XPointer features (Tittel & Boumphrey, 2000, p.243).

2.4.3.5 XML Path Language (XPath)

The XPath specification is used to find a way to and describe a section of an XML document and to point to a certain section of the document (Tittel & Boumphrey, 2000, p.227). XPath is used by XSLT and XPointer (<http://www.w3.org/TR/xpath>).

For example, to select the last child element of Vessel.Information, it can be represented as:

`Child:Vessel.Information[position()=last()]`

2.4.4 Related standards and technologies

There are many standards and technologies developed and being developed around XML standards. Some of them are:

1. Namespaces in XML
2. Document Object Model

2.4.4.1 Namespaces in XML

Namespaces allow using tags and semantics of one type of XML document in another type of XML document without defining and declaring the other document's markup in a DTD. (Tittel & Boumphrey, 2000, p.211). XML namespaces provide a simple method for qualifying element and attribute names used in XML documents by associating them with namespaces identified by URI (Universal Resource Identifier) references (<http://www.w3.org/TR/REC-xml-names/>). Namespaces are very important in preparing an XML Schema document.

For example: An XML declaration in an XML document,

```
<Vessel.Information xmlns:html="http://www.w3.org/html">
```

tells the software that an HTML tag set is used in the XML document.

2.4.4.2 Document Object Model (DOM)

DOM describes the hierarchy of each element defined in the XML document. The DOM is a tool for manipulating data, not a data structure of an XML document. (<http://www.w3.org/DOM/>).

This chapter has covered the details of preparing XML documents and DTDs. It has also included an introduction to other XML technologies like XML Schema, XLink, XPointer, XPath and DOM. All these standards and technologies are useful for exchanging business information by using the Internet. EDI is one of the oldest technologies for e-business and has been used by shipping companies for more than 30 years. The very existence of EDI is threatened by the arrival of XML technologies. But the question arises, is XML the end of EDI? What is the difference between EDI and XML? And to what extent is EDI relevant in today's e-business. The next chapter compares the EDI and XML for E-commerce applications.

CHAPTER 3

EDI AND XML

3.1 Introduction to EDI

Electronic Data Interchange (EDI) is one of the oldest forms of e-commerce, which has been around since the 1970s, exchanging electronic commerce messages over communications networks before the Internet. EDI is mainly designed to replace the paperwork with electronic files, and it is generally used by large organizations to exchange business transactions with their suppliers (Morrison, 2000, p.558).

Some of the benefits of EDI include:

- It is faster to exchange and process electronic documents.
- It eliminates the typing errors by entering the data only once.
- It reduces the human resources required to process electronic documents.

3.2 EDI (Electronic Data Interchange) definition

EDI is defined as the transfer of business transactions from computer-to-computer in an agreed standard to structure the data related to that transaction (ESCAP, 1993).

3.3 Key elements of EDI

The key elements of an EDI system (see Figure 3.1) are:

1. Computer Hardware
2. Communication Network
3. Computer software for translation and processing
4. Message standard (Jayakody, 1994, p.5).

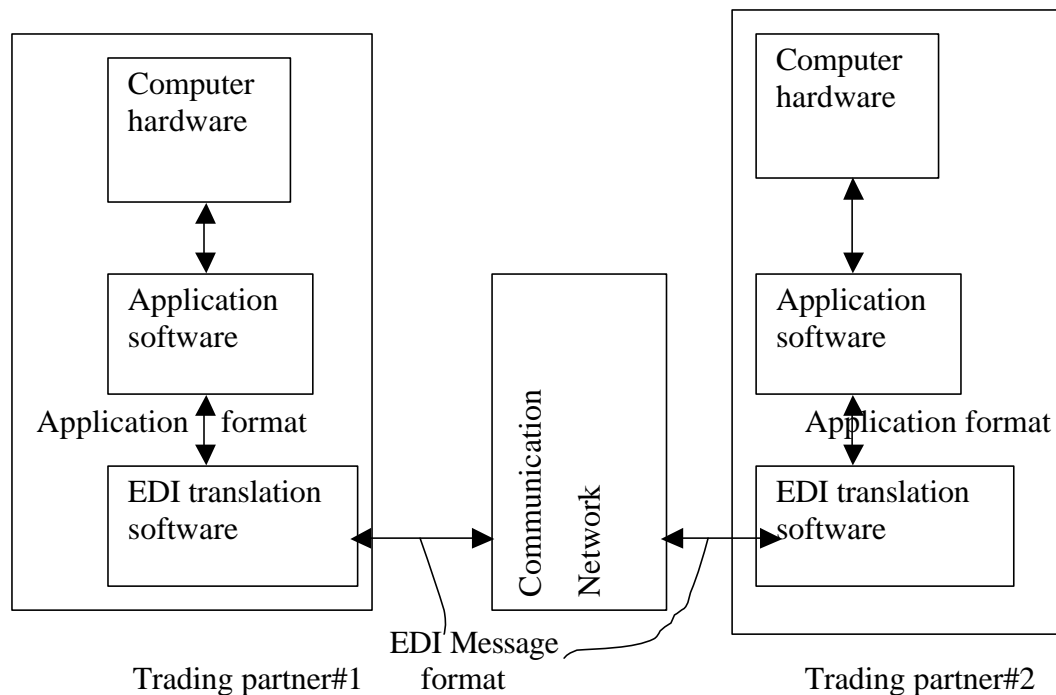


Figure 3.1: EDI system

Source adapted: Gerson, 1989, p.32

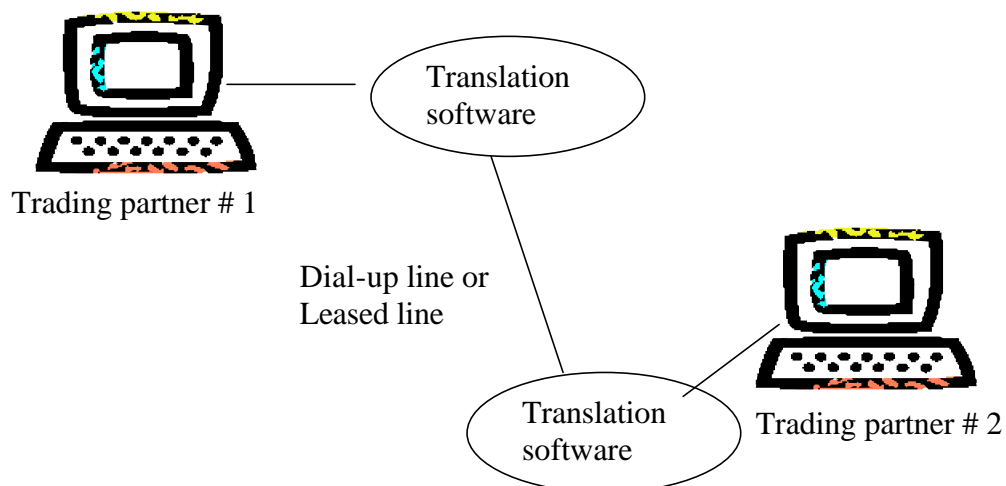
The computer hardware and the communication network are the physical infrastructure required for the data communication from computer-to-computer. It is necessary for any computer-based application, which is accessed over the network.

The communication network methods can be:

1. Direct communication by leased line or dial-up line (see Figure 3.2)
2. Indirect communication by using the Value Added Network (VAN) of a third party service provider (Figure 3.2).

The computer software and the message standard are the important key elements, which differ in computer applications.

1. Direct communication method



2. Indirect communication method

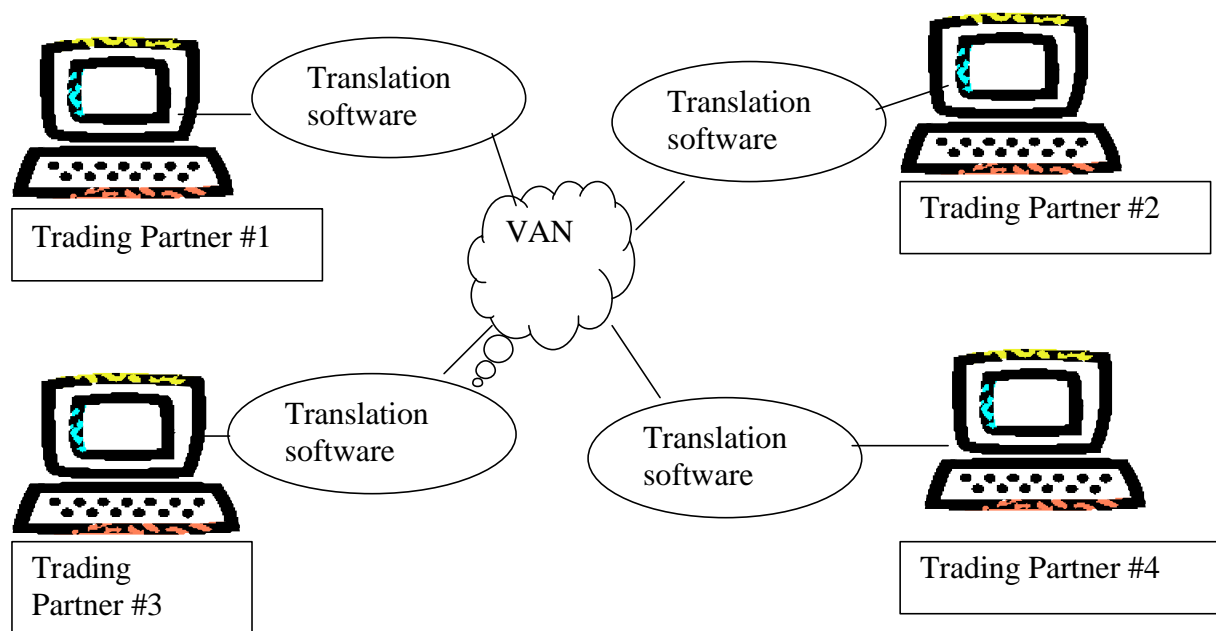


Figure 3.2: Communication Network Methods in EDI

Source: ESCAP, 1993, p.5

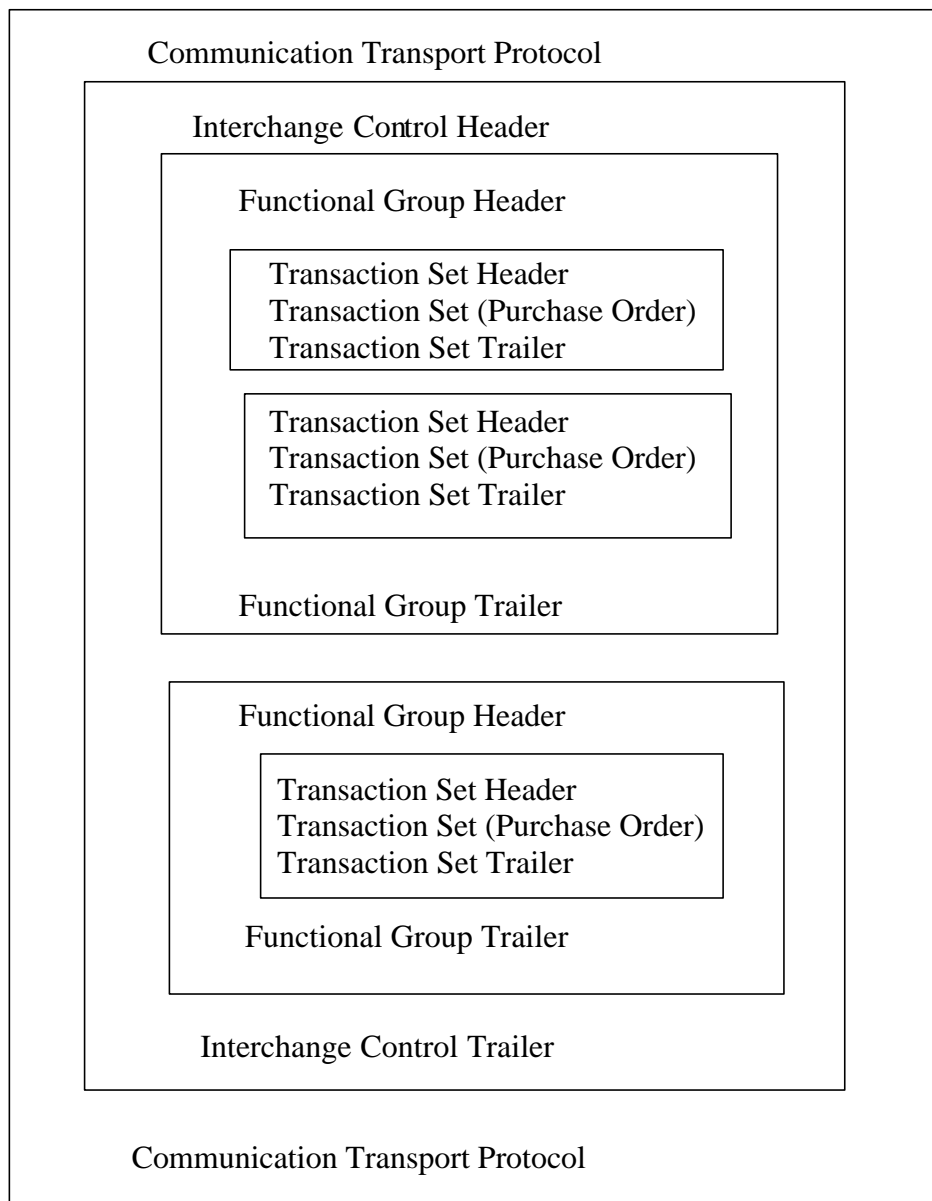


Figure 3.3: Data structure of an EDI transmission.

