

USING XML IN CONJUNCTION WITH EDI

by
Rachel Fuller

A Master's paper submitted to the faculty
of the School of Information and Library Science
of the University of North Carolina at Chapel Hill
in partial fulfillment of the requirements
for the degree of Master of Science in
Information Science.

Chapel Hill, North Carolina

April 2000

Approved by:

Advisor

Table of Contents

| | |
|--|----|
| Section 1: Introduction | 3 |
| Section 2: Electronic Data Interchange (EDI) | 5 |
| Section 2.1: Need for Standards | 5 |
| Section 2.2: EDI Components | 8 |
| Section 3: Extensible Markup Language (XML) | 12 |
| Section 4: EDI and XML | 16 |
| Section 4.1: Disadvantages of EDI | 16 |
| Section 4.2: Using XML with EDI | 19 |
| Section 5: Conclusion | 25 |
| Appendix 1 | 28 |
| Appendix 2 | 29 |
| References | 30 |

Section 1: Introduction

In 1993, in her book EDI: A Total Management Guide, Emmelhainz stated that “it is no longer a question of if EDI will become a major factor in business; it is only a question of when. And when is likely to be very soon” (1993, 3). Five years later it was noted that fewer than 100,000 companies have adopted EDI, a standard that has been in existence for over 20 years (Senn, 1998, p. 8). EDI, Electronic Data Interchange, is a set of standards that facilitate business to business processes such as ordering, receiving, and settlement. EDI is an important tool in electronic commerce. Electronic commerce is commonly defined as the exchange of information, goods, services, and payments by electronic means. Electronic commerce operates within two primary areas: business-to-business and business-to-consumer. Since the advent of the World Wide Web, business-to-consumer electronic commerce has proliferated. However, business-to-business electronic commerce has lagged far behind. Since the creation of XML, Extensible Markup Language, many see this new technology as a solution to the problems of EDI and as a way to bring business-to-business electronic commerce to the forefront of industry today.

EDI's beginnings started within the transportation industry with the Transportation Data Coordinating Committee. There was a need to get various modes of transportation such as train, air, or ocean to share data such as schedules and to coordinate with each other and their clients to effectively deliver goods and supplies

(Desmarais, 1999, p. 86). From this EDI was born. It has been proven that in the companies that implement EDI correctly, those companies experience a vast reduction in the cost of processing business documents. Frequently, though, the savings associated with the implementation of EDI are often the result of reengineering of the underlying business processes together with the use of EDI (Colberg, 1995, p. 31).

There are many benefits to implementing EDI. Some of these benefits include reduced costs associated with the handing of these documents. EDI can eliminate data entry errors, which can be costly to repair. EDI can also eliminate manual tasks such as sorting, filing, or reconciling the documents. EDI implementation can reduce the time between processing an order and receiving the order. Decreased mailing costs are another benefit, as well as reduction in inventory (College of Business Administration, Oklahoma State University, 04/10/2000).

However, despite the many documented advantages to reorganizing a company's business processes with EDI, EDI can be costly to implement. Traditionally, only very large companies who constantly process a high volume of documents have benefited from EDI. Today's business climate now seems primed to operate within the realm electronic commerce and sees XML as the tool that will make EDI a viable proposition for all businesses. This paper will introduce the EDI standards and discuss advantages and disadvantages of EDI. This paper will also introduce XML and then discuss how EDI used in conjunction with XML is poised to lead the way into business-to-business electronic commerce.

Section 2: Electronic Data Interchange (EDI)

2.1: Need for EDI standards

EDI stands for Electronic Data Interchange. EDI is the computer to computer exchange of routine business documents in a standard electronic format between organizations. EDI enables computers to exchange data electronically, which is much faster, cheaper, and accurate than exchanging information by paper. Documents that have a high frequency rate of use or are time-critical are the best candidates for EDI conversion. These documents include things such as purchase orders, quotes, or invoices. Traditionally, companies transmitted business documents to each other using paper as the primary medium. A company who wished to purchase products from another company would use some internal application to type a purchase order. This order would then be sent to the company from which the goods or services were to be bought. Once the order was received, the information would have to be re-keyed into the application of the second company so it could then be processed. Inaccuracies become very common as a human has to read, interpret and then re-key information. This whole process tends to be very expensive and labor intensive. Additionally, if the document is being sent through the mail, then there is no control over the time of arrival or transit of the business document. These are the problems that had businesses searching for another way to transmit information. A major goal and purpose of EDI is to improve the flow and management of business information. EDI reduces costs and improves efficiency in

organizations that adopt it. It does this by creating an error free flow of routine documents between a company and its trading partners. Trading partners are the companies or business partners with whom business or technical information is exchanged using EDI. These business partners are either suppliers or customers.