Source: Powers, 1989, p.9

3.4 Translation software

Translation software is used for converting the EDI message into the required format by the end-to-end computers. Each trading partner agrees the EDI message standard

but the trading partners can have different translation software. However, the output format of the EDI message should be the same so that they can transfer the messages.

3.5 Message standard

At present, most EDI transactions are using either the American National Standard Institutes (ANSI) Accredited Standards Committee (ASC) X12 standard or the UN EDIFACT (Electronic Data Interchange For Administration Commerce and Transport). X12 is used predominantly in the US while EDIFACT enjoys a worldwide audience. Other popular EDI formats include Odette (for the automobile industry), Tradacoms (which is UK-based) and Swift (which is used in international banking). Although the various EDI formats are different in details, they are based on the same principles (Marchal, 2001,p132). Generally, each electronic document is treated as a single EDI document transaction. Transmission headers distinguish between document types like purchase order or invoice. As a result, a variety of EDI documents can be combined and transmitted as a single EDI document (see Figure 3.3) (Powers, 1989, p.9).

EDIFACT has four key components:

1. Data elements to describe data
2. Data Segments – groups of data elements
3. Syntax rules to assemble a message
4. Standard message which consists of the above three key elements i.e. Data elements, data segments and syntax rule.

(Samaranayake, 1997, p.52).

Each EDI message has

1. User data segments to describe the actual content of data
2. Service segment to describe the syntax rule used.
3. Control segment to describe the start and end of each segment.

The syntax rule describes whether the segment or data is mandatory or optional or conditional. It also describes how the data is to be represented and grouped (Samaranayake, 1997, p.60).

Example of EDI message in EDIFACT:

NAD+CN+THE SCI LTD:MADAM CAMA ROAD:MUMBAI:INDIA'

3.6 Comparison between EDI and XML

EDI has been around since the 1970s and it has developed over a period of time. It is mainly used for E-commerce to exchange business transactions over the network. EDI is the generic term used for transferring business information. XML itself cannot displace EDI except in conjunction with many other technologies and a service infrastructure (Morrison, 2000, p.559). Hence, while comparing EDI and XML, the comparison between the EDI message standard, the main rules of EDI and the underlying technology are important.

3.6.1 E-commerce

EDI has served the objective of exchanging business transactions for more than 30 years. It has developed from the purpose of converting paper-based information into electronic information. Now, EDI is also used for exchanging crucial information between trading partners over the network. XML can also be used to exchange business information over the Internet. Thus, EDI and XML serve the common purpose of E-commerce for the industry.

3.6.2 Computer hardware

EDI can be implemented on any computer from the mainframe computer to the present PCs and the UNIX computers. Since XML is a new standard, it works with PCs and the UNIX computers. The users of EDI and XML can use the same

computers for E-commerce. For instance, the present users of email and Internet can use their computer hardware for EDI as well as for XML.

3.6.3 Communication network

EDI trading partners use private networks to exchange the EDI messages. They have direct communication either by leased line or dial-up connection. It solves the problem of authentication and the privacy of information since only trading partners communicate over the network. The VAN (Value Added Network) service providers can also be used to connect two or more EDI trading partners. The VAN acts as the intermediate channeling agency to transfer EDI messages between trading partners. Since the communication network is private and direct, the cost of transmission of EDI messages become expensive. XML is an internet-based standard and hence the Internet network can be used for transferring the XML-based messages. As the Internet is the most popular network, cheap, reliable and easily available, XML based systems can be deployed on the Internet communication network. The shipping companies are widely using Internet for email and Websites and also for many online e-commerce applications. Trading partners need not subscribe to the VAN service provider or connect by a leased line or dial-up connection. This will save communication costs of business transactions over the network.

3.6.4 Translation software

EDI translation software is required to convert the data created by the proprietary applications into the standard EDI message format. These kinds of software are deployed by each trading partner. They are customized to connect to the proprietary applications of the trading partners. There are limited numbers of suppliers of these software packages due to which they tend to be expensive to deploy. Furthermore, they are developed in old programming languages. XML is a new standard based on the Internet technology and it is widely supported by the existing tools and software. Many of them are free to download and well supported by the international community. They are used for developing the translation software for XML based

solutions. Many applications are being developed by using the Internet technology and hence XML based systems can be directly integrated with the applications of the trading partners. Therefore, new XML based systems may not require any translation software.

3.6.5 Message standard

EDI trading partners must use the same message standard like EDIFACT to exchange business transactions. If a trading partner uses a different message standard or upgrades it due to changes in EDI standard, other trading partners have to change or upgrade their translation software as well. It becomes expensive to maintain and support an EDI system and discourages trading partners to connect to different message standards. The dominant trading partner makes the other partners to use the same message standard..

The EDIFACT message standard is relatively old and is developed to limit the size of an EDI message by using abbreviations and codes, which are difficult to remember and understand for the normal human being. Figure 3.4 shows an example of an EDIFACT message for a purchase order of engine spares by an India based company. Now the same message can be sent by using the XML specification and it can be written in XML (Figure 3.5).

1. EDIFACT message

```

UNH+1+ORDERS:D:96A:UN'BGM+220+ABC123+9+AB'DTM+137:20020310:10
2'
DTM+61:20020410:102'NAD+BY+++SCI LTD+245 MADAM CAMA
ROAD+MUMBAI+MS+400021+IN'NAD+SE+++NIPPON CORPN+TAMACHI
ROAD+TOKYO+120+JP'
LIN+1'PIA+5+ E123456:SP'QTY +21:5'PRI+AAA:24.99::SRP'
LIN+2'PIA+5+ E45678:SP'QTY +21:10'PRI+AAA:42.50::SRP'
UNS+S'CNT+3:2'UNT+17+1'

```

2. Explanation of message

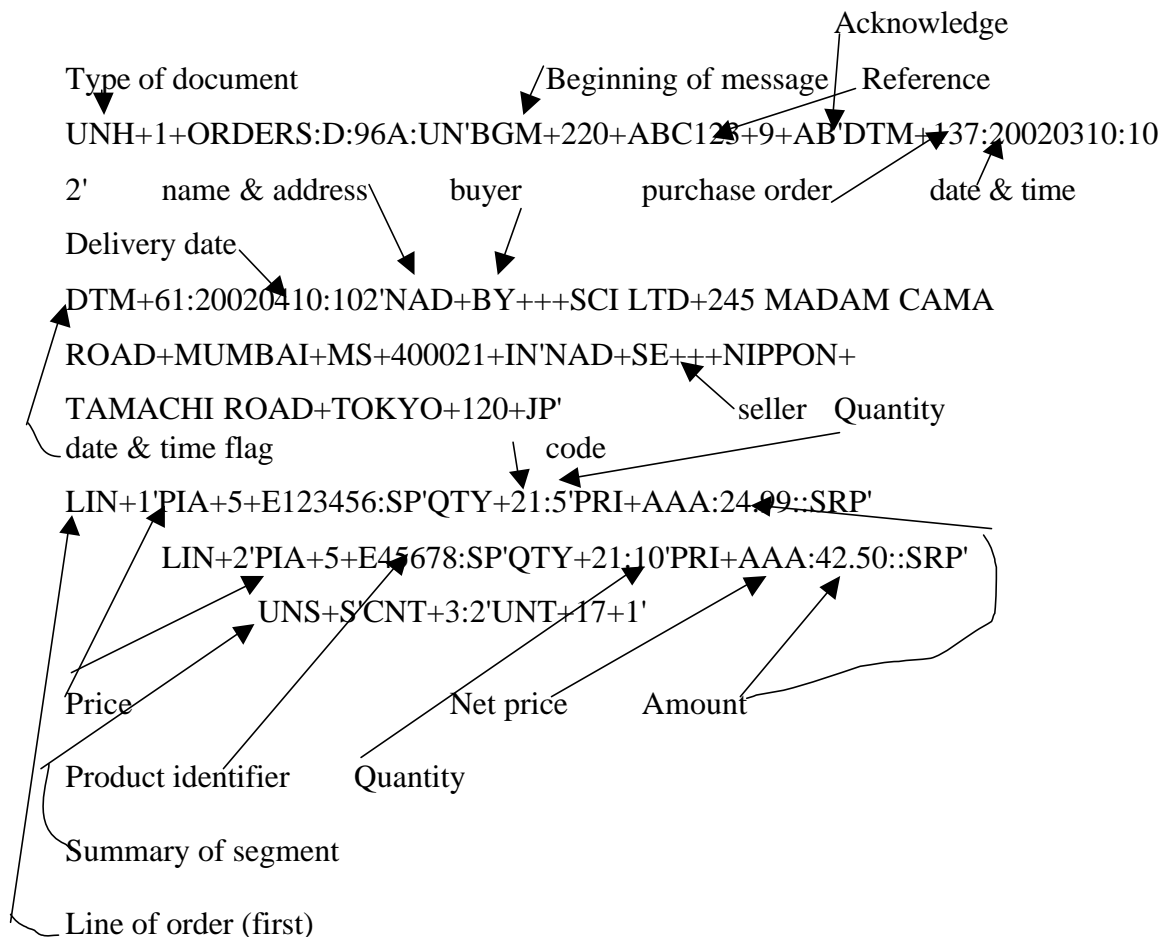


Figure 3.4: EDIFACT message to purchase engine spares

Source adapted: Marchal, 2001, p.134

```

<?xml version="1.0"?>
<Order confirm="true">
  <Date>2002-03-10</Date>
  <Reference> ABC123</Reference>
  <DeliverBy>2002-04-10</DeliverBy>
  <Buyer>
    <Name> SCI LTD </Name>
    <Address>
      <Street>245 MADAM CAMA ROAD </Street>
      <City> MUMBAI </City>
      <State>MS</State>
      <PostalCode>400021</PostalCode>
      <Country>IN</Country>
    </Address>
  </Buyer>
  <Seller>
    <Name> NIPPON CORPN </Name>
    <Address>
      <Street> TAMACHI ROAD</Street>
      <City> TOKYO </City>
      <PostalCode>120</PostalCode>
      <Country>JP</Country>
    </Address>
  </Seller>
  <Lines>
    <Product>
      <Code type="SP">E123456</Code>
      <Description>Main Engine Ring</Description>
      <Quantity>5</Quantity>
      <Price>24.99</Price>
    </Product>
    <Product>
      <Code type="SP">E45678</Code>
      <Description>Auxiliary Engine Ring </Description>
      <Quantity>10</Quantity>
      <Price>42.50</Price>
    </Product>
  </Lines>
</Order>

```

Figure 3.5: List of purchase order in XML

Source adapted: Marchal, 2001, p.141

Thus, it is easy to write the message in XML and a human being can understand it. Furthermore, the syntax rules for XML are easy to understand and they are developed for non-programmers. The Web browser can directly display the XML messages and the user can correct the information. The XML message can also be updated in the computer (Figure 3.6).

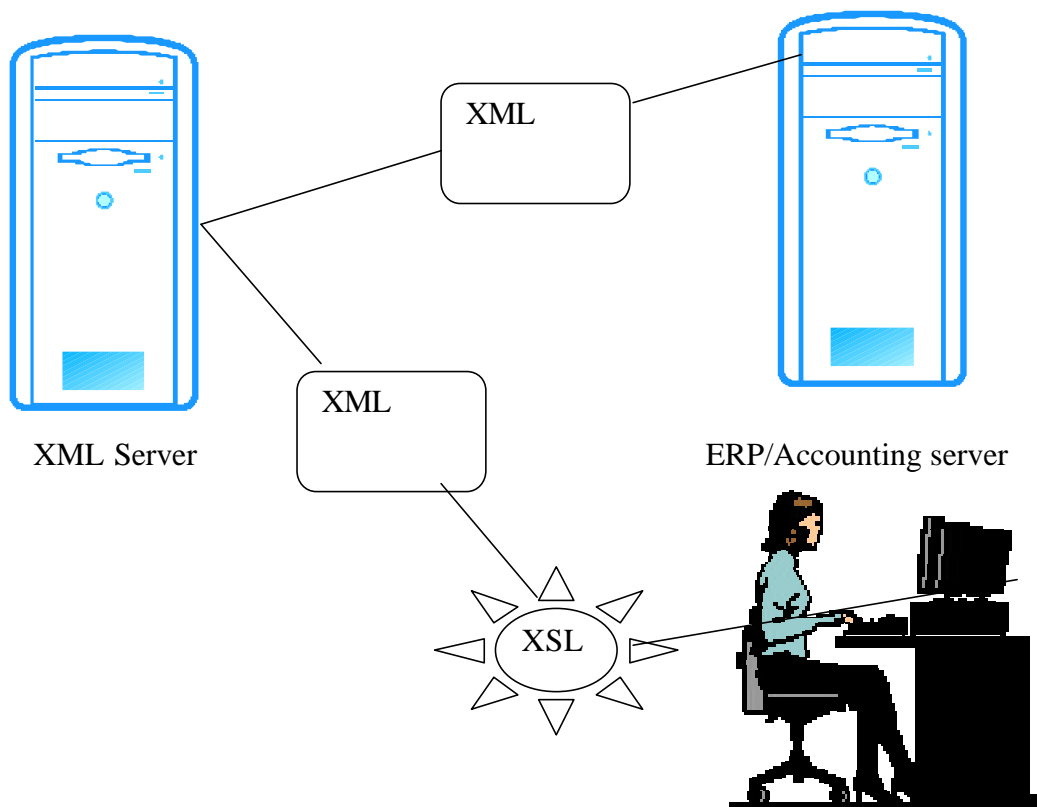


Figure 3.6: XML supports both users on a workstation and a server

Source: Marchal, 2001, p.140

3.6.6 Data and document integration

EDI technology separates the document and data into business and legal context making it inflexible and complex to handle. XML technology is based on Internet,

which enables the integration of document and data, which makes the document simpler, less expensive and more efficient to handle (Asay, 2000, p.293).

3.6.7 Business issues

The deployment of EDI is mostly done by large corporations, which have connected their EDI system with their trading partners. The centralized i.e. hub and spoke, approach is followed by EDI users while Web technologies support less centralized business models. Thus, large organizations have always tried to control the EDI setup, which gives less incentive to small organizations. XML based systems can be deployed by any organization irrespective of its size due to the decentralized nature of Web technology (Linn, 2001).

3.6.8 Standard

Though EDI message standards have been around for 30 years, EDI systems have not become the standard for e-commerce. Today, many e-commerce systems are developed around the Internet technology. Hence, Internet has become the de facto standard for deploying e-commerce applications. XML fits into the Internet technology (Atkinson, 2001).

3.6.9 Online transactions

EDI cannot be used for online commerce applications, which is the need of many new businesses. EDI is not a front-end technology but rather a back-end technology, which requires a separate front-end support of other technology. XML can be deployed at the front end as well and data can be directly updated into the XML messages.

3.6.10 Internationalization

EDI systems do not support many languages due to their limited character set developed on the ASCII (American Standard Code for Information Interchange) standard, which supports only 256 different characters. XML supports 65,535 different characters for any international language since XML uses the UNICODE character set. Thus, XML based systems can handle characters of any language.

3.6.11 Availability of development tools, software and support

EDI systems have limited development tools, software and support while XML based systems have a large number of development tools, software and support. The present software packages are developed by using Internet based technology and hence supporting the XML based system is easier and cheaper.

3.6.12 Cost and time to implement

The cost of deploying an EDI system is substantial. Particularly, the software, hardware and network connectivity of an EDI system is expensive and specialized. It is also costly to operate and maintain (Linn, 2001). Today, companies can deploy Internet based applications faster and more easily. Further, EDI systems are using multiple standards, which also adds to the cost of operating EDI. XML based systems are developed by using the Internet technology, which makes the e-commerce less expensive and easier.

Thus, XML based systems are much superior in all respects compared to the present EDI message standards. However, XML and EDI can be deployed together by employing basic EDI practices but describing the data by using XML specification instead of a proprietary format. The flexibility of XML and the established system of data exchange can be combined to develop the e-commerce faster and easier. Many organizations can benefit by following this approach and agree on the common XML/EDI standard. Furthermore, information can be exchanged and shared over the Internet irrespective of the computer system setup of each trading partner. (Tittel & Boumphrey, 2000, p.49).

CHAPTER 4

IMPLEMENTATION OF XML BASED SYSTEMS

In the previous two chapters, the specification of XML, its related technology and its advantages over EDI has been explained. This chapter explains the specific features of XML, types of XML-based applications, various areas in shipping companies in which it can be implemented, and the process of implementing an XML-based system. A case study on the implementation of an XML-based system is also discussed. Finally, the benefits and limitations of XML-based systems are explained.

4.1 Features of XML

Before identifying the various application areas for implementing XML based systems, it is worth highlighting some of the features of XML, which are suitable for those application areas.

4.1.1 Standard File format

XML promises to liberate information from the tyranny of proprietary file formats such as RTF (Rich Text Format) used in Microsoft Word, PDF (Portable Document Format) used in Adobe Acrobat (<http://www.adobe.com/products/acrobat/adobepdf.html>), NSF (Notes Format) used in IBM Lotus Notes (IBM, 2002) and ASCII (American Standard Code for Information Interchange). XML makes it possible to transfer information among multiple computer programs running on different computer platforms without the need for conversion programs (Laurent, 1999, p. 4).

4.1.2 Self-describing and clear document

XML allows the users to define their own elements and attributes within a document and, therefore, it is called an Extensible Markup Language. The users define their own markup language in XML, which provides a general syntax to describe how a markup language looks and behaves (Tittel & Boumphrey, 2000, p. 18). For instance, a vessel name can be described as <VESSEL.NAME> in an XML document. XML documents are reasonably clear to the laypersons. The structure of an XML-document is easy to define by specifying a Document Type Definition (DTD) or an XML Schema. An XML document can be validated automatically by XML parsers. Furthermore, an XML document can be reused and exchanged easily due to the element-wise definition in XML rules and syntax (Turk, 2000, p. 19).

4.1.3 Development tools and support

Web protocols such as HTTP (Hyper Text Transfer Protocol) and SMTP (Simple Mail Transfer Protocol) and mechanisms such as URLs are used in XML and they do not impose any additional requirements. XML documents and applications can be developed and processed by using internet programming languages like Java, SmallTalk, C, C++, JavaScript, Tcl, Perl and Python. These are the main programming languages for creating web-based applications (North & Hermas, 2002, p.13). XML allows transferring of documents straightforwardly over the Internet and supports a wide variety of applications. An application that understands XML can send output data to a terminal, printer or for a speech synthesis program that reads the text to a blind or low vision person. Alternatively that same data can be used by another program updating the central database (Tittel & Boumphrey, 2000, p. 21).

4.1.4 Open standard

XML is an open standard, which cannot be monopolized by anyone. No single person or commercial entity has control over its destiny. Documents in XML belong to their owner; they do not belong to any software package, any computer

configuration, or any operating system (McGrath, 1998, p. 79). The complete specification for XML is available to the public at the website of the World Wide Web Consortium (W3C), <http://www.w3.org/XML>. These specifications are non-proprietary, open and well documented. All the development in XML is completely available to the public. XML is the successor of HTML standards, which are also developed by W3C. (Laurent, 2000, p. 8).

4.1.5 Multi-lingual support

XML is built with internationalization in mind, using the UNICODE (<http://www.unicode.org/>) character set and providing language identifiers in elements. The UNICODE character set provides 65,535 possible characters to create XML documents, which enables countries to use their own language (Laurent, 1999, p. 9). The UNICODE Standard has been accepted by the computer industry leaders like Apple, HP, IBM, Microsoft, Oracle, SAP, Sun, Sybase, Unisys and many others. UNICODE is required by modern standards like XML, Java, ECMAScript (JavaScript), LDAP, CORBA 3.0, and WML. Current operating systems, Web browsers, and many other programs have built-in support for UNICODE, which enables a single software program or a single website to be accessed across multiple platforms, languages and countries without any changes in the software. It allows transmission of data through many different systems without corruption or loss (<http://www.unicode.org/>). It is particularly important for shipping companies, which operate at a global level and shipping documents and information have to be transmitted to different countries in different languages (Verboom & Jellema, 2001).

4.1.6 Data and document capability

XML has the capability to describe both data and documents and this capability is poor in HTML. Due to the support for custom markup vocabularies, XML opens the door for carefully describing data and the relationships between pieces of data.

Since XML describes content, document search is possible on a particular information and it can be displayed on the web browser. At present, searching

HTML described documents displays too much unnecessary information. This approach to searching information is light years ahead of the present search capabilities of HTML (Morrison, 2000, p.13).

4.2 Type of applications

XML can be applied to various kinds of applications such as:

1. Document management
2. Business-to-business data exchange
3. Channel Definition Format (CDF)
4. Synchronized Multimedia Integration Language (SMIL)
5. User defined markup language
6. Web services

4.2.1 Document management

In document management applications, contents can be stored in the XML format and the display of the content can be managed by CSS (Cascading Style Sheet) or XSL (Extensible Style Language). It is easier to concentrate on the development of content and assign the development of display to web developers. XML is well suited for delivering content to devices ranging from cell phones to desktop PCs and from satellite feeds to shopping mall kiosks. By storing raw content encoded in XML, content-management systems can use XML style sheets to format the look and feel of the content for the appropriate device. The display and delivery of the contents can be managed by web developers and they can write appropriate XSL to deliver content to a standard web browser in HTML or to a mobile device in WML (Wireless Markup Language) or to any other output device. It can be achieved by using software tools like Cocoon (<http://xml.apache.org/cocoon/introduction.html>). The document management application is widely used in the publishing industry. XLink (a feature of XML) enables the display of a particular part of a document, which makes the search faster.

4.2.2 Business-to-business data exchange (next generation EDI)

The exchange of data in XML format frees shipping companies from using proprietary formats. The XML format can be used for exchanging the business data asynchronously and it can be implemented as a low cost EDI implementation (Voltsch, 2002). The data can be transferred from computer-to-computer and can be updated directly into the database. This feature can be used for e-commerce and e-business applications. Moreover, XML can be used to share business data between business partners or to integrate data flow between different applications across the Internet.

Shipping companies can use the XML format to exchange data in the following cases:

- Shore-to-ship communication for sending instructions from the head office;
- Ship-to-shore communication for reporting status and indents;
- Ship-to-third parties such as agents, ports, weather “routers”, bunker suppliers, insurance, P and I club, class, and shipyard;
- Ship-to-port for communicating the notice of readiness (NOR);
- Business-to-Business (B2B) such as communication with other carriers, MTOs (Multi-modal Transport Operators) and suppliers.

(Drewry Shipping Consultant, 2000).

4.2.3 Channel Definition Format (CDF)

CDF is an XML application designed to categorize and describe a web site content and to make that data available to the users on demand. It allows a web server to send automatically all the latest information from a web site to the user’s web browser instead of the web browser requesting for the information. The information is sent to only those users who have subscribed to those web servers (<http://www.w3.org/TR/NOTE-CDFsubmit.html>).

The CDF feature is supported by web servers like Internet Information Server (IIS) (Microsoft, 2002). It can be used to automatically send the information about the

vessel's schedule to various clients and agencies. This will make the information available to the users instantly.

4.2.4 Synchronized Multimedia Integration Language (SMIL)

SMIL (current version 2.0) is an XML application to describe the temporal behavior of a multimedia presentation, associated hyperlinks with media objects and to describe the layout of the presentation on a screen (<http://www.w3.org/TR/smil20/>). This application can be used for imparting web-based training to seafarers by shipping companies.

4.2.5 User defined markup language

XML allows the users to define their own elements and attributes within a document and, therefore, it is called an extensible markup language. The users define their own markup language using XML, which provides a general syntax to describe how a markup language looks and behaves. A Document Type Definition (DTD) or an XML Schema is created to define the rules for the elements (markup tags) and attributes (values associated with specific tags), which are allowed in documents (Tittel & Boumphrey, 2000, p. 18). This feature of XML allows the shipping companies to create their own language and transfer the information by using the common vocabulary. For instance, MTML (Marine Trading Markup Language) or Bolero-XML can be used to prepare XML documents by shipping companies. Bolero.net has actually provided a set of electronic documents for trade facilitation (Bolero, 2002b). There are many XML-based languages such as MathML (Mathematical Markup Language) (<http://www.w3.org/Math/>), CML (Chemical Markup Language), GeoML (Geological Markup Language), RELML (Real Estate Listings Markup Language), HRMML (Human Resource Management Markup Language), and VoxML (Voice Markup Language).

4.2.6 Web services

Web services enable different applications to communicate with each other, share data and invoke each other by using XML messaging over standard Web protocols such as HTTP. Almost any platform and language combination can run Web services (Ruben, 2001). At present, .Net platform from Microsoft and J2EE (Java 2 Platform Enterprise Edition) from Sun Microsystems are Web services products. Shipping companies can implement these web services to send the information about container tracking, vessel schedules or weather reports to their offices, ships or business partners. They can also use Web services for communicating with applications of their business partners or clients (Cerami, 2002). Web services are particularly important for sending electronically the information about the containers. Recently, the US government has asked shipping companies to declare US shipments electronically before loading cargo (DigitalShip, 2002).

4.3 Areas in shipping companies

Shipping companies can implement XML-based applications in many areas. Some of these areas are mentioned below although the list is not exhaustive:

4.3.1 E-business

Shipping companies can implement XML-based applications for transferring data to their business partners or suppliers (Figure 4.1). The business partners may be carriers, multi-modal operators, freight forwarders, shippers, consignees, customs, and ports. The suppliers may be bunker suppliers, ship chandlers and spare parts manufacturers. Shippers may require services like online cargo bookings, online cargo tracking, automatic customs documentation, and automatic export documentation. These services can be offered by implementing XML-based applications. Information Systems Agreement (ISA), a consortium of shipping lines, has launched an XML standard for bookings (Shipping lines..., DigitalShip, April, 2001).