Early electronic messaging systems were based on proprietary formats between a company and its trading partner. However, due to these differing formats, it was difficult for companies to exchange electronic data with many trading partners. The need was for a standard format for electronic data exchange. EDI began in the railroad industry. The TDCC, Transportation Data Coordinating Committee, worked from 1968 to 1975 to develop standards for EDI documents. These standards were used within air, rail, ocean and the trucking industry. More industries expressed interest in EDI-type systems and began developing such technologies within their own industries. There was no coordinated effort to develop inner-industry standards. Each industry was developing its own standards for EDI which were often not compatible with other EDI industry standards. Within the 1970s ANSI, the American National Standards Institute, took over the management of the EDI standards and committed to making the standards work across industries. Both users and vendors input their requirements for the standard format. These requirements specified that the standards were hardware independent; were unambiguous, such that they could be used for all trading partners; reduced the labor-intensive task of exchanging data (e.g., data re-entry); and allowed the sender of the data to control the exchange, including knowing if and when the recipient received the transaction (NIST, 1996, 04/10/2000).

The two major standards for EDI used today are the ANSI X.12 standard, and the UN/EDIFACT standard. The UN/EDIFACT standard is used primarily in Europe, while ANSI X.12 is used more commonly in the United States. Within ANSI there are numerous committees which oversee various standards for differing industries. These committees are accredited by ANSI. Such a committee is referred to as an ASC or accredited standards committee. Each committee has a name, and each committee that deals with the subject of communication starts with the letter 'X'. The twelfth committee in the communication section of ANSI standards deals with issues of EDI. So the committee that oversees the EDI standards is referred to as ANSI ASC X12 or as ANSI X.12. Work done by ASC X.12 is submitted to ANSI for review every three years. After a successful review, ANSI publishes any new standards that have been developed. EDIFACT is a European standards committee that operates under the direction of the United Nations. EDIFACT stands for Electronic Data Interchange for Administration, Commerce and Transport. EDIFACT syntax was adopted by the International Organization for Standardization (ISO) in 1987. This was also the year that DISA, the Data Interchange Standards Association, took over management and publication of the ANSI X.12 standards. ANSI X.12 and EDIFACT perform equivalent functions, but there are differences in their underlying structures. There is not a one-to-one correspondence between ANSI X.12 and EDIFACT.

Section 2.2: EDI Components

“EDI is not a technology in itself. It is, rather, a set of technologies that are designed to allow companies to use the ANSI X.12 standards to automate certain aspects of their businesses.” (Netscape, 1998. 04/10/2000.) There are three major components of an EDI system. Those components are the standards, the software, and the communication medium.

The EDI standards are a set of rules that proscribe how to transform a paper document into its electronic counterpart. The EDI software accomplishes the transformation of these documents. However, the ANSI standards dictate the required pieces of an EDI document. EDI documents are referred to as transactions or transaction sets. A transaction set is an electronic version of the paper document. Individual lines of information within the transaction sets are called segments. Each word that makes up a segment is called an element or a data element. EDI documents or transaction sets are identified by a 3-digit code and are typically called by their 3-digit number. For instance, 130 is the code that refers to a student educational record or transcript, 204 is the code that refers to motor carrier shipment information, 264 is the code for mortgage loan default status, and 850 is the code for a purchase order. The transaction set for a purchase order would be referred to as an 850 rather than as a purchase order. Within each segment, elements are separated by a delimiter. The most common delimiter used within an EDI document is an asterisk (*). Delimiters also mark the beginning and end of segments. The ANSI X.12 standard can be generally divided into three levels. The first

level contains transaction set tables. This is the highest level of the ANSI standard.

Transaction set tables list all the allowable segments for each transaction set. The next level of the ANSI X.12 standard is the Segment Directory. The segment directory specifies segments for particular transaction sets and the allowable data elements. The third level is the Data Element Dictionary. The data element dictionary defines each data element found within the Segment Directory.