At present, some shipping companies are exchanging business data in proprietary business format or the EDIFACT or ANSI X12 message standards. The business data could be exchanged data in a standard XML message format. XML interfaces are required to translate the data from the existing database into an XML format. The interfaces can be developed in Internet programming languages; plenty of tools, and skills are available in the market to develop such interfaces. Shipping companies can use the Internet to transfer data instead of using their own private networks or private VAN (Value Added Network).

Shipping companies could use the XML based e-business standards developed by ebXML (Electronic Business XML) or RosettaNet. RosettaNet has developed an e-business framework based on XML standards for the IT and electronic industry (<http://www.rosettanet.org>). These could be used in implementing certain XML based applications like supply-chain management.

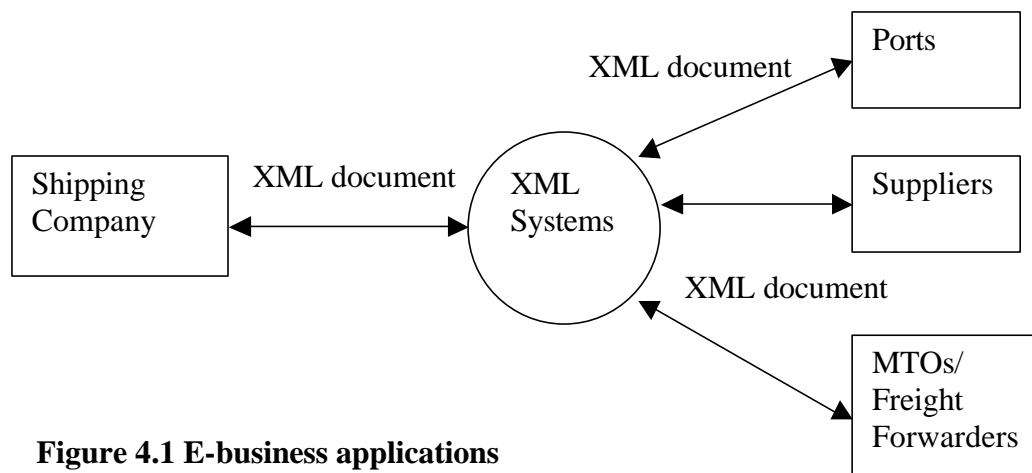


Figure 4.1 E-business applications

4.3.2 Building of a content management system and IETM (Interactive Electronic Technical Manual)

Planning and maintenance of ships is very important for their safe and efficient operations. There are many manuals and operating instructions to be followed to operate onboard systems. Ships have to maintain drawings of these systems. The

manuals, drawings, and operating instructions can be electronically stored on a CDROM. The content of the CDROM can be prepared in XML. XML allows creating of content, which can be further displayed on a Web browser such as Internet Explorer or Netscape Navigator by using the XSL. The content can be regularly updated from the head office and it can be delivered in many languages. Thus, ships with crew of different nationalities can access the planning and maintenance system and manuals of the on board system. A similar kind of system has been developed for maintenance, repair and upgrades of the F-117 stealth fighters (Figure 4.2) by Lockheed Martin Aeronautics, and Calian and Luoma & Associates, jointly. The main features of the system are:

- Contents, i.e manuals of operation, and maintenance of systems are created in XML. The contents are displayed dynamically on the Internet Explorer 5 by using XSL and XSLT. The interactive features are added by using JavaScript.
- Contents are stored on a CDROM for security reasons.
- The system can be easily linked to the ground shore systems such as inventory and ordering systems.
- It meets the US military standard JCALS (Joint Computer Aided Logistics and Support) 38784 to deliver paper documentation on computer by formatting and presenting the paper-based information.

(Luoma, 2001).

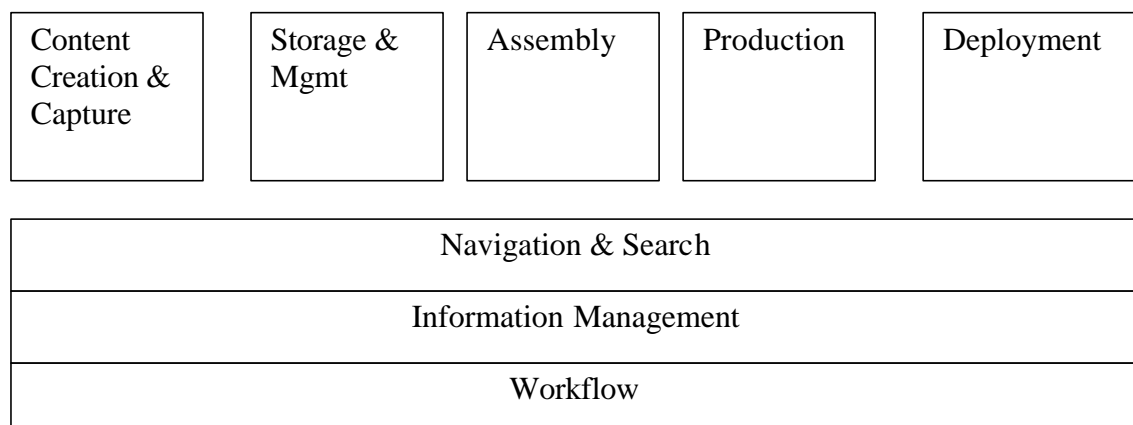


Figure 4.2: Architectures and Technical Solutions for F117 Fighter
source: Luoma, 2001

4.3.3 Ship-to-shore communication

Ships regularly have to send information to shore offices and ports. This information can be stored in an XML format and to be sent to the shore offices by email. The information may be related to

- Indents for spares,
- Requirement of supplies
- Crew planning and management
- Cargo planning and monitoring onboard maintenance
- ISM/STCW code compliance reports
- Video/image of ship parts
- Advice from shore
- Vessel survey - classification surveys and survey information requirements
- Hull maintenance
- Vessel drawings
- Port clearance, position and data reporting, vessel turnarounds, fuel stocks, consumption
- Parameters affecting vessel's overall operation
- Arrival reports, stores inventory

(Fairplay, 1999).

4.4 Implementation of XML based systems

Figure 4.3 shows the schematic representation of a common XML architecture to implement an XML-based system, which consists of the following components:

1. XML Documents
2. XML Parser
3. XML Processor
4. Application server
5. Database server

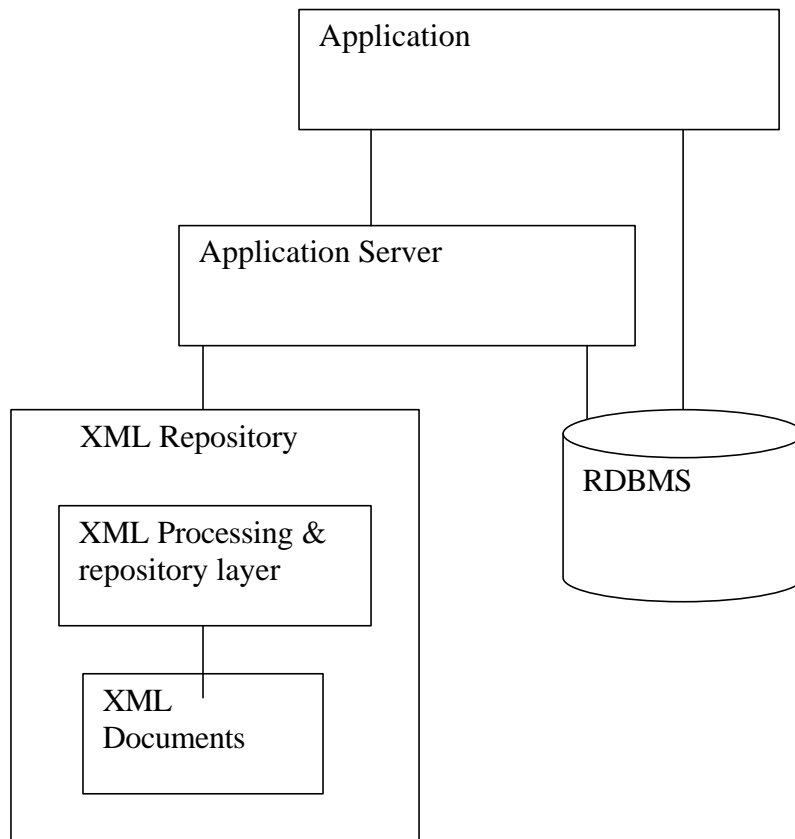


Figure 4.3: Common XML Architecture

Source: Oracle, 2002

A complete solution of an XML-based system has an user interface to display the XML document, business logic to process the XML document and database to store an XML document (Figure 4.4). An XML document is created in an XML standard format and the business logic is developed by using Internet programming languages. The database may be the native XML database or any relational database (RDBMS) such as Oracle, DB2 and MS SQL server.

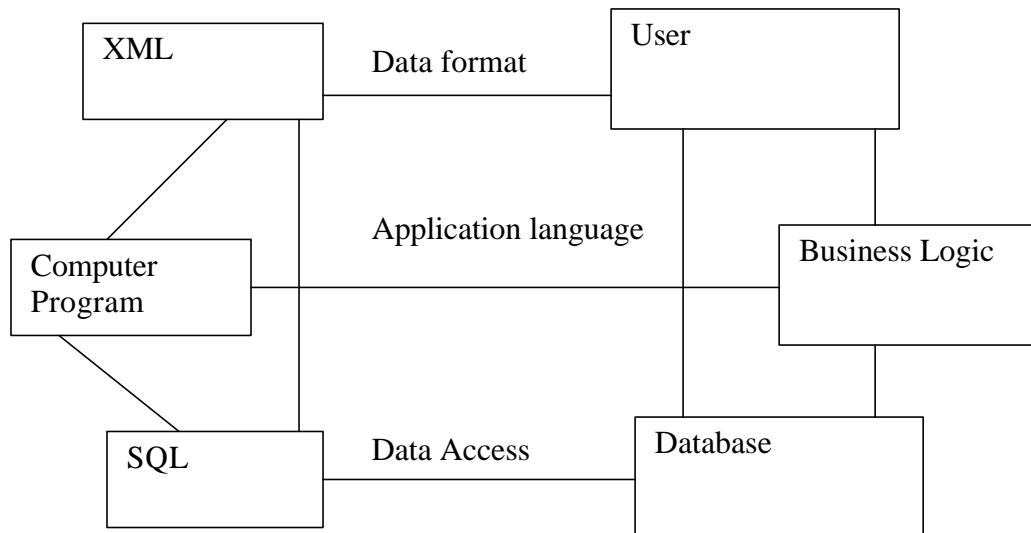


Figure 4.4: Complete XML Solution

Source adapted: Shmeitzer, 2002

4.4.1 Preparation of XML document

Initially, the kind of information to be exchanged between business partners should be identified. Then a common XML vocabulary should be agreed upon to identify the data items. The vocabulary used by another organizations can also be used. For example, MTML (Marine Trading Markup Language) is being developed for the maritime industry. It can be used in preparing XML documents. Similarly the XML standards developed in ebXML or Bolero-XML are useful in preparing XML documents. After identifying the information, XML documents can be generated by using XML editor tools¹. XML editor tools are available on a commercial basis as well as free of cost. An XML document may have a DTD or an XML Schema. DTD is generally used for documents with mainly group of texts while XML Schema is great for documents with lots of data (Tittel, Pitts & Boumphrey, 2002, p.111). The

¹ XML editing tools are mentioned in chapter 2 (section 2.4.1.1)

DTD or XML Schema of any organization can be reused so that errors can be minimized to save time and also to avoid errors.

4.4.2 Selection of application server

An application server consists of a business logic program, and an XML processor to extract data from the XML documents. The business logic program has the business rules and instructions to update the database. The business logic program and XML processor can be developed in any Internet programming language. These programs can be specifically developed in Java since XML and Java fit extremely well together, both supporting hierarchical structures and a common foundation of UNICODE for text. XML provides a sophisticated format to store and transfer data while Java provides an elegant development environment for manipulating and presenting all kinds of information. In a combination, the two give developers an extremely powerful tool set, capable of handling information in a variety of structures from a variety of sources and keeping it all interchangeable. (Laurent, 1999, p.3).

4.4.3 Selection of database server

A database server is required to store XML documents. Now-a-days new versions of databases are available which also support XML formats. These databases can directly store the XML data and generate the required XML documents. For example, dbXML is an XML compliant database and it can be downloaded freely (<http://www.dbxml.com/index.xml>). Ozone is another XML compliant database freely downloadable (http://www.ozone-db.org/ozone_main.html). The other databases are Oracle, MS SQL Server, IBM DB2, Access or MySQL (it is available free of cost from the Internet). To gain some experience in implementing XML based solutions, it is recommended to use the existing database servers. Any database can be selected depending upon requirements and budget.

4.4.4 Selection of platform

Since XML is independent of platform, the users can select the platform of their choice. Initially, the users can implement the XML applications and database on their existing platforms.

4.4.5 Choosing the mode of transmission

As an XML document can be transmitted by email, FTP (File Transfer Protocol), or HTTP, the users can choose the mode of transmission based on their needs as well as the infrastructure of their business partners.

4.4.6 Decision on security policy

XML security can be handled by using cryptographic techniques such as digital signatures and message encryption. XML Signature and XML Encryption are two new security initiatives designed to handle the XML data and messages in a secured way. XML Signature is a joint effort between the World Wide Web Consortium (W3C) and Internet Engineering Task Force (IETF), and XML Encryption is solely a W3C effort. W3C has recommended the standards for XML-Signature and syntax processing on February 12, 2002 (<http://www.w3.org/TR/xmlsig-core/>). Standards for XML Encryption and XML Public Key are currently being developed. These standards allow identifying the sender and maintaining the integrity of data (Siman, Madsen, & Adams, 2001).

The list of various tool kits for security is available on the W3C's website (<http://www.w3.org>).

XML security standards are not yet fully ready for convenient use, reliability and robustness, but they are likely to be available very soon (Mactaggart, 2001).

4.5 Case study on implementation of XML based systems

Since XML offers a standard format to exchange data, a case study is demonstrated on how business data can be transferred in an XML format. In this case study, a

shipping company desires to send the container delivery notice to consignees so that consignees can directly receive the information in their computers and take the necessary action to collect the container. This will save the time and resources of the shipping company as well consignees and the shipping company can also improve the service to its customers.

4.5.1 Objective

The objective of the case study is to demonstrate how business data in an XML document can be transferred between a shipping company and consignees. The business data consists of the details on the shipment to be delivered to the consignees.

4.5.2 Description

A shipping company would like to automatically send the container delivery notice to its consignees. The company has a computerized container system to store the information about containers to be delivered to consignees. The company generates container delivery notices as XML documents and sends them to its consignees over the Internet so that the consignees can directly receive an XML document into the computer for further processing. Once the delivery notice is received, consignees can approach the container yard to take delivery of the containers. Consignees can also link their XML based systems with customs and ports for Customs clearance (Kim, 2001).

4.5.3 Information flow

A ship sends the information about its likely arrival to the port and the shipping company (Figure 4.5). This information is updated in the database of the container system of the shipping company. The port sends the information on the berth allocated to the ship to the shipping company. Initially, this information is sent to consignees as a pre-delivery notice so that consignees are aware about the arrival of

shipments. Once containers are discharged and stored in a container yard, the port informs the location of these containers to the shipping company. The shipping company updates its container database and prepares a delivery notice as an XML document. This document is, thereafter, sent to the consignees.

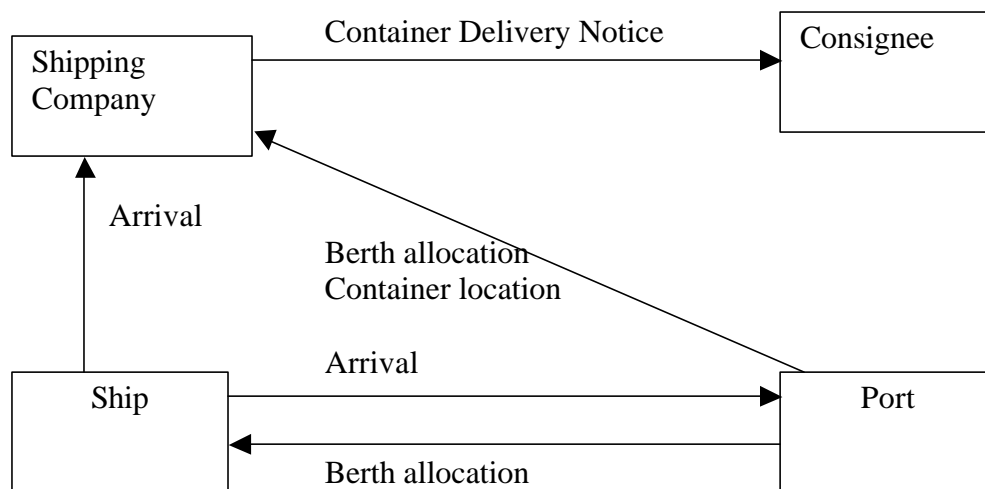


Figure 4.5: Information Flow Diagram

4.5.4. System architecture

The system architecture of the Container Delivery System (Figure 4.6) consists of

- A database server to store records of containers to be delivered to different consignees.
- The XML application server, which has the business logic to process information. It communicates with the XML processor to send/receive XML data to/from XML documents.

- The XML processor to extract the information from the XML document. It also generates the XML document. The XML processor also has an XSL code to display the XML document on the Web browser.
- Internet connectivity to transfer XML documents from the shipping company to the consignees. The sender and receiver of the XML document are identified by the unique Internet IP address.
- The authentication of the sender can be verified by the user name and password by the computer system running at the consignees' locations. This will handle the authorization of the sender of the XML document.
- The XML document is sent in encrypted form; the encryption software is installed by the shipping company as well as by the consignees. This will handle the security and the integrity of the XML document.

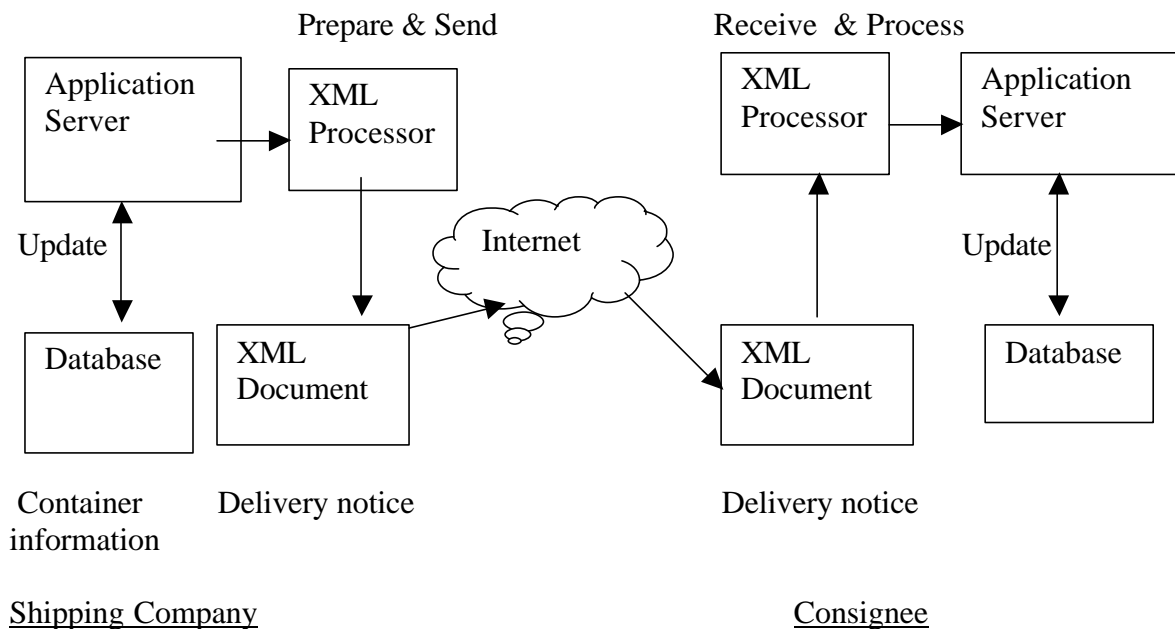


Figure 4.6: System architecture of container delivery system

4.5.5. XML documents used

Each unit i.e. ship, port and shipping company, sends the data as an XML document, hence the following XML documents are required:

1. Container delivery notice
2. Ship arrival notice
3. Ship berthing notice and container location notice

For each XML document, an XML specification and DTD are created. The detailed list of XML documents, their corresponding DTD and explanation are given in Table 4.1.

Table 4.1: Cross reference table of XML documents

Document name	Document explanation	Document specification	Document DTD
Container delivery notice	Table 4.2	Figure 4.7	Figure 4.8
Ship arrival notice	Table 4.3	Figure 4.9	Figure 4.10
Ship berthing and container location	Table 4.4	Figure 4.11	Figure 4.12

Table 4.2: Explanation of Container delivery notice document

Element	Attribute	Description
CONTAINER.DELIVERY		Root element
SHIPPING.COMPANY		
CONSIGNEE.NAME		
VESSEL.NAME		
VOYAGE.NO		
ARRIVAL.DATE		
DISCHARGE.PORT		
BERTH.ID		
BL.NUMBER		
DELIVERY.REFNO		
CONTAINER.DETAILS		Group element
CONTAINER.NO	TYPE, SIZE	
CARGO	DANGEROUS	
CONTAINER.YARD.LOCATION		

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE CONTAINER.DELIVERY SYSTEM "file://c:\CARGO.DELIVERY.DTD">

<CONTAINER.DELIVERY>
  <SHIPPING.COMPANY>&SCI;</SHIPPING.COMPANY>
  <CONSIGNEE.NAME>ABC LTD</CONSIGNEE.NAME>

  <VESSEL.NAME>TAJ MAHAL</VESSEL.NAME>
  <VOYAGE.NO>786</VOYAGE.NO>
  <ARRIVAL.DATE>2002-09-11</ARRIVAL.DATE>
  <DISCHARGE.PORT>MUM</DISCHARGE.PORT>
  <BERTH.ID>12AD</BERTH.ID>

  <BL.NUMBER>SCI5489124</BL.NUMBER>
  <DELIVERY.REFNO>SWE456</DELIVERY.REFNO>

  <CONTAINER.DETAILS>
    <CONTAINER.NO TYPE="GT" SIZE="20">SCIU1234560</CONTAINER.NO>
    <CARGO DANGEROUS="NO">SWEDISH BEER</CARGO>

  <CONTAINER.YARD.LOCATION>MLD12</CONTAINER.YARD.LOCATION>
  </CONTAINER.DETAILS>

  <CONTAINER.DETAILS>
    <CONTAINER.NO TYPE="RF" SIZE="40">CTIU1234595</CONTAINER.NO>
    <CARGO DANGEROUS="NO">DANISH CHEESE</CARGO>

  <CONTAINER.YARD.LOCATION>CHB44</CONTAINER.YARD.LOCATION>
  </CONTAINER.DETAILS>

</CONTAINER.DELIVERY>

```

Figure 4.7: XML Document for Container Delivery Notice

```

<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY SCI "THE SCI LTD MUMBAI">

<!--ELEMENT CONTAINER.DELIVERY
(SHIPPING.COMPANY,CONSIGNEE.NAME,VESSEL.NAME,VOYAGE.NO,ARRIVAL
.DATE,DISCHARGE.PORT,BERTH.ID,BL.NUMBER,DELIVERY.REFNO,(CONTAINER.DETAILS)+)-->

<!--ELEMENT SHIPPING.COMPANY (#PCDATA)-->
<!--ELEMENT CONSIGNEE.NAME (#PCDATA)-->

<!--ELEMENT VESSEL.NAME (#PCDATA)-->
<!--ELEMENT VOYAGE.NO (#PCDATA)-->
<!--ELEMENT ARRIVAL.DATE (#PCDATA)-->
<!--ELEMENT DISCHARGE.PORT (#PCDATA)-->
<!--ELEMENT BERTH.ID (#PCDATA)-->

<!--ELEMENT BL.NUMBER (#PCDATA)-->
<!--ELEMENT DELIVERY.REFNO (#PCDATA)-->

<!--ELEMENT CONTAINER.DETAILS
(CONTAINER.NO,CARGO,CONTAINER.YARD.LOCATION)-->
<!--ELEMENT CONTAINER.NO (#PCDATA)-->
<!--ATTLIST CONTAINER.NO
    TYPE CDATA #IMPLIED
    SIZE (20|40) "20"-->

<!--ELEMENT CARGO (#PCDATA)-->
<!--ATTLIST CARGO DANGEROUS (YES|NO) #REQUIRED-->

<!--ELEMENT CONTAINER.YARD.LOCATION (#PCDATA)-->

```

Figure 4.8: XML DTD for Container Delivery Notice

Table 4.3: Explanation of Ship arrival notice document

Element	Attribute	Description
ARRIVAL.NOTICE		Root element
VESSEL.NAME	VESSEL.CODE	
VOYAGE.NO		
LOAD.PORT		
DISCHARGE.PORT		
ARRIVAL.DATE		
ARRIVAL.TIME		
BERTHING.DATE		
BERTHING.TIME		

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE ARRIVAL.NOTICE SYSTEM "file://c:\ARRIVAL.NOTICE.DTD">
<ARRIVAL.NOTICE>
<VESSEL.NAME VESSEL.CODE="M01">TAJ MAHAL</VESSEL.NAME>
<VOYAGE.NO>786</VOYAGE.NO>
<LOAD.PORT>CPH</LOAD.PORT>
<DISCHARGE.PORT>MUM</DISCHARGE.PORT>
<ARRIVAL.DATE>2002-09-11</ARRIVAL.DATE>
<ARRIVAL.TIME>1100</ARRIVAL.TIME>
<BERTHING.DATE>2002-09-11</BERTHING.DATE>
<BERTHING.TIME>1230</BERTHING.TIME>
</ARRIVAL.NOTICE>
```

Figure 4.9: Arrival Notice XML Document

```

<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT ARRIVAL.NOTICE
(VESSEL.NAME, VOYAGE.NO, LOAD.PORT, DISCHARGE.PORT, ARRIVAL.DATE, AR
RIVAL.TIME, BERTHING.DATE, BERTHING.TIME)>

<!ELEMENT VESSEL.NAME (#PCDATA)>
<!ATTLIST VESSEL.NAME VESSEL.CODE CDATA #REQUIRED>
<!ELEMENT VOYAGE.NO (#PCDATA)>
<!ELEMENT LOAD.PORT (#PCDATA)>
<!ELEMENT DISCHARGE.PORT (#PCDATA)>
<!ELEMENT ARRIVAL.DATE (#PCDATA)>
<!ELEMENT ARRIVAL.TIME (#PCDATA)>
<!ELEMENT BERTHING.DATE (#PCDATA)>
<!ELEMENT BERTHING.TIME (#PCDATA)>