The following is an example of what a purchase order might look like in its unstructured format, and the corresponding transaction set.

| | |
|---------------------------------------|-----------------------------|
| P.O. Number 4001 | ST*850*0001^ |
| P.O. Date December 31, 1992 | BEG*00*NE*4001**921231^ |
| Buyer: Allen Manufacturing | N1*BT*Allen Manufacturing^ |
| 123 North Street | N3*123 North Street^ |
| Largetown, NY 11111 | N4*Largetown*NY*11111^ |
| Vendor: Baker Supplies | N1*VN*Baker Supplies^ |
| P.O. Box 989 | N3*P.O. box 989^ |
| Somewhere, NY 10009 | N4*Somewhere*NY*10009^ |
| Ship to: Plant 1 | N1*ST*Plant 1^ |
| 456 West Ave | N3*456 West Ave^ |
| Smallsville, NY 10006 | N4*Smallsville*NY*10006^ |
| 5 cases part number BC436 @\$12.50/cs | PO1*1*4*CA*12.50**VP*BC436^ |
| Number of line items | CTT*1^ |
| | SE*14*0001^ |

(Emmelhainz, 1993, p. 62)

All transaction sets that are being sent to the same department of a company's trading partner are sent together in a bundle called a functional group. One functional group might go to the department which processes orders, while another functional group might be sent to the finance department. One EDI transmission can include two or more functional groups and is called an interchange (Krock, 1999, 04/04/2000). When a trading partner receives an EDI transaction set, the partner sends back an

acknowledgement called a functional acknowledgement that indicates the transmission has been received.

The task of EDI software is to translate information from unstructured, human-readable information into the structured EDI format. The software also translates from an EDI format back into a standard business format. To translate information into an EDI transaction set, three jobs must be performed. Those are mapping, extraction and generation. Software mapping identifies elements within a company database that are needed to create an EDI message. The act of mapping looks at the standards to see what information is needed to create the transaction set. This information is then located within the company's file system. Data is then extracted from the database or file system and parsed into a usable format that can be used to create the EDI. Usually the data is extracted and then restructured into a flat data file which will have fixed positioned records. Once the necessary information has been located and extracted into a flat file, the generation of the EDI transaction set can be accomplished using the translation software. The software will format the data into the required structure for EDI. The translation software will use tables of the data dictionary and syntax rules for data segments and elements of a transaction set. The transaction sets are then arranged into the appropriate functional groups and are then ready for communication to a trading partner.

There are two common ways in which an EDI transmission occurs. The first option is direct communication between commercial partners exchanging EDI documents. The sender directly links to the receivers computer systems through a modem. The second option for communication transmissions is the most popular way in use today. This option is to use a third party network, called a VAN. A VAN is a Value

Added Network. It is an electric clearinghouse for data. Some of the major VANs in operation today are GE, Advantis, MCI, and Harbinger. Utilizing the services of a VAN eliminates the need for companies to support differing communication configurations with their trading partners and also greatly reduces internal support requirements. The most common analogy used to describe the function and purpose of VANs is to compare a VAN to the post office. Just as the post office delivers mail from the sender to the receiver, so too, does a VAN ensure that the proper company receives an EDI transmission. A VAN receives transaction sets from a sender and places them in the electronic mailbox of the recipient. VANs came into being to allow trading partners with different hardware and software systems to communicate and share EDI documents. VANs were developed as a service to bridge the gap between companies who were primarily PC based and those companies utilizing EDI with mainframe computers. It is usually more cost efficient for most companies to use the services of a VAN to handle EDI transmissions than to change the companies internal communication network. VANs also offer a degree of security as they act as a buffer between trading partners. With the use of a VAN, trading partners will not have direct access to each other's systems.

Section 3: Extensible Markup Language (XML)

XML, extensible markup language, was derived from SGML, standard generalized markup language. SGML is an international standard (ISO 8879) that was developed to standardize the production process for large document sets. SGML was developed by Charles Goldfarb, Ed Mosher, and Ray Lorie at the behest of IBM who asked Goldfarb (a researcher at IBM) to build a system for storing and managing their legal documents. In solving this problem, the researchers noted three critical elements that had to be addressed. The first element was that for different machines to share documents, the programs needed to support a common document representation. Second, the format should be specific to legal documents; a domain specific vocabulary was needed. Lastly, the computer needed to be made to understand the text of the document as much as possible, and to accomplish this task, the documents would have to follow certain rules or be structured. In 1969, the IBM team developed a language that was not specific to a particular system. It was called Generalized Markup Language or GML. In 1974 a parser was developed. The parser was a software program that could read the specifications for the GML document and check that the document was marked up accurately according to those specifications. From this, SGML was born, which became an IOS standard in 1986. SGML has become the de facto standard for the interchange of large, complex documents today.

XML was designed for the digital representation of documents. However, XML was designed to be more flexible and robust than HTML. HTML, Hypertext Markup Language, was also derived from SGML. It was a simple specification used to share documents through hyperlinks. It was designed to be easy to implement without the complexity of SGML. It was designed with a fixed number of elements or tags. But the very simplicity that makes HTML so easy to use is also one of its main liabilities. Because of its fixed tag set, HTML has none of the extensibility of SGML or XML. HTML cannot be tailored to individual document types. XML, like SGML before it, is a meta-language. It is a set of rules for designing specifications for new documents. The documents are marked up with tags. Unlike HTML, whose tags are primarily used to describe the formatting and representation of the document, the tags used in SGML, and XML, are used for specifying information or content about the actual text of the document. HTML tags describe how the elements should appear, XML tags describe what the words within the elements are. This distinction means that XML documents are more portable and can be used in many different types of applications. According to the W3C specification, the goals for XML are as follows:

1. XML shall be straightforwardly usable over the Internet.
2. XML shall support a wide variety of applications.
3. XML shall be compatible with SGML.
4. It shall be easy to write programs with process XML documents.
5. The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
6. XML documents should be human-legible and reasonably clear.
7. The XML design should be prepared quickly.
8. The design of XML shall be formal and concise
9. XML documents shall be easy to create.
10. Terseness in XML markup is of minimal importance.

(W3C, 1998, 04/10/2000).

In essence, XML is designed to be easy to create, easy to read, and designed to be used over the Internet.

Just as with HTML, XML documents use *tags* to markup the document. These tags surround elements within the XML documents. Unlike HTML, the content of the tag is descriptive, not procedural. For instance: `<job_title>Webmaster</job_title>` describes the element. *Attributes* can be added to the tags to give additional information or processing instructions about the elements. The format of using attributes is as such: `<element attribute="value">cdata</element>`. The attribute is enclosed within the element tag. The value of the attribute must be enclosed within quotes: `<job type="internship">Webmaster</job>`. Cdata stands for character data. Cdata is the text of a document.

XML documents can be of two types, well-formed or valid. All XML documents are “well-formed” documents. A well-formed document is an XML document that was created without a DTD, or document type definition. A well-formed document complies with several rules regarding the formation of the document. For a document to be well-formed, all elements within an XML document have start and end tags. Element tags must be nested correctly; there can be no overlapping of tags. Attribute values must be enclosed within quotes. A valid XML document is a document that conforms to a DTD or document type definition. A DTD is a specification for creating documents of a certain type. A DTD is a set of rules that declare what types of elements are necessary or allowable for creating a specific document. A DTD is a separate file from the main XML document that provides a set of rules for the XML document to which it is attached. These rules are instructions for how the document has to be structured. DTDs define the

different elements that will be allowed within the XML document. For an XML document to be considered a valid XML document, it must be validated against the DTD.

An XML document specifies the content of the text of the document. Unlike HTML, however, the XML markup is not used to designate presentation of the document. To define a particular look for a document, a stylesheet must be used. A stylesheet, much like a DTD is a separate file from the XML document. The stylesheet is a set of rules that stipulates how the elements of the XML document should appear. One of the advantages to using a stylesheet, rather than specifying the presentation of a document within the document itself is that one stylesheet can be used to format an unlimited number of documents. It is also much easier to change the style of a document or set of documents when a stylesheet has been used.

Section 4: EDI and XML

Section 4.1: Disadvantages of EDI

Before the advent of the Web and the Internet, business to business electronic commerce was being practiced through Electronic Data Interchange. This was the first attempt of e-commerce. EDI eliminated manual processes by allowing the internal applications of different companies to exchange information directly. However, it has been shown that although “several million businesses participate in commerce every day, fewer than 100,000 companies have adopted EDI” (Senn, 1998, p. 8). It has been predicted that the value of business-to-business commerce over the Internet will grow from under \$100 billion in 1999 to about \$500 billion in 2002 and to \$1.3 trillion by 2003. (Kotok, 1999a, 04/10/2000.) Many see EDI used in conjunction with XML as a driving force in this new onslaught of Internet-based business-to-business commerce. Although it has been shown that EDI implementation can bring significant savings to companies, there are many reasons why the majority of companies in the United States choose not to implement EDI. EDI implementation can be a very expensive operation to undertake. The mapping, the startup costs for hardware and software, and the maintenance costs can be prohibitive for small to medium-sized companies thinking of starting EDI. As Desmarais explains, companies must realize huge savings to break even with the costs of implementing an EDI system. Because the quantity of documents exchanged determines the amount of savings, large companies that process many

transaction sets can offset their costs in a shorter time than small and medium-sized companies, who will find it difficult to justify EDI (1999, p. 87). Today, many see XML as a way to revitalize EDI and make it accessible to the millions of businesses who are currently operating without it, driving the future of business-business electronic commerce in the process.

One of the most important steps in generating an EDI transaction set is the process of mapping. The information that is being sent in a transaction set must be mapped from within the company's file or database system. Mapping of a company's data is often cited as one of the barriers to implementation of EDI. It can be very costly to maintain the mapping. Often the mapping involves proprietary software between a company and its trading partner. The role of EDI translation or mapping software was to support a variety of private system formats used by companies. Typically, a company and its trading partner would enter into a contract and design a tailored software program that would be dedicated to mapping between their two types of datasets. Each time a company entered into a new trading agreement with a new trading partner, a new translation software program would be needed by the new company to format their data to conform to the standards in use by the established companies. This becomes very expensive to maintain. A company basically has to start new with each trading partner. EDI interactions do not occur in one direction only. EDI is more than a company sending transactions sets to his trading partner. EDI is a two-way interaction between systems. Because these systems usually adopt different file formats, the process of exchanging data is very difficult.

One of the goals of the ANSI X.12 standard was to facilitate electronic commerce transactions by the establishment of a common, uniform business language. The language, EDI, comprises more than 300 transaction sets. ANSI X.12 tries to address the needs of nearly all industries and businesses, and therefore tries to cover all contingencies. EDI standards are large, complex, and difficult to implement. These transaction sets are fixed in detail and scope. The transaction sets also use strict processes for handling data. As noted by Kotok, this rigidity is a necessary evil and can be helpful when a company plans to quickly process thousands of detailed transaction sets. In this case, predictability in the incoming data stream is an advantage, and “rigidity can also mean stability” (1999b, 04/10/2000). However, as Goldfarb points out, this very rigidity and fixed syntax within the X.12 standards is another major disadvantage to the implementation of EDI. It is impossible to add additional tags or business information to a transaction set without additional costly mapping of the software between trading partners. Companies become frozen by the fixed transaction sets. As he states, “This inflexibility inherent in the current custom solutions required to map data between each trading partner pair is untenable, despite the significant benefits of EDI” (Goldfarb, 2000, p. 162).

Kotok and Goldfarb are both in agreement, though, that the current pace of standards evolution for EDI has in no way kept pace with the business needs of companies operating in today’s world of electronic commerce. It can take years to develop standards for new transaction sets. Also, since ANSI X.12 is not compatible with UN/EDIFACT, companies who wish to conduct business electronically with trading partners in Europe had to carry at least two sets of electronic formats for each transaction.

Section 4.2: Using XML with EDI

The advent of the World Wide Web has seen the proliferation of business-to-consumer electronic commerce and is primed for business-to-business electronic commerce. Due to limitations of HTML, XML is fast being touted as the new standard for Web documents. The characteristics that may make XML the successor to HTML are the same characteristics that will be used to revamp EDI in light of today's changing business environment. "Using XML as the standard for EDI transmissions could provide standardized protocols that allow for change, and at the same time, preserve existing EDI content" (Harvey, 1998, p. 58).

According to the goals for XML, XML is meant to be easy to create, and easy to read. Unlike an EDI transaction set, which is meant to be parsed together by complex mapping software, an XML document is designed to be created a human. The document itself can be processed and understood by both humans and computers. Each piece of information can be identified in terms that are semantically understood by both individuals and computer applications such as browsers, databases, and spreadsheets. No longer will companies have to invest in costly software solutions to be able to transmit data to a trading partner. One reason for the complexity of traditional EDI messages is that the creators of EDI were very concerned about the size of their messages. In an effort to save bandwidth, EDI messages were designed to be compressed and used codes to represent complex values within the original document. There is no metadata in any of the messages. This complexity makes EDI applications expensive to buy and maintain.

Goldfarb presents an example of an EDI transaction set and its equivalent XML representation (2000, pp. 183-84). See Appendix 1. The EDI transaction is unreadable by humans, but it is fairly simple to pick out pertinent information such as the purchase order number in the XML representation (appendix 2).

Traditional EDI was designed with the assumption that individual trading partners would agree upon both the format and the structure of transactions sets for their particular business needs. The format was then frozen and translators were built around those specifications. Any change in the format would result in costly additions to the original software design. Using XML to format EDI, this would no longer be a problem. The extensibility of XML, the ability to design tags according to individual needs, would negate costly negotiations and reprogramming of business systems.

It is not feasible to think that companies who have already invested heavily in EDI will abandon EDI to work towards XML/EDI solutions solely because of new, emerging technologies (Harvey, 1998, p. 62). She sees one solution to be the use of XML as a wrapper for traditional EDI information, much as XML can carry any other web data type, such as audio or video. This information would be transmitted using HTTP protocol just as an XML document would be.

There are many different schemes for implementing EDI with XML. One of the most active groups today taking part in this new revolution is the XML/EDI group. The XML/EDI group is a grassroots organization who has devoted themselves to improving business to business electronic commerce through the use of XML and EDI. The goal of the XML/EDI group is to “deliver unambiguous and durable business transactions via electronic means” (NIST, 1996, 04/10/2000). The XML/EDI group sees the need for

more than just an XML wrapper for EDI. They see XML/EDI as the fusion of five technologies: XML, EDI, templates, agents, and repository. The XML/EDI group sees XML/EDI as a dynamic framework where each individual technology is used to leverage the others.

- XML is seen as the base for this framework. XML tokens would replace or supplement existing EDI segment identifiers. These tokens would be used as the syntax that would transport the other components across the Internet.
- XML/EDI would provide 100% backward compatibility to existing EDI systems. The framework would be based around templates. These templates would travel with the XML document and be a supplement to the DTD.
- Agents, developed either with Java or ActiveX, would interpret the templates to perform whatever task is needed, or to interact with the user to create new templates for each new job.
- The final piece of the framework is a shared Internet directory or repository which would provide automatic lookups of EDI elements for their meaning and definitions.

The XML/EDI group envision the technical layers upon which a base XML/EDI structure can be built as such:

Application & Repository
Rules
Template & Java/ActiveX
XML tags/Agents, DataBots
XML parser/generator
XML/EDI data
DOM or File/Message
Storage/Transport

These are flexible layers of which not all are required to be used. Different layers support different targeted electronic commerce systems. As a company's system for electronic

commerce becomes more demanding, each successive layer provides more sophisticated capabilities so as to handle more complex needs.

An integral part of the XML/EDI framework is the repository. A repository is a location, usually the Internet, where information about EDI and XML can be deposited, stored, maintained, and updated. These repositories would also supply utilities for manipulating and processing XML, providing stylesheets, and other utilities. Harvey notes that several working groups are beginning to create repositories, even before defining a framework for electronic commerce. She states, “Whoever holds the key to the repository hold the key to the process.” (1998, p. 63). In addition to the XML/EDI group, Commercenet.com is just one of the many working groups who recognize the importance of a repository upon which an electronic commerce framework can be based.

Commerce.net introduced a registry service in May of 1998. Called eCO Framework, their framework is based around three core services. These services are the “semantic integration of multiple database types with data libraries, trusted open registries, and agent-mediated buying” (CommerceNet). At the base of their registries are schemas drawn from XML-based commerce libraries. These libraries will consist of information models for business concepts that includes business descriptions of companies, services and products; business forms such as purchase orders and invoices; standard measurements such as date, time, and location; and classification codes.

The XML/EDI group states in their document, “White Paper on Global XML Repositories for XML/EDI”: “[the] goal is to facilitate interoperable EDI methods. The wholesale use of XML repositories is thus required to ensure common definition points across standards” (XML/EDI, 1999, p. 15). The task before XML and EDI repositories is

to make sense of the differing semantic interpretations for standards. XML/EDI group notes that over 50 standards bodies have published XML based standards, or are actively developing them. (XML/EDI, 1999, p. 4). Alschuler defines this problem as a problem of “semantic interoperability” (Alschuler, 2000, 04/15/2000). For a schema or standard to be useful, there will need to be an information model documenting the semantics of the standard or definition in question. Alschuler clearly states the requirements of semantic interoperability:

To pull a schema off the shelf or down from a repository site and put it to work, the schema has to be a known quantity, part of a known framework of interoperable schemas or one with an unambiguous derivation from a known information model.”

(Alschuler, 2000, 04/15/2000)

The XML/EDI group has posited a framework for XML repositories that would allow for the separation of layers of information within the repository. They see the repository as a server that serves semantic distinctions depending on the domain in which the item will be used. The repository is not just a clearinghouse for differing XML standards, but is a solution to the proliferation of these differing standards. The repository will act as a cap on the extent of continuing semantic explosion of XML standards by steering reusable information content into the repository. The repository will be divided into functional roles which are categorized and then implemented. These roles are :

- Information Content
- Context Views
 - Physical World
 - Document World
 - Narrative Sequence
 - Domain Knowledge
 - Workflow
 - Reference Model
- Rules

Language
Terminologies
(XML/EDI, 1999, pg 5).

For instance, the Narrative Sequence would deal with the structure and text of the documents. The Document World would deal with metadata relating to ownership, medium, security, access profiles, etc. Each layer will map to a specific technology that provides the functionality. Semantic dispersal is addressed by allowing users to map their specific local needs onto the standardized definitions within the repository. In this way the XML repository would provide a single reference that XML/EDI based applications could use to ensure consistency of shared definitions and processes.

The XML/EDI group believes that there is no one solution for any electronic commerce transaction. Each transaction has its own requirements and goals. It is for this reason that the model XML/EDI proposes is a framework and not an application or module. “The goal of the framework is to provide formal interfaces for commercial EC components to interoperate. For XML/EDI to be successful these interfaces will be open and yet standardized.” (XML/EDI, 1998, 04/15/2000).

Section 5: Conclusion

Together, EDI and XML can create more than just the sum of their parts.

Although XML is seen as an answer to the problems that have traditionally plagued businesses implementing EDI, several groups, including XML/EDI see XML as more. XML combined with EDI is being touted as a new framework for business-to-business electronic commerce. EDI was one of the first technologies developed for electronic commerce. With the advent of the Internet and the World Wide Web, however, there has been a shift towards digital processing of information and away from electronic processing. With this shift has been a change in the way companies conduct business. Business-to-consumer e-commerce has quickly become a de facto way of life for consumers. Business-to-business exchanges are now taking the lead in the expansion of the Internet.

XML has expanded the reach of EDI to many companies who were previously prohibited from implementing EDI because of the high costs of startup and maintenance. XML's extensibility has provided an answer to the problem of differing EDI standards between trading partners. With the flexibility to create tags that specify the content of a document, trading partners should no longer have to suffer through costly upgrades to their mapping and translation software. XML's extensible tag set can create EDI objects that can either be passed or dynamically referenced to objects stored in repositories.

These repositories are critical because of XML's extensibility and will become a key technology in any XML/EDI implementation. It will be critical to have a

clearinghouse of documentation and standards for electronic commerce and be able to leverage the incongruous semantics between those standards.

The XML/EDI group is just one of many who are working today to change how organizations manage and transfer business information. Using XML in combination with EDI, a new paradigm is being developed for electronic commerce. However, the success of their endeavor is not guaranteed.

Right now, there are many obstacles that must be overcome before using XML in conjunction with EDI can be a reality. EDI is an expensive, rigid standard that few companies in the United States have implemented. Large companies who have a high turnover of documents benefit most from EDI. Smaller companies are at a disadvantage because the costs to implement and maintain EDI often override any savings they may experience. For any EDI/XML standards to become a reality, the needs of the minority of large companies who have currently implemented EDI must be leveraged against the majority of US businesses who are prohibited from using EDI because of size and costs. Current EDI-using companies will be resistant towards any solution that does not include full backwards compatibility for EDI systems already in place. Yet, a solution must also include companies who have no EDI systems and no plans to implement them. For EDI to continue today and be a viable part of electronic commerce in the realm of the Internet and World Wide Web there must be a major overhaul of the existing technologies. Probably the most overwhelming problem with EDI is that the EDI standards are not really standard. Individual trading partners who need to add elements to a transaction set to represent their companies own needs do so at a high cost which involves proprietary translation software. This action then limits a company to using the software with their

trading partner. If this company wants to do business with a third company using EDI, they then have to repeat the process with another translation software program that can only be used with this new company.

The EDI/XML group has proposed XML as a vehicle for EDI implementation. The extensibility of XML is seen as a way to overcome the rigidity of EDI without having to implement costly proprietary translation programs. It is not certain that XML will provide the answers that EDI needs. XML is such a new standard that the tools for creating and viewing XML documents are still being developed. Also, for XML schemas to be used as a standard for EDI transaction sets, there must be a set of standard schemas. This is a problem not just for EDI but for any endeavor in which XML will serve as the base. Right now, there are DTDs being developed by organizations and committees for a realm of different areas. Yet, these documents are difficult to find. There may be multiple versions of the same DTD from one company. It is often unclear if a DTD for a specific area is definitive. There be multiple DTDs developed by competing organizations that deal with the subject area. Repositories are being planned for XML, but as of yet, there are no viable repositories.

XML may very well become the solution to EDI, allowing EDI to become an integral part of electronic commerce. However, it is too early to say that XML will be the solution for the problems inherent in EDI implementation. XML is a new standard that has issues of its own that must be addressed first before it can be of any use to EDI

Appendix 1: AN EDI purchase order

ISA*00* *00* *08*61112500TST *01*DEMO WU000003
 *970911*1039*U00302000009561*0*P?
 GS*PO*6111250011*WU000003 *970911*1039*9784*X*003020
 ST*850*397822
 BEG*00*RE*194743**970911
 REF*AH*M109
 REF*DP*641
 REF*IA*000100685
 DTM*010*970918
 N1*BY*92*1287
 N1*ST*92*87447
 N1*ZZ*992*1287
 PO1*1*1*EA*13.33**B*80211*IZ*364*UP*718379271641
 PO1*1*2*EA*13.33**B*80211*IZ*382*UP*718379271573
 PO1*1*3*EA*13.33**B*80211*IZ*320*UP*718379271497
 PO1*1*4*EA*13.33**B*80211*IZ*360*UP*718379271848
 PO1*1*5*EA*13.33**B*80211*IZ*364*UP*718379271005
 CTT*25
 SE*36*397822
 GE*1*9784
 IEA*1*000009561

(Goldfarb, 2000, pp. 183).

Because of the high bandwidth necessary to transmit an EDI transaction set, all metadata is stripped from the document. Codes are used to represent every element that would occur on the physical document, such as the purchase order number, the item number, the quantity, etc. The document becomes impossible for a human to read and the use of translation software becomes necessary.

Appendix 2: Purchase order from figure 1 represented as an XML document

```

<?xml version="1.0" ?>
<?xml:stylesheet?>
<purchase-order>
  <header>
    <po-number>1234</po-number>
    <date>1999-02-08</date><time>14:05</time>
  </header>
  <billing>
    <company>XMLSolutions</company>
    <address>
      <street>601 Pennsylvania Ave. NW</street>
      <street>Suite 900</street>
      <city>Washington</city>
      <st>DC</st><postcode>20004</postcode>
    </address>
  </billing>
</order items="1">
  <item>
    <reference>097251</reference>
    <description>Widgets</description>
    <quantity>4</quantity>
    <unit-price>11.99</unit-price>
    <price>47.96</price>
  </item>
  <tax type="sales">
    <tax-unit>VA</tax-unit>
    <calculation>0.045</calculation>
    <amount>2.16</amount>
  </tax>

```

...

(Goldfarb, 2000, pp. 184).

Although this document is designed to be parsed by software that can distinguish between XML tags and data, it is also readable by humans. It is very simple to pick out important information such as the purchase order number or sales tax amount for an item.

References

- Alschuler, Liora. (2000). Schema Repositories: What's at Stake? [Online]. Available: [<http://www.xml.com/pub/2000/01/26/feature/>].
- Colberg, Thomas, P., Nicole Willenz Gardner, Kerry Horan, Dennis McGinnis, Phillip McLauchlin, Yuk-Ho So. (1995). The Price Waterhouse EDI Handbook. New York: John Wiley & Sons.
- College of Business Administration, Oklahoma State University. Background and Peripheral Information for the Electronic Data Interchange (EDI) Course. [Online]. Available: [<http://www.bus.okstate.edu/sharda/mba5161/edipic.htm>].
- CommerceNet.[no date]. What is the eCO Framework? [Online]. Available: [<http://eco.commerce.net/what/index.cfm>].
- Desmarais, Norman. (1999). Chaos - XML and its Potential for Ecommerce. Against the Grain. 11. (2), 86-89, 93.
- Emmelhainz, M.A. (1993). EDI: A Total Management Guide. New York: Van Nostrand Reinhold.
- Goldfarb, Charles F. and Paul Prescod. (2000). The XML Handbook. Upper Saddle River, NJ: Prentice-Hall.
- Harvey, Betty. (1998). XML and EDI – Natural Partners. EDI Forum: The Journal of Electronic Data Interchange. 11. (3), 58-64.
- Kotok, Alan. (1999a). Less is More in E-Business: The XML/EDI Group. [Online]. Available: [<http://www.xml.com/pub.1999/11/edi/>].

Kotok, Alan (1999b). EDI, Warts and All. [Online]. Available:
[<http://www.xml.com/pub/1999/08/edi/index2.html>].

Krock, Eric. (1999). Electronic Data Interchange (EDI) on the Internet. [Online].
Available: [http://developer.netscape.com/viewsource/krock_edi/krock_edi.html].

National Institute of Standards and Technology (NIST). (1996). Guidelines for the
Evaluation of Electronic Data Interchange Products. [Online]. Available:
[http://www.nist.gov/itl/div896/ipsg/eval_guide/evaluation_guid.html].

Netscape Communications. (1998). Electronic Data Interchange (EDI)
Fundamentals. [Online]. Available:
[<http://developer.netscape.com/docs/presentations/xpert/edi/edi.html>].

Senn, James A. (1998). Expanding the Reach of Electronic Commerce: The
Internet EDI Alternative. Information Systems Management, 15, (3), 7-15.

XML/EDI Group. (1998). Guidelines for using XML for Electronic Data
Interchange [Online]. Available:
[<http://www.geocities.com/WallStreet/Floor/5815/guide.html>]

XML/EDI Group (1999). White Paper on Global XML Repositories for
XML/EDI. [Online]. Available: [<http://www.xmledi.com/repository/xml-repWP.htm>].

W3C. (1998). Extensible Markup Language (XML) 1.0 Recommendation.
[Online]. Available: [<http://www.w3.org/TR/REC-xml>].