```

Figure 4.10: Arrival Notice XML Document DTD

Table 4.4: Explanation of Ship berthing and container location document

Element	Attribute	Description
PORT.CONTAINER.POSITION		Root element
SHIPPING.COMPANY		
VESSEL.NAME		
VOYAGE.NO		
DISCHARGE.PORT		
ARRIVAL.DATE		
ARRIVAL.TIME		
BERTHING.DATE		
BERTHING.TIME		
CONTAINER.DETAILS		Group element
CONTAINER.NO	TYPE, SIZE	
CONTAINER.YARD.LOCATION		
OFFLOAD.DATE		

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE PORT.CONTAINER.POSITION SYSTEM "file://c:\PORT.CNTR.DTD">
<PORT.CONTAINER.POSITION>
  <SHIPPING.COMPANY>THE SCI LTD MUMBAI</SHIPPING.COMPANY>
  <VESSEL.NAME>TAJ MAHAL</VESSEL.NAME>
  <VOYAGE.NO>786</VOYAGE.NO>
  <LOAD.PORT>CPH</LOAD.PORT>
  <DISCHARGE.PORT>MUM</DISCHARGE.PORT>
  <ARRIVAL.DATE>2002-09-11</ARRIVAL.DATE>
  <ARRIVAL.TIME>1100</ARRIVAL.TIME>
  <BERTHING.DATE>2002-09-11</BERTHING.DATE>
  <BERTHING.TIME>1230</BERTHING.TIME>
  <CONTAINER.DETAILS>
    <CONTAINER.NO TYPE="GT"
SIZE="20">SCIU1234560</CONTAINER.NO>
<CONTAINER.YARD.LOCATION>MLD12</CONTAINER.YARD.LOCATION>
  <OFFLOAD.DATE>2002-09-12</OFFLOAD.DATE>
  </CONTAINER.DETAILS>
  <CONTAINER.DETAILS>
    <CONTAINER.NO TYPE="RF"
SIZE="40">CTIU1234595</CONTAINER.NO>

    <CONTAINER.YARD.LOCATION>CHB44</CONTAINER.YARD.LOCATION>
    <OFFLOAD.DATE>2002-09-12</OFFLOAD.DATE>
  </CONTAINER.DETAILS>
  <CONTAINER.DETAILS>
    <CONTAINER.NO TYPE="GT"
SIZE="40">WMIU2103202</CONTAINER.NO>
<CONTAINER.YARD.LOCATION>MLD12</CONTAINER.YARD.LOCATION>
  <OFFLOAD.DATE>2002-09-13</OFFLOAD.DATE>
  </CONTAINER.DETAILS>
</PORT.CONTAINER.POSITION>

```

Figure 4.11: Berthing and Container Location XML Document

```

<?xml version="1.0" encoding="UTF-8"?>
<!--ELEMENT PORT.CONTAINER.POSITION
(SHIPPING.COMPANY,VESSEL.NAME,VOYAGE.NO,LOAD.PORT,DISCHARGE.POR
T,ARRIVAL.DATE,ARRIVAL.TIME,BERTHING.DATE,BERTHING.TIME,(CONTAIN
ER.DETAILS)+)-->
<!--ELEMENT CONTAINER.DETAILS
(CONTAINER.NO,CONTAINER.YARD.LOCATION,OFFLOAD.DATE)-->

<!--ELEMENT SHIPPING.COMPANY (#PCDATA)-->
<!--ELEMENT VESSEL.NAME (#PCDATA)-->
<!--ELEMENT VOYAGE.NO (#PCDATA)-->
<!--ELEMENT LOAD.PORT (#PCDATA)-->
<!--ELEMENT DISCHARGE.PORT (#PCDATA)-->
<!--ELEMENT ARRIVAL.DATE (#PCDATA)-->
<!--ELEMENT ARRIVAL.TIME (#PCDATA)-->
<!--ELEMENT BERTHING.DATE (#PCDATA)-->
<!--ELEMENT BERTHING.TIME (#PCDATA)-->
<!--ELEMENT CONTAINER.NO (#PCDATA)-->
<!--ATTLIST CONTAINER.NO
      TYPE CDATA #REQUIRED
      SIZE CDATA #REQUIRED-->
<!--ELEMENT CONTAINER.YARD.LOCATION (#PCDATA)-->
<!--ELEMENT OFFLOAD.DATE (#PCDATA)-->

```

Figure 4.12: Berthing and Container Location XML Document DTD

4.5.6 Computer programs required

The following four computer programs are required to exchange XML documents between the shipping company and consignees:

1. Create XML document

This program is used for preparing an XML document from the database. The shipping company will use this program to generate the XML document, “Container Delivery Notice”. As XML does not have any database manipulation capabilities, the XML document can be dynamically created by using Internet programming languages (Vaswani, 2002).

2. Send XML document

This program sends the document via the Internet to consignees. It checks the XML document is well formed and valid. It encrypts the document, logs in to the consignee's computer and sends the XML documents to consignees.

3. Receive XML document

This program receives the XML document, "Container Delivery Notice", via the Internet from the shipping company. It decrypts the document and checks the XML document is well formed and valid. It passes on the details of the XML document to the application server of the consignee.

4. Process XML document

This program processes the details of the XML document and it sends the data to the database server.

4.6 Benefits and limitations of XML

4.6.1 Benefits of XML

- **Saving in infrastructure investment**

The XML format is independent of platform for exchanging information between computers. This allows the users to transact their business with anyone using XML standards. The users need not invest in any special platform. They have the choice of selection of platform and they can continue to use this existing platform. This saves the investment in new infrastructure.

- **Saving in manpower**

The XML data can be directly stored in the computer database without reentering it again. This saves retyping time and avoids typing errors and, saves the manpower to reenter data.

- **Reduction in paperwork**

Documents can be transferred in XML format, which avoids the need to print the document. Since the data is directly transferred, paperwork is not required. This reduces the paper work.

- **Enhances the decision making**

XML data can be made available anywhere, anytime, which enhances the capability of ship managers to get the right information at the right time. This saves their time in getting the information and decisions can be made effectively for better ship operations and planning.

- **Better customer service**

Shipping companies can offer better customer service by delivering the necessary information to their customers, which therefore can avail themselves of the right services of shipping companies. Moreover, shipping companies can decide their CRM (Customer Relationship Management) strategy in an effective way.

- **Competitive advantage**

XML gives leverage to shipping companies in managing the information efficiently and effectively. It also helps in managing their relations with suppliers and customers. This gives shipping companies the competitive advantage in retaining their customers and market share.

- **Can be implemented by shipping companies of any size**

Shipping companies of any size can implement XML based systems for exchanging information since the investment in infrastructure is relatively low and XML based systems can be implemented faster.

- **Multi-linguistic delivery of content and data**

Due to the global nature of shipping, shipping companies have to transact business with shippers, clients and suppliers of many countries using different languages. Since XML data can be stored in any language, shipping companies can exchange data in any language, which gives them the advantage of managing business effectively and efficiently.

4.6.2 Limitations of XML

- **Standard vocabulary**

XML defines standards and rules to describe the data but it does not specify the vocabulary required to describe the data. Shipping companies will have to agree on the common vocabulary before implementing XML-based systems. If they use different names to describe the same data, it will be difficult to exchange XML documents.

Document type definitions (DTDs) and XML Schema both need to be set up and agreed globally to allow XML to be used really efficiently. The answer to this is ebXML (<http://www.ebxml.org/>), which is being developed jointly by UN/CEFACT (United Nations Center for Trade Facilitation and Electronic Business, <http://www.unece.org/cefact/>) and OASIS (<http://www.oasis-open.org/>). The main focus of UN/CEFACT is the worldwide facilitation of international trade, through simplification and harmonization of procedure and information flows. OASIS is an international non-profit consortium that designs and develops industry standard specifications for interpretability based on XML. The objective is “to provide an open XML-based infrastructure enabling the global use of electronic business information in an interpretable, secure and consistent manner by all parties” (Power, 2001). Marine Trading Markup Language (MTML), an XML-based language is released for the electronic exchange of purchasing information in the marine industry. It is derived from and compatible with the International Marine Purchasing

Associations' (IMPA) Electronic Trading Standard Format (ETSF) (<http://www.meca.org.uk/>). Shipping companies can use this e-business standard for conducting e-business with their suppliers.

- **Few applications available**

XML standards have been around since 1998 and they are continuously being developed. However, there are hardly any shipping companies, which have implemented XML-based systems yet. Recently APL has announced that it will implement EDI/XML solutions (Mercator, 2002). APL provides container transportation services to customers around the world (http://www.apl.com/html/company_profile.html).

- **Security of information**

Security is one of the important requirements of e-business. XML Security standards are under development by the W3C and the Internet Engineering Task Force (IETF), jointly. VeriSign and other vendors have already started introducing the security standard for XML (Marshall, 2001). IBM and Microsoft have jointly proposed a security model for Web services by considering XML technologies (VeriSign, 2002). It will however, take some time to establish these standards and to incorporate appropriate security into the application packages.

- **Legal implication**

Electronic law is still not in force in many countries. The US has announced the implementation of electronic law. Other countries may follow the suit but shipping companies will still face the problem of legality of data exchanged electronically. Tools can be used to verify the integrity of data and the identity of the sender, but due to lack of a legal framework support system for electronic law, shipping companies are hesitant to do e-business.

- **Development of standards is slow**

The XML standards are being developed by the W3C but the process of development of these standards is slow and it takes time to recommend and implement those standards. It affects the development of systems and it takes time for the users to accept new standards since they will have to change their systems accordingly.

- **Many standards**

XML consists of a group of standards. Some are recommended while others are in the development stage. There seem to be too many standards. Some overlapping in standards takes place when changing from the old standard to a new standard. For example, DTD or XML Schema can both be used to validate XML documents. This creates confusion in the users while implementing the XML based systems.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

- **XML is better than HTML**

XML and HTML are developed from the common markup language, SGML. A comparison between XML and HTML shows that XML is more suitable for data exchange and XML supports many international languages. Further, XML has a robust linking mechanism. The document search in XML can be done on a specific item for displaying the information. An XML document can be reused. Thus, XML provides better features of data exchange and content management than HTML

- **Easy to prepare XML document**

XML describes the information in simple and easy to understand language. It allows the users to define elements and attributes within an XML document. This gives the users the freedom to present the information in their own terminology. The DTD and XML Schema, though not mandatory, ensure the data are valid and satisfy XML standards. There are XML based standards like ebXML, Bolero-XML and MTML, which are useful in preparing XML documents. Thus, shipping companies can use these standards for exchanging data with their business partners.

- **Data portability**

XML is an open standard and can be used on any platform. This offers the flexibility to implement XML based systems for exchanging data between computers. It is now

easier to transfer data in XML format rather than to depend upon proprietary formats. Thus, XML offers portability of data for exchanging information over the Internet.

- **EDI and XML work together**

The comparison between EDI and XML shows that XML enables the integration of data and document. XML can be implemented by a shipping companies of any size and it can be used for online transactions processing. XML supports almost any international language. There are ample tools and support available for XML based systems. Moreover, XML systems can be implemented in less time and cost. Thus, XML based systems are better than the traditional EDI. However, EDI and XML can work together by using EDI practices and XML data standards.

- **Easy to implement**

The underlying technology for implementing XML-based systems is the Internet. XML uses Internet protocols like HTTP and SMTP to exchange information. Shipping companies are already using the Internet for e-mail and Web based applications. Further, Internet programming languages are now-a-days used to develop software for Web-based systems. These programming languages can also be used for developing the XML based systems. It makes the task of implementing XML based systems easier for shipping companies.

- **Easy to maintain and support**

The entire XML specification is open and is driven by the user communities. The XML standard is not monopolized by any person or commercial entity. The complete specification for XML is available to the public at the website of the World Wide Web Consortium (W3C), <http://www.w3.org/XML>. There are skills and tools readily available in the market for Internet based applications. Therefore, shipping companies can maintain and support their XML based systems.

- **More scope in E-business and E-commerce**

E-business and E-commerce are necessary in the present business environment. They give competitive advantages to shipping companies in controlling costs and expanding their customer base and market share. XML is the best tool for developing E-business and E-commerce applications for shipping companies, as XML offers the data portability on any computer platform.

- **Applications in shipping companies**

Shipping companies can use XML and its related technologies for developing various kinds of applications. For instance, some of these applications are document management systems, E-Business, E-commerce, Channel Definition Format (CDF), Synchronized Multimedia Integration Language (SMIL) and Web services. These applications are useful in saving infrastructure cost and manpower cost, in reducing paperwork, for making effective decisions and for offering better customer service. These applications give a competitive advantage to shipping companies in retaining and extending the customer base and the market share. Moreover, shipping companies of any size can implement the XML based applications.

- **Technology of next generation**

XML is the technology of the future since all the development in Information Technology (IT) is being developed around XML. XML has very wide support of the user communities and major IT companies like Microsoft, IBM, Sun Microsystems, and Oracle. Furthermore, standards of XML are open to the public, which makes XML technologies available and accessible for everyone. This is driving the development in IT quite fast. New databases, protocols, languages and tools are being developed. It is a healthy sign and shipping companies can take advantage of new XML technologies in cutting their costs and meeting their customer expectations.

- **Limitations of XML**

Any new technology has its own limitations and XML is not an exception in this respect. XML only defines the standard and rules to describe the data but the users have to develop their own terminology to describe the data. Since XML is an over four-years old technology, it takes time to adapt these standards to the requirements of the users. Standards have to be continuously developed to meet the requirements and it takes time for these standards to mature. The security and legality of data in E-commerce and E-business are concerns of shipping companies. But no system can be made 100% secure without incurring cost in proportion. It is true for any IT systems. The security standards for XML digital signatures were already developed in February, 2002 and the standard for XML Encryption is currently being developed. However, the present standards can also be used for transacting e-business and e-commerce. Thus, the advantages of XML outweigh its limitations.

5.2 Recommendations

- **Education and Training**

XML being a new technology, shipping companies should educate their employees about the features and advantages of XML so that the users can appreciate the XML technology and be able to participate in the implementation process. IT staff are the backbone for developing the required software and maintaining the XML applications. Shipping companies should train their IT staff in developing XML based systems. Identifying the right products and right technology partner for implementing XML based applications is very important.

- **Use XML editor tools to prepare XML documents**

Though XML documents and DTD or XML Schema can be prepared by using any text editor like Notepad, it is recommended that the users should use an XML editor tool for preparing XML documents and DTD or XML Schema. This will help the users to prepare XML documents easily and it will eliminate many mistakes. Further,

the users can test XML documents for their validity and correctness. There are many XML tools available for free in the market as well as on a commercial basis. The users can decide to use any one of the XML editor tools. Initially, it will take time to learn these tools. However, before using any XML editor tool, the users should have the full knowledge of XML specifications so that they can use XML editor tools effectively.

- **Implementation in phased manner**

Shipping companies can initially start with a small XML based system and learn from the experience. This will help in identifying the actual requirements and the pitfalls of technology so that in the next phase of implementation of XML solutions past mistakes can be avoided. Shipping companies can start with converting their existing HTML based web site into an XML based web site.

- **Replacement of EDI**

EDI consists of exchange of data in electronic format such as the proprietary formats, EDIFACT or ANSI X12. This exchange format can be replaced with the XML format without changing the implementation platform. The private network or Value Added Network (VAN) can be replaced with the Internet network. Shipping companies will require XML interfaces to link the database of a legacy system and the XML format. These interfaces can be developed in any Internet programming language. This will save communication costs and shipping companies can set up EDI links with any business partner. The XML messages can be transferred by e-mail or HTTP.

- **XML compliant products**

IT companies have already developed various XML compliant tools and databases. These tools can be used to develop XML documents and the applications programs. There are many free products for developing XML applications, which can be downloaded from the Internet. Shipping companies should select a database, which

supports both the legacy and XML formats. This will eliminate the need for XML interfaces and the data can be transferred from computer-to-computer without the need for interfaces.

- **Security and legality of data**

The present security products can be used to encrypt the data and to authenticate the users. Shipping companies can choose these products depending upon their requirements and budget. IT companies are continuously developing new products to meet the new standards and shipping companies can decide to implement new security products. The legality of data is a concern for any e-business and shipping companies have to agree on the legal aspects. Electronic laws are still not implemented in many countries and hence the enforcement of legal contracts based on electronic business may become difficult. Shipping companies should therefore have mutual agreements with their counterparts for E-business.

- **Development of XML standards**

As XML based applications become popular and the users demand more features, XML standards and the related technologies will continue to develop. The development in standards must consider the new requirements and shipping companies need to closely monitor the new standards. They need to upgrade and to develop new systems based on the new standards so that the benefits of the technologies can be realized.

REFERENCES

- Acrobat. (2002). Retrieved June 30, 2002 from the World Wide Web:
<http://www.adobe.com/products/acrobat/adobepdf.html>
- APL. (2002). Retrieved August 16, 2002 from the World Wide Web:
http://www.apl.com/html/company_profile.html
- Asay, A. (2000). Bolero.net represents an opportunity to grasp the digital trade revolution.. *Bimco Review 2000* (pp. 293-294). London: Book Production Consultants Plc.
- Atkinson, H. (2001). The perseverance of EDI. (June 4-10, 2001). *JoC Week*, 26-28.
- Bolero. (2002a). Retrieved June 22, 2002 from the World Wide Web:
<http://www.bolero.net/>
- Bolero. (2002b). Introduction to Bolero-XML
Retrieved August 5, 2002 from the World Wide Web:
<http://www.bolero.net/boleroxml/introduction/>
- Bruzek, S. (2002). Oracle XML DB: Unite Content & Data
Retrieved August 4, 2002 from the World Wide Web:
<http://www.oracle.com/iseminars/archive/index.html>
- Cerami, E. (2002). Top ten FAQ for Web Services.
Retrieved July 29, 2002 from the World Wide Web:
<http://www.oreillynet.com/pub/a/webservices/2002/02/12/webservicefaqs.html>
- Cocoon (2002). Retrieved August 3, 2002 from the World Wide Web:
<http://xml.apache.org/cocoon/introduction.html>
- dbXML (2002). Retrieved August 3, 2002 from the World Wide Web:
<http://www.dbxml.com/index.xml>
- DigitalShip. (2002). *Container shipping technology*.
Retrieved August 14, 2002 from the World Wide Web:
<http://www.thedigitalship.com/subscriptions.htm>
- Drewry Shipping Consultants (2000). *I.T. and shipping : new technology and new thinking leading to commercial advantage*. London : Author

- ebXML (2002).
Retrieved August 2, 2002 from the World Wide Web:
<http://www.ebxml.org/>
- Epic editor.
Retrieved June 30, 2002 from the World Wide Web:
http://www.arbortext.com/html/epic_editor_overview.html
- ESCAP (1993). *ESCAP/UNDP Manual on EDI in transport*. New York:UN
- Fairplay (1999). *Marine Computing & Internet Guide 2000*. Coulsdon:Author
- FrameMaker 7.0.
Retrieved June 30, 2002 from the World Wide Web:
<http://www.adobe.com/>
- Forsyth, G (2000). XML breaking down IT barriers in logistics. (2000, June).
American Shipper , 20-29.
- Gerson, G.M. (1989). *Data Mapping – Integration of EDI into the Corporate Information Structure*. Alexandria: TDCC
- Google. (2002).
Retrieved August 11, 2002 from the World Wide Web:
<http://www.google.com/search?hl=en&ie=UTF-8&oe=UTF-8&q=xml&btnG=Google+Search>
- IBM. (2002). Lotus Notes & Domino.
Retrieved June 22, 2002 from the World Wide Web:
<http://www.lotus.com/products/r5web.nsf/webhome/nr5noteshp>
- Jayakody, S.W.J. (1994). *EDI System for Shipping Organisations*.
Unpublished master's thesis, World Maritime University, Malmö, Sweden.
- Kayl, K. (2001). *Electronic Business XML: Revoltionizing the engines of business*
Retrieved August 11, 2002 from the World Wide Web:
<http://java.sun.com/features/2001/09/ebxmlrev.html>
- Kearns, D. (2002). *XML no Magic Bullet*.
Retrieved July 27, 2002 from the World Wide Web:
<http://www.nwfusion.com/newsletters/dir/2002/01463136.html>

- Kim, H.S & others. (2001a). Development of XML based EDI system for export customs clearance. *The International Association of Maritime Economists Annual Conference, 2001, 18-20 July 2001, Hong Kong*, (pp 451-462). Hong Kong : Hong Kong Polytechnic University.
- Laurent, S. & Cerami, E. (1999). *Building XML Applications*. New York: McGraw-Hill.
- Levitt, J (2000). *The Language of E-Business*.
Retrieved February 11, 2002 from the World Wide Web:
<http://www.informationweek.com/817/prxml.htm>
- Linn, G. (2001). Alternative to EDI. (2001, May 14-20). *JoC Week*, 34-35.
- Luoma, R (2001). *Using XML to enable low-cost deployment of contents at Lockheed Martin Aeronautics*. Retrieved February 11, 2002 from the World Wide Web:
<http://www.idealliance.org/papers/xml2001papers/tm/WEB/04-02-04/04-02-04.htm>
- Mactaggart, M. (2001). An introduction to XML encryption and XML signature.
Retrieved July 29, 2002 from the World Wide Web:
<http://www-106.ibm.com/developerworks/xml/library/s-xmlsec.html/index.html>
- Marchal, B. (2001). *Applied XML solutions*. New Delhi: Techmedia.
- Marshall, M. (2001). *XML-Like The Air That We Breathe*.
Retrieved February 11, 2002 from the World Wide Web:
<http://www.informationweek.com/827/prxml.htm>
- McGrath, S. (1998). *XML by Example : Building E-Commerce Application*. New Jersey: Prentice Hall.
- MECA. (2002). Maritime e-Commerce Association.
Retrieved August 2, 2002 from the World Wide Web:
<http://www.meca.org.uk/>
- Mercator. (2002). Mercator selected by APL to provide EDI/XML translation
Retrieved August 16, 2002 from the World Wide Web:
<http://www.mercator.com/default.asp?feature=press&topic=press&pr=252>
- Microsoft. (2002). Working with Channels
Retrieved July 28, 2002 from the World Wide Web:

<http://www.microsoft.com/windows2000/en/datacenter/help/default.asp?url=/windows2000/en/datacenter/help/cdf.htm>

Morrison, M. (2000). *XML unleashed*. New Delhi: Techmedia.

North, S. & Hermans, P. (2002). *Teach yourself XML in 21 days*. New Delhi: Techmedia.

OASIS. (2002).
Retrieved August 2, 2002 from the World Wide Web:
<http://www.oasis-open.org/>

Oracle. (2002). Oracle XML DB.
Retrieved August 4, 2002 from the World Wide Web:
http://technet.oracle.com/tech/xml/xmlldb/pdf/xmlldb_92twp.pdf

Ozone. (2002).
Retrieved August 3, 2002 from the World Wide Web:
http://www.ozone-db.org/ozone_main.html

Pfaffenberger, B. (1999). *Web publishing with XML in 6 easy steps*.
California: AP Professional.

Pierce, R. (2002). XML spells the end for EDI...eventually. (2002, July/August).
Information Management & Technology, 171-172.

Power, T. (2001). Standards for containers. (2001, November). *Digital Ship*, 5.

Powers, W.J. (1989). *EDI control & audit issues for managers, users & auditors*.
Alexandria: TDCC.

RosettaNet. (2002). Retrieved August 17, 2002 from the World Wide Web:
<http://www.rosettanet.org/rosettanet/Rooms/DisplayPages/LayoutInitial>

Ruben, P. (2001). *J2EE and .net: Two Roads Diverge in XML*
Retrieved February 11, 2002 from the World Wide Web:
<http://www.internetnews.com/asp-news>

Samaranayake, W.W.S.R. (1997). *The Impact of Information Technology on Shipping*. Unpublished master's thesis, World Maritime University, Malmö, Sweden.

Shipping lines launch XML standard for bookings. (2001, April). *DigitalShip*. 1.

- Shmeitzer, S. (2002). *Internet Seminar: Developing XML Powered Applications*. Retrieved August 4, 2002 from the World Wide Web:
<http://www.oracle.com/iseminars/archive/index.html>
- Siman, E., Madsen, P., & Adams, C. (2001). An introduction to XML signature. Retrieved July 29, 2002 from the World Wide Web:
<http://www.xml.com/pub/a/2001/08/08/xmlsig.html>
- SoftQuad. (2002). XML Solutions for E-business. Retrieved June 30, 2002 from the World Wide Web:
http://www.softquad.com/top_frame.sq
- Stylus Studio. Retrieved June 30, 2002 from the World Wide Web:
<http://www.exln.com/products/stylusstudio/>
- Tittel, E. & Boumphrey, F. (2000). *XML for Dummies* (2nd ed.). California: IDG books.
- Tittel, E. , Pitts, N. & Boumphrey, F. (2002). *XML for Dummies* (3rd ed.). New York: Hungry Minds.
- Turk, A. (2000). The WebConsultant – A Flexible Framework for Dynamic Web Applications. *Electronic Commerce and Web Technologies. First International Conference, EC-Web 2000 , 4-6 September 2000, London*, (pp 13-24). London: Springer.
- UN/CEFACT. (2002). United Nations Center for Trade Facilitation and Electronic Business. Retrieved August 2, 2002 from the World Wide Web:
<http://www.unece.org/cefact/>
- UNICODE. (2002). Retrieved June 29, 2002 from the World Wide Web:
<http://www.unicode.org/>
- Vaswani, V. (2002). *Dynamically Generating XML from a Database*. Retrieved August 3, 2002 from the World Wide Web:
http://www.informit.com/content/index.asp?product_id={AFFDEDB8-4C56-460B-A2E1-B47D0ABD77AF}&session_id={92BBB199-C872-46C8-BADD-7F1555DE6938}
- Verboom, J. & Jellema, J. (2001). *Ten best bets for XML Applications*. Retrieved February 11, 2002 from the World Wide Web:
<http://www.oneworld.orf/thinktank/iktools/edit4.html>

- VeriSign. (2002). Security in a Web Service World: A Proposed Architecture and Roadmap. Retrieved August 5, 2002 from the World Wide Web:
<http://www.verisign.com/wss/architectureRoadmap.pdf>
- Vervet. (2002). XML Pro.
Retrieved June 30, 2002 from the World Wide Web:
<http://www.vervet.com/>
- Vollmer, K. (2001). *Don't believe the Hype: EDI and XML are just perfect together*.
Retrieved February 11, 2002 from the World Wide Web:
<http://www.internetweek.com/columns01/beat011501.htm>
- Votsch, V. & Walter, M. (2002). *Oracle XML DB: Uniting XML Content & Data*.
Retrieved August 4, 2002 from the World Wide Web:
http://technet.oracle.com/tech/xml/xmlldb/pdf/xmlldb_buswp.pdf
- World Wide Web Consortium. (2002).
Retrieved July 17, 2002 from the World Wide Web:
<http://www.w3.org/>
- W3C. (2002a). Amaya – W3C's editor and web browser.
Retrieved July 17, 2002 from the World Wide Web:
<http://www.w3.org/Amaya/>
- W3C. (2002b). Channel Definition Format (CDF).
Retrieved July 28, 2002 from the World Wide Web:
<http://www.w3.org/TR/NOTE-CDFsubmit.html>
- W3C. (2002c). Document object model.
Retrieved July 4, 2002 from the World Wide Web:
<http://www.w3.org/DOM/>
- W3C. (2002d). Mathematical Markup Language (MathML).
Retrieved July 28, 2002 from the World Wide Web:
<http://www.w3.org/Math/>
- W3C. (2002e). Namespaces in XML
Retrieved July 5, 2002 from the World Wide Web:
<http://www.w3.org/TR/REC-xml-names/>
- W3C. (2002f). Soap (simple object Access Protocol).
Retrieved July 28, 2002 from the World Wide Web:
<http://www.w3.org/TR/2002/WD-soap12-part1-20020626/>

- W3C. (2002g). Synchronized Multimedia Integration Language (SMIL).
Retrieved July 28, 2002 from the World Wide Web:
<http://www.w3.org/TR/smil20/>
- W3C. (2002h). XML. Retrieved July 5, 2002 from the World Wide Web:
<http://www.w3.org/XML>
- W3C. (2002i). XML linking. Retrieved July 5, 2002 from the World Wide Web:
<http://www.w3.org/XML/linking>.
- W3C. (2002j). XML Path language (XPath). Retrieved July 5, 2002 from the World Wide Web:
<http://www.w3.org/TR/xpath>.
- W3C. (2002k). XML Pointer language (XPointer).
Retrieved July 9, 2002 from the World Wide Web:
<http://www.w3.org/TR/xptr/>.
- W3C. (2002l). XML Schema. Retrieved July 15, 2002 from the World Wide Web:
<http://www.w3.org/XML/Schema>
- W3C. (2002m). XML Schema Datatypes.
Retrieved July 15, 2002 from the World Wide Web:
<http://www.w3.org/TR/xmlschema-2/>
- W3C. (2002n). XML Schema Structure.
Retrieved July 15, 2002 from the World Wide Web:
<http://www.w3.org/TR/xmlschema-1/>
- W3C. (2002o). XML-Signature Syntax and Processing.
Retrieved August 4, 2002 from the World Wide Web:
<http://www.w3.org/TR/xmlsig-core/>
- W3C. (2002p). XSL. Retrieved July 5, 2002 from the World Wide Web:
<http://www.w3.org/style/XSL>.
- W3C. (2002q). XSLT. Retrieved July 5, 2002 from the World Wide Web:
<http://www.w3.org/TR/xslt>.
- W3Cschoools. (2002). XML Tutorial. Retrieved August 11, 2002 from the World Wide Web: <http://www.w3schools.com/xml/default.asp>
- XMLSpy. (2002). Retrieved June 30, 2002 from the World Wide Web:
<http://www.xmlspy.com/>

SELECTED BIBLIOGRAPHY

Boumphrey, F., Dorenzo, O., Duckett, J., Graf, J., Hollander, D., Houle, P., Jenkins, T., Jones, P., Kingsley-Hughes, A., Kingsley-Hughes, K., Mcqueen, C., & Mohr, S. (1998). *XML Applications*. Birmingham: Wrox.

Extensible Markup Language (XML).
Retrieved July 15, 2002 from the World Wide Web:
<http://www.textuality.com/xml/>

Extensible Markup Language (XML).
Retrieved June 30, 2002 from the World Wide Web:
<http://xml.coverpages.org/xml.html>

FileMaker Pro.
Retrieved July 30, 2002 from the World Wide Web:
<http://www.filemaker.com/>

Finkelstein, C. (2000). XML for Enterprise Application Integration.
Retrieved July 29, 2002 from the World Wide Web:
[http://www.planetmarkup.com/xmlarena/xap/Friday/EnterpriseApplicationInt
egration.pdf](http://www.planetmarkup.com/xmlarena/xap/Friday/EnterpriseApplicationIntegration.pdf)

Flynn, P. (2002). The XML FAQ.
Retrieved June 22, 2002 from the World Wide Web:
<http://www.ucc.ie/xml/>

Gonzalez, J.C.S.M. (1998). *The Role of Information Technology in Shipping Companies at the end of the 90's.* Unpublished master's thesis, World Maritime University, Malmö, Sweden.

Java Technology & XML.
Retrieved July 20, 2002 from the World Wide Web:
<http://java.sun.com/xml/resources.html>

Kim, H.S & others. (2001b). Development of XML/EDI based data warehouse for customs clearance of maritime exports. *The International Association of Maritime Economists Annual Conference, 2001, 18-20 July 2001, Hong Kong*, (pp 468-480). Hong Kong : Hong Kong Polytechnic University.

Holzner, S.. (1998). *XML complete*. New York: McGraw-Hill.

- Marine Trading Markup Language.
Retrieved July 28, 2002 from the World Wide Web:
<http://www.oasis-open.org/cover/mtml.html>
- Melander, J. & Inki, J. (2000). *XML and its Applications on Software Documentation*. Unpublished master's thesis, Lund Institute of Technology, Lund, Sweden.
- Myerson, J.M. (2002). MS Access Speaks XML.
Retrieved August 3, 2002 from the World Wide Web:
http://www.webreview.com/2002/02_04/developers/index01.shtml
- Ogbuji, U. (2002). XML Messaging with SOAP.
Retrieved August 1, 2002 from the World Wide Web:
<http://www-105.ibm.com/developerworks/education.nsf/xml-onlinecourse-bytitle/52CEEC8554DB0C1586256A0D005A6E41?OpenDocument>
- O'Reilly's XML. Retrieved July 12, 2002 from the World Wide Web:
<http://www.xml.com/>
- Organisation for Economic Co-operation and Development (1999). *The Economic and Social Impact of Electronic Commerce*. Geneva: Author.
- Patrizio, A. (2001). XML passes from development to implementation.
Retrieved February 11, 2002 from the World Wide Web:
<http://www.informationweek.com/shared/printArticle?article=info.../xml.htm&pub=iw>.
- PerfectXML. (2002). XML Standard References.
Retrieved August 11, 2002 from the World Wide Web:
<http://www.perfectxml.com/articles/xml/stan.asp>
- Savourel, Y. (2001). *XML Internationalization & Localization*. New Delhi: Techmedia.
- Scardina, M. (2002). *Internet Seminar: Understanding the Oracle9i XML Infrastructure By Example*.
Retrieved August 4, 2002 from the World Wide Web:
<http://www.oracle.com/iseminars/archive/index.html>
- VeriSign. (2002). XML Trust Services.
Retrieved July 29, 2002 from the World Wide Web:
<http://www.verisign.com/developer/xml/>

Waxberg, L. (2001). *Pre-ordering System using XML*. Unpublished master's thesis, Lund Institute of Technology, Lund, Sweden.

XML. Retrieved June 15, 2002 from the World Wide Web:
<http://www.xmlx.com/lander/index2.php?name=xmlx.com&term=xmlx>

XML. Retrieved August 11, 2002 from the World Wide Web:
<http://www.perfectxml.com/xml.asp>.

XML applications. Retrieved June 29, 2002 from the World Wide Web:
<http://www.finetuning.com/>

XML : Education : Tutorials. Retrieved July 30, 2002 from the World Wide Web:
<http://www-105.ibm.com/developerworks/education.nsf/dw/xml-onlinecourse-bynewest?OpenDocument&Count=500>

XML Europe 2002. Retrieved July 28, 2002 from the World Wide Web:
<http://www.xml europe.com/>

XML : FAQ. Retrieved July 22, 2002 from the World Wide Web:
<http://www.xml.org/xml/xmlfaq.shtml>.

XML files. Retrieved June 20, 2002 from the World Wide Web:
<http://webdeveloper.com/xml/>

XML for Absolute Beginners. Retrieved August 11, 2002 from the World Wide Web: <http://www.perfectxml.com/articles/xml/begin.asp>

XML Industry Portal. Retrieved July 28, 2002 from the World Wide Web:
<http://lists.xml.org/>

XML Information. Retrieved June 11, 2002 from the World Wide Web:
<http://www.xmlinfo.com/>

XML Signature An introduction to. Retrieved July 29, 2002 from the World Wide Web: <http://www.xml.com/pub/a/2001/08/08/xmlsig.html>

XML Software. Retrieved June 18, 2002 from the World Wide Web:
<http://www.xmlsoftware.com/>

XML Standard References. Retrieved August 11, 2002 from the World Wide Web:
<http://www.perfectxml.com/articles/xml/stan.asp>

XML Trust Services. Retrieved July 29, 2002 from the World Wide Web:
<http://www.verisign.com/developer/xml/>

Questionnaire on Application of XML for Shipping Companies

Dear Sir/Madam

I am a student of World Maritime University, Malmö , Sweden. I am doing my research project on application of Extensive Markup Language (XML) for shipping companies as a part of my Post-graduate study for the year 2002.

I. Introduction

The main purpose of the research project is to find out the ways to save the cost, time and resources of shipping companies by using the new Internet technology called XML. Now a days Internet and email are extremely popular in shipping companies. These computer technologies are used for day-to-day business operation and decision-making. In fact, these technologies have almost become indispensable.

However, these technologies do not always allow the exchange of information in the day-to-day business operation. For example, a web page of any website can be viewed by using the web browser like Internet explorer or Netscape navigator. But it does not allow exchanging of data in a required format from one PC to any PC in another location. It requires special software or the assistance of a software engineer to write the software. This is a cumbersome and time-consuming process in today's competitive world. We need the technology, which is user friendly and can be used by ourselves like the Internet browser or word processor. We should be able to exchange the data from one PC to any PC in another location in required business format.

XML technology solves the problem of exchanging data in a required business format between two or more PCs at different locations. This technology is being used in industries like health, defence, finance, chemical, etc. The efforts are

going on to develop a standard Marine Text Markup language (MTML) based on XML for shipping industry.

In order to explore the various areas of shipping which can use the XML technology, I need the following information about functioning of your organisation and your expectations from the information technology. Kindly spare your valuable time to supply the information. The information will be kept confidential and will be used solely for research project only.

II. Company Profile:

1. Name of your organisation:
2. Business address:
3. Telephone No.
4. Fax No.
5. Email address:
6. Website address:
7. Annual sales turnover:
8. Number of employees:
9. Number of offices:
10. Number of ships owned if shipowner:
11. Business of your organisation:
12. Please mention business activities of your organisation.

III. Data Communication:

13. Do you use EDI in your organisation? Yes/No
If yes, please mention uses of EDI in your organisation.
14. Do you use Internet in your organisation? Yes/No
If yes, please mention purpose of using Internet in your organisation:
15. Please mention the business information generally communicated by email:

16. Are you exchanging any business data from your PC to any PC in another office or business contact? Yes/no

If yes, what kind of business data is directly exchanged by you?

17. Are you accessing any website for online business with another office or business contacts? Yes/No

If Yes, please mention what kind of business are you doing online?

IV. Application software used:

18. Please mention the kind of documents prepared on computer in your organisation.

19. Please give a list of application software used by your functional departments like operation, purchase, finance, etc. These software may be developed or purchased from outside.

20. Please mention benefits in using existing application software in your organisation.

21. Please mention issues and challenges faced in using existing application software in your organisation.

V. Resources used:

22. What is your annual total cost on communication and correspondence:

Please give the total communication cost under following categories:

Mode of Communication	Annual cost	% of total cost
Telephone		
Fax		
Telex		
Email		
Internet		
Documentation		

23. Please mention how is your organisation spending time on the communication and correspondence:

Mode of Communication	% of total communication time spent	% Time spent on decision making	% Time spent on day-to-day operation
Telephone			
Fax			
Telex			
Email			
Internet			
Documentation			

VI. IT setup:

24. Please describe the IT set-up in your organisation

- a. Number of PCs:
- b. Number of servers:
- c. Operating systems used: Windows/NT/Unix

- d. Name of the database used: Oracle/SQL/DB2
- e. Number of LAN & WAN Network: LAN WAN
- f. Internet access: leased line/dial-up/broadband
- g. Number of people in computer department:

Thank you very much for sparing your valuable time.

Place:

Date:

