

Enterprise Extension through Extensible Markup Language¹

Mohamed E. Hussein. University of Connecticut. U.S.A.
mo@sba.uconn.edu

Kinsun Tam. University at Albany, State University of New York. U.S.A.
tam@albany.edu

Abstract. An extended enterprise is comprised of multiple linkages between and amongst a company and its suppliers, distributors, customers and others. Linkages are long term collaborative agreements based on synergies and the ability to create value. An extended enterprise eliminates wasteful transaction costs and removes redundancies, delays and inefficiencies from the supply chain. This is accomplished through the coordination of demand forecasting, production planning, deployment and transportation as well as creating organizational and process links with seamless information flows between them. Extended enterprise is enabled by developments in technology, especially information technology. Sharing information about sales forecast, production schedules, inventory, etc. makes an extended enterprise a win/win situation. Extended enterprise networks have used information technology systems such as electronic data interchanges, enterprise resource planning, and the Internet with different degrees of success. The Integrated Manufacturing Technology Initiative has identified several information technology criteria as critical to the success of future enterprises. Based on these criteria, this paper discusses XML's contribution to the extended enterprise paradigm. Dell's direct sale model is used to illustrate the role of XML in enterprise extension.

Key words: *Extended enterprise, Extensible Markup Language, Electronic data interchange, Enterprise Resource Planning; Extensible Business Reporting Language*

1. INTRODUCTION

The extended enterprise (EE) concept has its beginning in the implementation of new management practices such as just-in-time (JIT) manufacturing. JIT depends on timely delivery of the right number of defect-free components. This requires close relations with suppliers. As companies realized the benefits of closer

¹ The authors are thankful to Gim Seow for working examples on extended enterprise, to Paulo Goes and Michael Kraten for suggestions on improving the paper, and to an anonymous reviewer for constructive comments. The work of K. Tam was supported by a Faculty Research Award Program grant of the State University of New York at Albany.

relations with suppliers, higher quality and lower costs, the concept of closer relations was expanded to include all linkages in the chain from raw material extraction to disposal and/or recycling of product. This came to be known as supply chain management. As companies have positive experiences with the supply chain concept, they started to see the value of a close relationship wherever there is synergy and value to pooling resources and talents to meet customers' demand. For example, AOL and Coca-Cola agreed to pool their marketing resources and brands name recognition to further their promotional strategies. In some cases alliances were created among competitors. For instance, six major airline companies banded together to sell discounted ticket on the Internet through an alliance named hotwire.com. In addition, GM, Ford, DaimlerChrysler, Renault and Nissan formed a consortium to create the digital trade exchange Covisint. Hence, the term extended enterprise is used to reflect the fact that these close relationships go beyond the traditional supplier/customer relations. According to Duffy (2000), "Organizations are no longer linked together in a linear value chain; they are connected to form a web, which presents a much more complex tableau."

EE is enabled by developments in technology, especially information technology. McNair *et al.* (2000) argue that EE is more about information flows than physical flows. Sharing information about sales forecast, production schedules, inventory, etc. makes EE a win/win situation. EE networks have used Information Technology (IT) systems such as electronic data interchanges (EDI), enterprise resource planning (ERP), and the Internet with different degrees of success. The purpose of this paper is to discuss the IT criteria critical to EE and examines XML's role in enterprise extension. In doing our analysis, we refer to criteria identified by the Integrated Manufacturing Technology Initiative (IMTI, 2000) as critical to the success of future enterprises. Such criteria include information sharing capacity, minimal additional investment, flexibility and interoperability, and security and integrity.

The remaining sections are organized as follows Section 2 and 3 describe the EE and its information technology requirements. Section 4 discusses the benefits and limitations of ERP and EDI. Properties of XML are given in Section 5.

Section 6 explains how XML and its derivatives automate data flow across the EE. Section 7 discusses XML's benefits and limitations with respect to EE's information technology requirements. Summary and conclusions of this study are provided in Section 8.

2. EXTENDED ENTERPRISE

An EE is comprised of multiple linkages between and amongst a company and its suppliers, distributors, customers and others. Linkages are long term collaborative agreements based on synergies and the ability to create value. According to McNair *et al* (2000), EE eliminates many of the wasteful transaction costs and removes redundancies, delays and inefficiencies from the supply chain. This is accomplished through the coordination of demand forecasting, production planning, deployment and transportation as well as creating organizational and process links with seamless information flows between them.

An EE provides many of the benefits of arm's length market transactions and of vertical integration while avoiding many of their risks. EE offers the flexibility and efficiency of the market while avoiding traditional adversarial relations that may lead to high transaction costs and uncertainty. EE also avoids the disadvantages of internal vertical integration in a rapidly changing environment. Companies can have access to economies of scale and skills without the necessary capital investment. EE partners can provide as well as gain access to innovations and new technology. Another important contribution of EE is knowledge. Duffy (2000) argues that knowledge is difficult to replicate because the validation process depends on certain skills and experiences. Therefore, an organization cannot possibly know everything it needs to know. Knowledge sharing is a key benefit of the EE.

The impetus for EE is the change in competitive environment where companies found themselves forced to compete on both quality and costs. New management and manufacturing methods such as JIT and total quality management were developed to enable companies to compete in the new competitive environment. To get the full benefit of these methods, companies found that they have to cooperate with their suppliers and distributors.

To deliver high quality and low cost products, all entities involved must be efficient. Any inefficient linkage in the supply chain will make the final product less competitive. It is argued that competition occurs between supply chains and between EE's rather than between companies. An EE requires its members to surrender some of their autonomy and share proprietary information to enhance the efficiency throughout the chain. Without trust and a clear articulation of what benefit they will receive in exchange, companies may not be willing to surrender autonomy or reveal proprietary information. Hence, complete and timely communication is critical to the success of EE.

3. INFORMATION TECHNOLOGY REQUIREMENTS FOR EE

An EE derives competitive advantages through collaboration among constituents. These advantages include, among others, quicker adaptation to changing requirements, productivity increases, lower costs, faster processes, shorter cycle times and greater customer responsiveness. To realize these advantages, EE constituents collaborate in areas such as strategy, operations, organizations, and technology.

Information Technology (IT) collaboration provides the foundation for collaborations in the other areas. For instance, to collaborate in production and delivery, EE constituents may need to not only synchronize business processes, but also integrate back-office applications such as Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) to enhance information flow. In addition, performance measurement and monitoring systems have to be extended across the EE to communicate control information. Accordingly, collaborations in various business functions typically presuppose considerable collaboration in IT. In fact, many of the benefits promised by EE are contingent upon an IT framework that enables the necessary communications among constituents.

The Integrated Manufacturing Technology Initiative (IMTI) shares a similar view on the role of IT on EE. IMTI recognizes the ongoing migration towards the EE paradigm. A totally connected EE, according to IMTI, will reduce development

and manufacturing costs, improve product quality, speed up time to market, enable greater customer responsiveness, increase competitiveness and market share, and generate higher return on capital. IMTI expects future manufacturing information systems to incorporate product libraries to enable product data interchange between individuals, department units, and EE partners. Future asset management systems will dynamically and rapidly match assets to requirements across the EE. Moreover, future supply chain management systems will be fully interoperable across the EE. All these future EE capabilities need to be supported by corresponding IT developments. According to IMTI, IT systems necessary for the success of EE should have the following attributes: (1) information sharing capacity, (2) minimal additional investment, (3) flexibility and interoperability, and (4) security and integrity.

3.1. Information sharing

To operate accurately, quickly, and cost effectively, instantaneous integration of data both within the enterprise and across collaborating enterprises is needed. Efficient information sharing through interconnected business systems will ensure that decisions are made in real time and with a clear understanding of enterprise-wide impact.

IMTI expects future enterprise processes, equipment and systems to be linked via a robust enterprise-wide communications infrastructure that delivers the right information at the right time. Knowledge stored externally (such as industry standards) at repositories and knowledge captured locally (such as manufacturing knowledge) need to be accessible across corporate boundaries throughout the EE. An EE therefore needs an enterprise-wide consistent model of information sharing.

In addition, the rise of the EE model reflects a shift from managing resources physically within corporate boundaries to coordinating resources and processes owned by external parties. Dell Computer, for instance, is known for outsourcing manufacturing, shipping, and other business processes to its partners. Because there could be a large number of external partners (possible with different IT platforms) to collaborate with, EE's data sharing model must be widely adaptable

so as to allow disparate enterprise systems to share information and business processes while still operating autonomously.

3.2. Minimal investment

Extensive use of IT by an EE should be cost effective. IMTI calls for high compatibility in future business systems, so that new modules can be integrated with existing systems at minimal costs. However, in reality, there is usually a significant initial investment in systems and in the reengineering of processes and other back-end systems that small companies may not afford.

As an EE reaches out to its constituents, the complexity of collaboration rapidly multiplies. Collaboration is needed not only among multiple regional and departmental units within a single organization, but also among units of different organizations within the EE. Therefore, collaborations in IT must be scalable. In other words, an EE's IT framework must maintain its cost-effectiveness and service quality as the EE expands its boundaries. In particular, the IT framework must enable the EE to embrace smaller and less technologically advanced constituents such as the workforce, consumers, and suppliers. Currently, however, integration across an EE between sophisticated electronic systems of large companies and paper-based systems of small companies is limited and inefficient. A cost-effective integrative solution is needed.

3.3. Flexibility and interoperability

As business partners are electronically connected, the entire economy becomes more and more real-time. The Economist (February 2, 2002) describes this as a "now" economy. An EE has real-time capabilities because transactions are electronically transmitted and automatically processed in one continuous set across the supply, order, and financing chain. Ideally, when an order is placed, all affected systems — supply replenishment, credit checks, financial accounting, sales reporting — are updated at real-time so that an EE can react instantaneously to business changes. However, real-time processing may breakdown if some information items cannot be delivered by the communication protocols. This is common when an EE adds new businesses or forms ad hoc partnerships, or when

a customer needs extra attention. New businesses, new partnerships, or extra attention may require modification in the format of communication messages (e.g. adding extra pieces of data to a message). For an EE to become real-time or close to real-time, it needs the ability to immediately update communication protocols as conditions require. Communication standards therefore must be flexible.

In addition, an EE's IT system must overcome configuration differences in constituents' software and hardware. Multiple systems from different vendors across an EE need to be integrated to create competitive knowledge and generate comparative advantage. In addition, as business relationships evolve overtime, an EE needs to be able to modify system functionality at minimal cost. Cap Gemini Ernst & Young (2002) remarks that "customer expectations are high, but loyalties are low." In an era of e-business and web-commerce, competition has become so intense that a customer could be lost in a few mouse-clicks. The traditional approach that integrates one application at a time fails to meet the dynamism of the Internet economy. An EE needs flexible integration that can be quickly created or dissolved as needed (Radding, 2000).

Responding to this dynamism, IMTI highlights "Plug-and-Play Interoperability" as an important feature of future systems. The massive and expensive systems of today should be replaced by sets of flexible modules that plug into the enterprise's information system and become operable instantaneously. A new supplier or vendor, regardless of its business systems, should be able to plug into the enterprise's system with no integration cost. Distributed operations can then establish interoperable partnerships whenever necessary to respond to emerging opportunities.

3.4. Security and integrity

IMTI summarizes its vision on information systems for manufacturing enterprises as "providing the right information, at the right time, in the right place." An important component of this vision is precision and accuracy in manufacturing, product delivery, and business process. In an EE, production will depend on shared manufacturing knowledge bases to optimize

manufacturing processes. Greater customer responsiveness can be achieved by actively engaging the customer in product design and post- production support. Financial and resource information such as performance and costs need to be communicated across the EE to foster trust between partners and to monitor collaborative events (Tomkins, 2001). With communications among constituencies taking place electronically, EE has to develop control systems to ensure data security and integrity in information sharing.

4. BENEFITS AND LIMITATIONS OF ERP AND EDI

Several IT technologies were developed to facilitate the electronic exchange of data within and between organizations. Enterprise Resource Planning (ERP) and Electronic Data Interchange (EDI) are two IT technologies currently used for electronic transaction processing. ERP aims at integrating processes internal to an organization while EDI focuses on standardizing information on transactions with external parties. These technologies digitalize process and transaction information so that compatible business systems can exchange data electronically without human intervention. However, compatibility is limited by differences in standard, platform, and vendor. As an enterprise reaches out to collaborate with external constituents, it needs technologies that offer high compatibility.

4.1. ERP strengths

ERP is an information system that manages data for core business functions including finance, supply chain, production, sales, and logistics within an organization. By standardizing and integrating data among business functions, ERP system enables a company to efficiently execute business transactions within its boundaries.

ERP improves the cooperation and interaction between disparate business units of an organization by integrating department functions onto a single computer system. Since a single computer system serves all departments, the flow of paper documents among departments is eliminated. For instance, a customer order can flow through the ERP electronically, rather than in paper form. This avoids rekeying as the order passes from one department's systems to the next, and eliminates

human errors. Real-time and accurate information flow resulting from ERP improves coordination between departments.

If aided by an enterprise application integration (EAI) module, an ERP system can communicate with other computing resources.²

For instance, as a purchase order flows through the ERP's order-fulfillment process, a properly configured EAI module can automatically update the customer relationship management (CRM) system and billing and accounting systems.

4.2. ERP limitations

However, EAI is expensive, integrates only a limited set of applications, and is not flexible enough to support real-time changes in business processes (The Economist, 2002). Therefore, current ERP systems are only partially interoperable with other IT resources such as client-server, Web, e-commerce sites, and legacy applications even within the same enterprise. Integrating ERP systems with IT systems across enterprises is even more challenging.³ Because an integrative interface layer is missing, ERP systems cannot support direct communication with computer systems of suppliers, customers, and other external parties. Costly and inefficient integration methods are used for exchanging information among ERP applications from different vendors. Accordingly, direct and real-time information exchange is unavailable for joint processes distributed across multiple enterprises. Circuitous communication between disparate systems undermines the efficiency of collaboration among EE constituents.

EE requires more than what ERP is currently providing. ERP systems need to become easier to implement. They also need to integrate with Internet applications and link with systems and software used by other partners (Foster, 2000). Because ERP focuses on a company's internal efficiency, it is naturally deficient in inter-

² EAI are a set of plans, methods, and tools aimed at modernizing, consolidating, and coordinating computer applications in an enterprise.

³ The inability to use real-time data is another limitation of ERP systems. External partners such as suppliers may need real-time data for collaborative design, testing, and manufacturing. In addition, customers may demand real-time communication as regards personal preference, product availability, shipment status, and payment. Current ERP systems with planning functions based on historical forecasts do not capture or provide such real-time information (Foster, 2000).

enterprise operations. In an e-business age where considerable economic benefits come from closer ties with customers, suppliers, and other external constituents, ERP's internal focus is a costly limitation.

Efficiency gained internally can create only limited competitive advantage. ERP needs to be expanded beyond enterprise boundaries to capture additional efficiency through collaborating with external constituents. This means that ERP systems across an EE need to be integrated to form a collaborative framework. An example of this cross-enterprise integration approach is provided by EDI.

4.3. EDI strengths

EDI refers to the computer-to-computer transfer of business transaction information using standard, industry-accepted, message formats⁴. EDI enables automated inter-company business processing such as sending purchase orders, exchanging shipping information, and receiving confirmations. EDI standards specify how business transactions are encoded to form an EDI message (such as a EDI purchase order) to be communicated electronically between organizations. EDI software transports the output from one system across a network, electronically and without human intervention, to be processed as input to another system.

A typical EDI directly links the computers of customers, manufacturers, and suppliers. It removes the need for paperwork such as purchase orders and invoices between suppliers and their customers. By eliminating the paper documents medium, EDI avoids unnecessary processing time, transaction costs and costly errors.⁵

4.4. EDI limitations

However, EDI suffers from several limitations. EDI standards are inflexible. As industry standards aiming at uniformity, EDI standards are inherently restrictive. They

⁴ Current EDI standards includes Data Interchange Standards Association's ANSI X12 and the United Nations' EDIFACT.

⁵ Use of EDI has lead to increased efficiency and lower transaction costs in retailing. For instance, Wal-Mart's competitive advantage is built on an EDI-based information networking. Wal-Mart uses EDI to report point of sale data to suppliers to facilitate automatic replenishment.

are not flexible enough to cater to specific information needs of individual companies. Rigid EDI standards define what data is included in an EDI document, how such data is coded, and when a particular piece of data should or should not be used.

Unexpected data or omission causes the EDI message to be misinterpreted. If two trading partners need to specify alternate delivery address depending on the day-of-week, for instance, such information is too rich to be included in the EDI purchase order message. The two companies have to communicate such information separately, which requires extra effort and processing time to relate back to the original purchase order. Because current EDI standards are slow to respond to individual business needs, inefficiency exists in EDI-based EE.

EDI systems encode a business document into a continuous string of highly cryptic data elements separated by some delimiter. For instance, the following EDI message, taken from Gelinas *et al.*'s (1999) EDI illustration, start a transaction set, begin a new purchase order, identify the seller, and identify the buyer. The resulting EDI message, while understandable by EDI-enabled machines, is hardly readable by humans.

```
ST*850*73561 N/L
BEG*00*SA*BL2-1563***950901 N/L
N1*SE*Compu Supply**82645 N/L
N1*BY*Delta Fabricating**29327 N/L
```

EDI's human interfacing is awkward. The above message makes little sense because the metadata describing the meaning of the data is missing. Because EDI data is transmitted without its metadata, an EDI message requires special translation software that understands the metadata. When such metadata is missing, EDI information cannot be shared with a third system.

EDI is designed to support B2B, but not B2C transactions because a typical consumer does not have translation software. Presentation of EDI document for human understanding is clumsy. Where human interaction is needed, as in consumer participation in product design and in B2C transactions, EDI data is deficient.

Moreover, current EDI systems provide only an expensive and hence limited cross-enterprise integration solution. EDI is expensive because its complex standards require special EDI technical support staff. A limited market size makes EDI software costly. The use of proprietary communication networks such as value added network further raises EDI's cost⁶.

Small companies usually cannot afford the significant initial investment in EDI systems and in the reengineering of processes and other back-end systems (Kogan, 1997; Smith 2001)⁷. Only about 5% of companies outside the Fortune 1000 in the top 10 industrial countries are EDI users (OASIS, 2000). Therefore, the paperless electronic business environment, and its cost-effectiveness and efficiency, are yet to be realized by about 25 million small and medium enterprises in the world (OASIS, 2000).

EDI users cannot share information with nonusers because EDI is highly cryptic. Non-EDI software cannot interpret EDI data. Non-EDI users do not have access to expensive translation software and private network. EDI is therefore confined to large companies that can afford the investment and achieve economies of scale. Medium and small companies are relegated to a paper system. An EE constituted of organizations of different sizes therefore needs to maintain both the EDI system and the paper system. As in the case of a supply chain, a single inefficient component in the EE could make the final product uncompetitive.

Since part of the EE is paper-based and since translation between the two systems is needed, it cannot realize the full benefits of real-time, error-free, low cost transactions. The drive to cut costs and increase productivity sometimes causes large corporations to force their smaller trading partners to become EDI compliant (Hart and Saunders, 1997; Hart and Saunders, 1998). Medium and small companies are left with the dilemma of losing business or investing in EDI.

⁶ A value-added network (VAN) is a private network provider (sometimes called a turnkey communications line) that is hired by a company to facilitate electronic data interchange (EDI) or provide other network services.

⁷ According to Kogan (1997): "Small companies cannot afford the investments in private network connections, protocol definitions, and software development required for EDI as well as the modifications toward compatible standards between the parties involved." Smith (2001) expresses similar concern, "Large companies have used EDI for many years as an electronic replacement for paper invoices, paper checks, etc., but high costs barred its use by smaller companies."

5. PROPERTIES OF XML

One technology that avoids the limitations of ERP and EDI and meets IMTI's four criteria is the Extensible Markup Language (XML) and its derivatives. The syntax of XML is specified in the document "Extensible Markup Language (XML) 1.0 (Second Edition)." This document describes the structure of XML documents and the behavior of computer programs that process them. It has been reviewed by members of the World Wide Web Consortium (W3C) and other interested parties and has been endorsed by the W3C Director as a W3C Recommendation.⁸

XML is created to support commercial use of the World Wide Web and to enhance distributed document processing. Following the success of HTML as a simple and easy-to-use language over the Web, the same development philosophy was used to design XML for maximum ease of implementation and straightforward usability over the Internet. These design principles make it easy to create XML documents, to create a new XML grammar as the business logic requires, and to write programs that process XML documents.

As discussed below, the simplicity in creating any new XML grammar through extension provides XML with immense flexibility to support a wide variety of applications. This extensibility explains why XML is called an extensible language.

Sun Microsystems, a major contributor to early stage XML development, summarizes many benefits of XML:⁹

"XML promises to simplify and lower the cost of data interchange and publishing in a Web environment. XML is a text-based syntax that is readable by both computer and humans. XML offers data portability and reusability across different platforms and devices. It is also flexible and extensible, allowing new tags to be added without breaking an existing document structure. Based on Unicode, XML provides global language support. XML is poised to play a prominent role as a data interchange format in electronic business Web applications such as e-commerce, supply-chain management, workflow, and application integration."

⁸ The W3C creates Web standards, which are formally called W3C recommendations.

⁹ <http://www.sun.com/software/xml/faqs.html#3>

To address the requirements of commercial Web use, XML incorporates important capabilities, namely extensibility, structure representation, and structural validation, which are missing from HTML (Bosak, 1997). In particular, XML information providers can create new tags and attributes to semantically qualify their data. Document structures can be nested to any level of complexity to reflect relationships among data. Moreover, any XML document can contain an optional description of its grammar to allow applications that import data for processing to check the data for structural validity.

The following example (Illustration 1) of an XML document abstract illustrates extensibility — the creation of new tags as needed. For instance, opening and closing purchaseOrderNum tags are created to enclose the data R49470, resulting in an XML element <purchaseOrderNum> R49470 </purchaseOrderNum>. Tags describe the meaning of data to the system receiving the message. In addition, the same example also illustrates document structure. In particular, the nesting of purchaseOrderNum, purchaseOrderDate, supplier, shipTo, billTo, and itemList elements within the purchaseOrder element conveys the structural relationship that a purchaseOrder is composed of multiple components. An importing application can use a grammar description document called XML Schema (shown in a later section) to check data for structural validity. By contrast, HTML does not allow any customized tag such as purchaseOrderNum, has no means to represent the hierarchical relationship that a parent element consists of a certain child elements, and does not support structural validation.

6. FLOW OF XML DATA ACROSS EE

XML is designed to store, carry and exchange data (Williams, 2001). This section describes XML data flows across an EE. The flows of purchase order data in XML format illustrate how XML-related technologies enable automated data processing at EE constituents' parsers, applications, processors, and browsers. The flows of financial statement data in the Extensible Business Reporting Language (XBRL) format provide additional examples of XML-enabled interoperability across the EE.

```

<purchaseOrder>
  <purchaseOrderNum>#49470</purchaseOrderNum>
  <purchaseOrderDate>2001-06-21</purchaseOrderDate>
  <supplier>
    <name>Dell Computer Corp</name>
    <department>Sales Department</department>
    <address>One Dell Way, Round Rock, TX, 78682-2244</address>
  </supplier>
  <shipTo>
    .....
  </shipTo>
  <billTo>
    .....
  </billTo>
  <itemList>
    .....
  </itemList>
</purchaseOrder>

```

Illustration 1. XML Extensibility and Structure

6.1. XML Purchase Order

The system flowchart (Figure 1) below presents the logical data flows across Dell's EE and the processing steps involving XML documents. An XML business document such as a purchase order can be created in XML format, transmitted electronically across the EE to the vendor's system, validated and processed at the vendor's system to update related records, and then displayed on screen.

In (1, 2, 3), customer Citigroup creates XML purchase order in electronic form. There may be some manual input procedures at this stage but rekeying is never needed subsequently. The XML purchase order is then transmitted across the EE through the Internet (4) to the vendor Dell. Dell's system will receive the electronic version of the XML purchase order (5). Before further processing, the vendor's system checks the structural validity of this XML purchase order with respect to the grammar of purchase order specified in a DTD or an XML Schema (6). The part of the vendor's application software responsible for this validity check is called the parser (7). The application software will also further processes the XML purchase order to update ERP, CRM, and billing and accounting systems (7, 8). The XML purchase order is both machine- and human- readable. Therefore, the vendor can display the validated XML document (9) using presentation instructions from an XSL document (10) and a processor program that is typically

embedded in a current-generation browser (11). The display is then visible on a computer screen (12).

Dell's application software could be programmed to order necessary components from suppliers, to contract for delivery and to send out invoice, all through automatically creating new XML documents. This is possible because all necessary information for these documents are either stored online (8) or can be extract from the validated XML purchase order (9). Suppose Citigroup's purchase order includes a monitor, which Dell needs to order from Sony. Dell's application Software can extract the productNum and quantity from the Citigroup's purchase order, and use this information to create an XML purchase order to be sent to Sony (13). As in the case of Dell, Sony needs to validate the incoming purchase order (14, 15, 16) before processing it to update its records (17). Likewise, from Citigroup's XML purchase order, Dell's application software can extract the shipTo element to create an XML delivery memo for FedEx to pickup the computer from a Dell's facility and the monitor from Sony to deliver to Citigroup, and extract the itemList element to create an XML invoice for Citigroup. This automatic flow of information between customer, manufacturer, supplier and freighter reduces time and transaction cost between these links in EE.

6.2. Key XML Technologies Enabling Data Flow and Processing in EE

The flow of XML data across the EE depicted above relies on several XML technologies. Specifically, W3C's XML Recommendation is used in (3). DTD/XML Schema and XML parser are used in (6) and (7). XSL and XML processor are used in (10) and (11). This subsection examines how such technologies support EE.

XML document complying with W3C's recommendation

The following XML document (Illustration 2), created in compliance with W3C's XML Recommendation, carries the contents of a purchase order. XML tags communicate the meaning of XML data element to XML applications. For instance, the opening and closing <address></address> tag pair embedded within the <supplier></supplier> tag pair makes machines as well as humans understand that "One Dell Way, Round Rock, TX, 78682-2244" is the supplier's address.

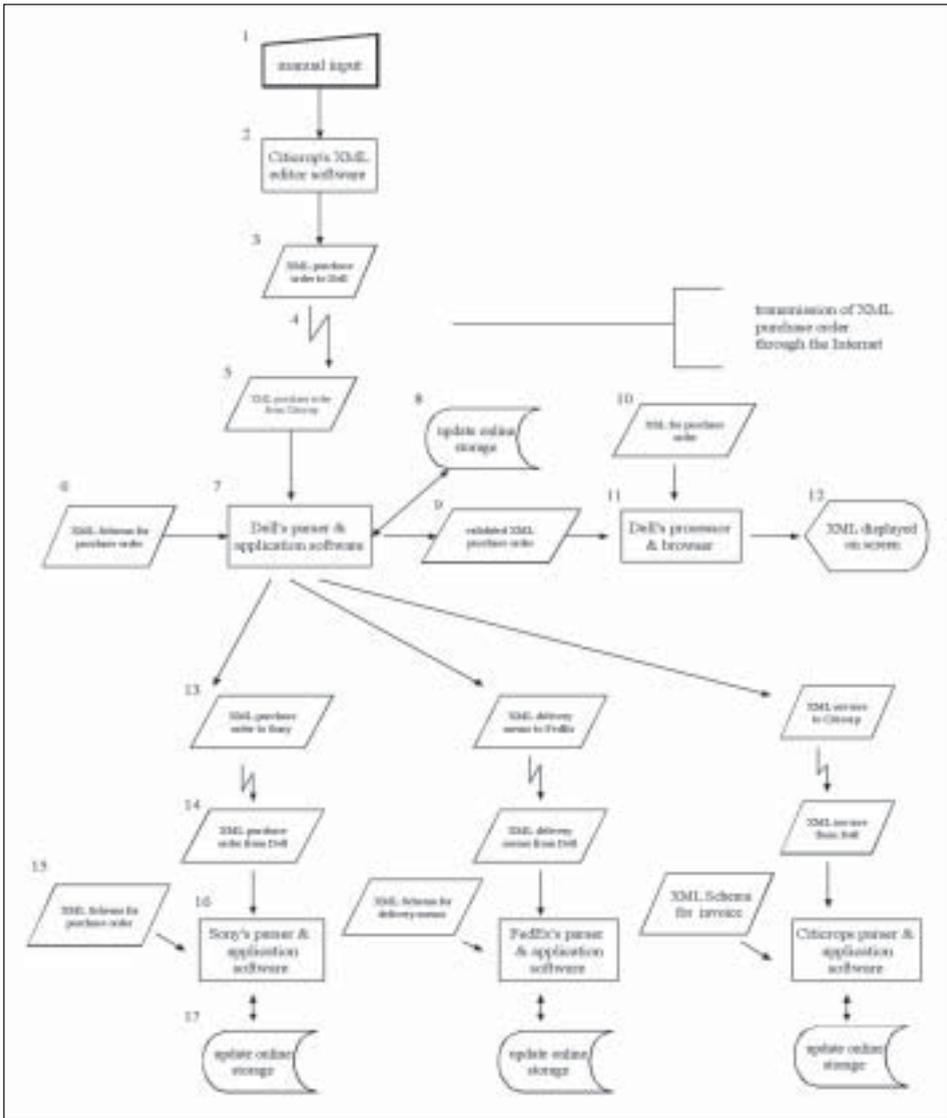


Figure 1. System Flowchart of Purchase Order Data Flows across EE
 Source: The chart is developed based on information from several public sources.

Likewise, data defining various attributes of a product item is enclosed in tag pairs such as <productNum> </productNum>, <productName> </productName>, <quantity> </quantity>, <price> </price>, and <amount> </amount>, which are embedded inside the <item> </item> tag pair.

```
<purchaseOrder>
  <purchaseOrderNum>R49470</purchaseOrderNum>
  <purchaseOrderDate>2001-06-21</purchaseOrderDate>
  <supplier>
    <name>Dell Computer Corp</name>
    <department>Sales Department</department>
    <address>One Dell Way, Round Rock, TX, 78682-2244</address>
    .....
  </supplier>
  <shipTo>
    .....
  </shipTo>
  .....
  <itemList>
    <item>
      <productNum>ETED-SD-411368-C</productNum>
      <productName>Dimension 2100 Desktop Computer</productName>
      <quantity>1</quantity>
      <price>888.00</price>
      <amount>888.00</amount>
    </item>
    .....
  </itemList>
</purchaseOrder>
```

Illustration 2. An XML Purchase Order

Grammar specified in DTD/XML schema

The XML grammar (a.k.a. taxonomy) for a specific purpose such as a purchase order can be defined either in a DTD (document type definition) or a XML Schema. The following XML Schema abstract (Illustration 3a) describes the expected structure of the purchase order in Illustration 2. It specifies, among others, how elements can be nested, what elements are allowed, and what data type is expected. For instance, it stipulates that a purchaseOrder element is of PurchaseOrderType, and must have, among others, an itemList as a nested element. An itemList element is defined to be of type ItemListType, which may have zero or more item elements. In addition, an item element must have productNum, productName, quantity, price, and amount as nested elements. Based on this XML Schema, a parser program can determine the structural validity of an imported XML purchase order document. Figure 2 describes these structural requirements pictorially.

```

<schema>
  <element name="purchaseOrder" type="xpo:PurchaseOrderType" />
  <complexType name="PurchaseOrderType">
    <all>
      .....
      <element name="itemList" type="xpo:ItemListType" />
    </all>
  </complexType>
  <complexType name="ItemListType">
    <sequence>
      <element name="item" minOccurs="0" maxOccurs="unbounded">
        <complexType>
          <sequence>
            <element name="productNum" type="xpo:SKU" />
            <element name="productName" type="string" />
            <element name="quantity" type="positiveInteger" />
            <element name="price" type="decimal" />
            <element name="amount" type="decimal" />
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
  .....
</schema>

```

Illustration 3a. An XML Schema for Purchase Order

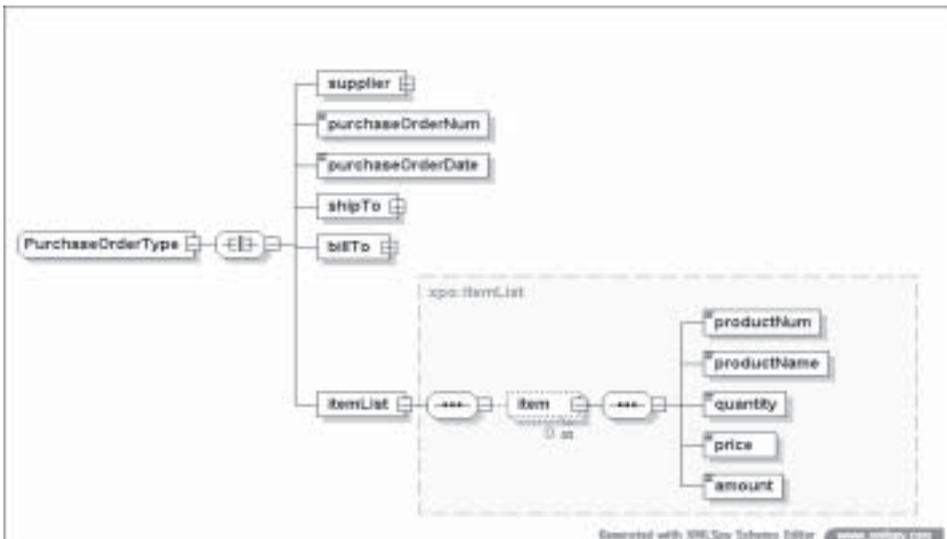


Figure 2: Hierarchy of Elements Stipulated by XML Schema

DTD and XML schema typically offers similar functionality. However, XML Schema is intended to replace DTD, and therefore, offers a richer and more advanced technology. DTD is more cryptic but less verbose. Just like XML Schema, DTD can be used to constrain an XML document. For instance, the following DTD fragment (Illustration 3b) stipulates the same structural validity requirements for the XML purchase order document as does the XML Schema.

Processing of XML document by application software

In addition to performing structural validation, the application software of the vendor will process the XML purchase order for necessary updates and business decisions. For instance, the application software may be programmed to extract the contents of the productNum and quantity elements from within the item element for further processing. If sufficient inventory exists, the inventory table needs to be updated for the quantity sold. The following illustration (Illustration 4a) shows how program codes for such a purpose can be written in the JAVA programming language using McLaughlin's (2000) JDOM application programming interface (API).

Specifically, doc is an object of the Document type defined in the JDOM API. This doc object invokes its `getRootElement()` method to access the root level element, namely `purchaseOrder`, of the XML purchase order document. The program then dives three levels below the root element, through invoking the `getChild()` method three times, to get to the `productNum` element. The content of `productNum` is subsequently returned through a call to the `getContent()` method, and assigned to the `pNum` variable. A similar strategy allows the content of the `quantity` element to be retrieved and assigned to the `pQty` variable. Retrieval of specific element contents

```
<!ELEMENT purchaseorder (supplier, purchaseorderNum, purchaseorderDate,
shipTo, billTo, itemList)>
.....
<!ELEMENT itemList (item*)>
<!ELEMENT item (productNum, productName, quantity, price, amount)>
<!ELEMENT productNum (#PCDATA)>
<!ELEMENT productName (#PCDATA)>
<!ELEMENT quantity (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT amount (#PCDATA)>
```

Illustration 3b. A DTD for Purchase Order

is possible because the XML Schema provides information about the nesting structure and the tag pair enclosing each piece of data.

```
//use the JDOM API
import org.jdom.Document;
Document doc = builder.build(new File(args[0]));

//retrieve contents of target elements productNum and quantity
Element root = doc.getRootElement();
String pNum = root.getChild("itemList").
                .getChild("item");
                .getChild("productNum");
                .getContent();

String pQty = root.getChild("itemList").
                .getChild("item");
                .getChild("quantity");
                .getContent();
```

Illustration 4a. Abstract of application software

The product number specified in the purchase order is now stored in the pNum variable. Therefore, pNum can be used to construct a Structured Query Language (SQL) statement to query first the MAKER table of the database to identify the producer for productNum (see Illustration 4b). If the MAKERID identifies a self-produced product, a second SQL statement can query the INVENTORY table for quantity of inventory currently on hand.

```
//query the MAKER table to identify producer from productNum
String queryMaker = "SELECT MAKERID FROM MAKER WHERE INVENTORYNUM = " +
                    pNum;
ResultSet rs = stmt.executeQuery(queryMaker);
While (rs.next())
{
    String mID = rs.getString("MAKERID");
}

//if mID refers to Dell itself,
//query the INVENTORY table for existing quantity;
//store result in currentQty
String queryQty = "SELECT INVENTORYQTY FROM INVENTORY WHERE INVENTORYNUM = "
                + pNum;
ResultSet rset = stmt.executeQuery(queryQty);
while (rset.next())
{
    int currentQty = rset.getInt("INVENTORYQTY");
}
```

Illustration 4b. Abstract of Application Software

The variable `currentQty`, representing the current inventory on hand, is compared with the numerical value of `pQty` to determine inventory adequacy (see Illustration 4c). If excess inventory exists, then the sale can be carried out and the INVENTORY table can be updated for the quantity sold using a third SQL statement.

```
//if adequate inventory exist, update inventory table for quantity sold
int excess = currentQty - Integer.parseInt(pQty);
if (excess >= 0)
{
    String updateQty =
        "UPDATE INVENTORY SET INVENTORYQTY = excess WHERE INVENTORYNUM = "
        + pNum;
    stat.executeUpdate(updateQty);
}
```

Illustration 4c. Abstract of Application Software

If MAKERID identifies an external supplier, the application software can automatically generate a new XML purchase order using the values of `productNum` and `quantity` extracted from customer Citigroup's original XML purchase order (see Illustration 4d). The same information is also useful for a new XML invoice to be sent back to Citigroup. Using a similar technique, Dell's application software can also copy the `shipTo` element from Citigroup's XML invoice onto a new XML delivery memo to FedEx.

```
//if mib refers to an external supplier,
//create a new XML purchase order to supplier
//using pNum and pQty from Citigroup's purchase order
System.out.println("<?xml version='1.0/'?>");
System.out.println("<purchaseOrder>");
....
System.out.println("<itemList><item><productNum>" + pNum +
"</productNum></item></itemList>");
System.out.println("<itemList><item><quantity>" + pQty +
"</quantity></item></itemList>");
....
System.out.println("</purchaseOrder>");
```

Illustration 4d. Abstract of Application Software

Displaying XML purchase order on screen

XML documents are readable by both machines and human beings. An XML-related technology called XSL (eXtensible Stylesheet Language) enables XML documents to be transformed into various formats (typically HTML, MS-Word,

and pdf) for presentation. To illustrate, the purchase order document in Illustration 2 is transformed into HTML by transformation instructions specified in an XSL file and then displayed in a browser (see Figure 3).

The XSL document (see Illustration 5) contains presentation instructions. For instance, XSL first requires looking for the purchaseOrder element, and specifies a tabular presentation format. Once the purchaseOrder element is matched, XSL looks inside for nesting elements purchaseOrderNum and purchaseOrderDate until it finds the targets. The node names of target elements are to be displayed in gold (FFCC33) and the contents in yellowish-white (FFFFCC).

```

<xsl:stylesheet>
  ....
  <xsl:template match="purchaseOrder">
    <table border="6" width="100%" cellpadding="20">
      <tbody>
        <tr>
          <xsl:for-each select="//purchaseOrderNum |
            //purchaseOrderDate">
            <td>
              <font color="FFCC33"/>
              <xsl:node-name/>
              <font color="FFFFCC"/>
              <xsl:value-of/>
            </td>
          </xsl:for-each>
        </tr>
      </tbody>
    </table>
    ....
  </xsl:template>
</xsl:stylesheet>

```

Illustration 5. Abstract of an XSL File for Displaying the Purchase Order as HTML

6.3. XBRL financial statements

The exchange of financial and resource information across the EE, as Tomkins (2001) argues, fosters trust between partners and helps monitor collaborative events. As in the case of the purchase order, XML and its related technologies facilitate timely and efficient processing of financial information across the EE. XBRL, an XML derivative, is developed as a framework for automatically preparing, publishing, exchanging and analyzing financial statement information (XBRL.org, 2001). The system flowchart in Figure 4 illustrates how financial statements in XBRL format can be automatically created and then electronically transmitted through the Internet to computer systems of EE constituents for further analyses.

Product Num	Product Name	Quantity	Price	Dollar Amount
ETED-SD-411362-C	Dell Dimension 2100 Desktop Computer	1	332.00	332.00
ETED-CP-411106-C	MSW Monitor	1	350.00	350.00

Figure 3. Purchase Order XML Displayed in a Browser

Dell's financial statements in XBRL format are automatically generated from application software and online storage (1, 2, 3)¹⁰. XBRL instance documents so generated can then be filed with the SEC and become available to financial statement users (4, 5), or transmitted to other EE constituents (7, 8). An EE constituent such as Sony, therefore, can obtain Dell's XBRL instances (9) either from the SEC (6) or directly from Dell (7). Before further processing, Sony's parser validates Dell's XBRL instances with respect to XBRL grammar (10, 11, 13). As a vendor to Dell, Sony may be interested in monitoring and analyzing Dell's financial health. Sony's application software (11), in collaboration with online storage (12), can be programmed to perform such an analysis. Automation is possible because XBRL tags specifically identify each piece of accounting data to Sony's application software. Finally, Sony's browser, aided by a XML processor that interprets XSL presentation instructions, can display Dell's financial statements on a computer screen (14, 15, 16).

¹⁰ Chou (2002) implements a mapping wizard that generates XBRL instances from accounting data in ERP.

XML technologies enabling XBRL data flows and processing are essentially the same as those supporting flows of purchase order data across the EE. Therefore, program codes of application software for XBRL processing are not detailed here.

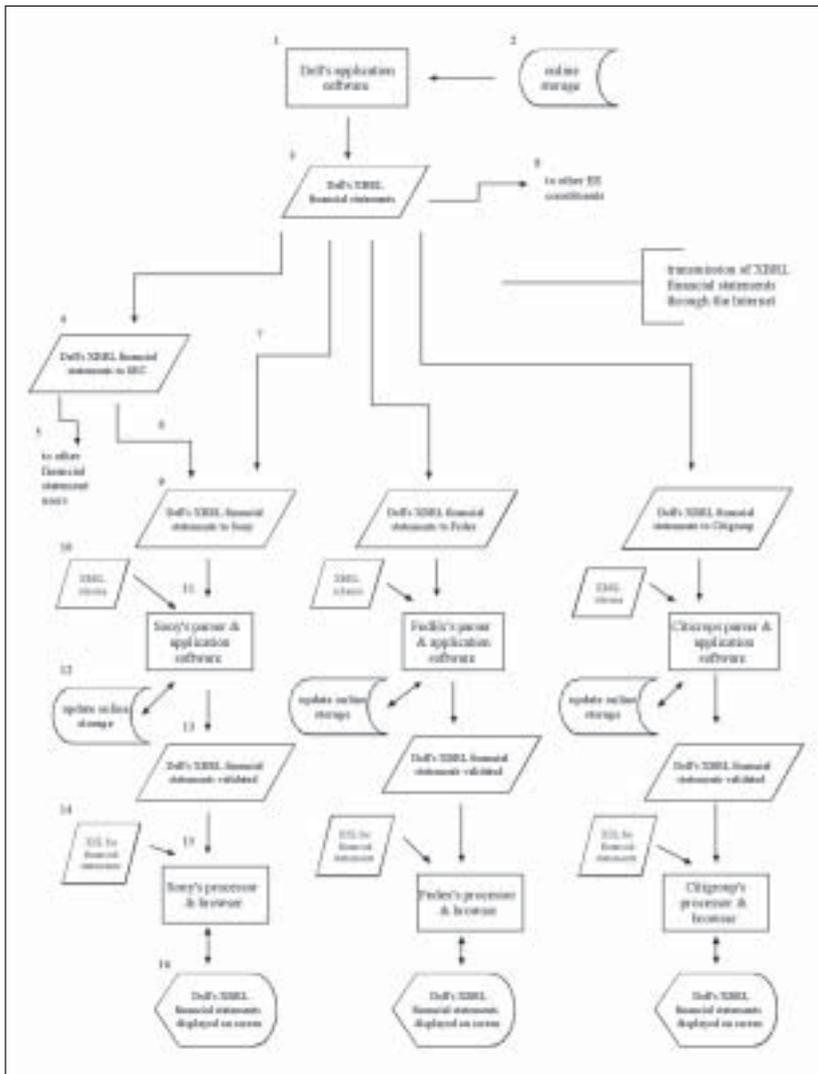


Figure 4. System Flowchart of XBRL Data Flows across EE
 Source: The chart is developed based on information from several public sources.

7. BENEFITS AND LIMITATIONS OF XML

7.1. How XML facilitates EE

XML is designed to simplify use, lower the cost of data exchange, offer data portability across different platforms, and support structural validity. These qualities are valuable to EE as we have seen in the Dell example. Accordingly, XML is adopted as a core technology in many EE applications and integrative solutions. This section discusses how XML meets EE's information sharing, minimal investment, flexibility and interoperability, and security and integrity requirements.

Information sharing

An XML document is self-describing because the data and metadata are stored together. XML enriches the message by adding context information through a hierarchy of tags. The following XML purchase order segment (Illustration 6) is understandable to both machines and humans. It conveys address information between systems. In addition, the hierarchical structure provides further information that the address belongs to a supplier on a purchase order. A third system, such as a search engine, can make reasonably good sense of this data. Unlike the case of EDI, any other systems that need to process this XML document do not need any special translation software¹¹.

```
<purchaseOrder>
  <supplier>
    <name>Dell Computer Corp</name>
    <department>Sales Department</department>
    <street>One Dell Way</street>
    <city>Round Rock</city>
    <state>TX</state>
    <zip>78682-2244</zip>
  </supplier>
  ....
</purchaseOrder>
```

Illustration 6. XML Data and Meta Data

¹¹ If a system needs to validate this XML document before processing, it may download the related grammar from an identified URL.

Representing data in plain text, XML is highly software- and hardware-independent. XML is thus an ideal common denominator for multiple legacy systems or multiple databases of diverse systems. Because different types of application can read XML data, XML provides a convenient way to share and exchange data across an EE.

Information systems of EE need to incorporate the consumer. IMTI expects future enterprises to actively engage the consumer in product design. The current generation of web browsers supports and displays XML. As a markup language, XML functions well in a Web based B2C environment, helping business to actively engage the consumer. As an EE reaches farther outside corporate boundaries and relies more on the Internet, XML's ability to share information with web applications will become more prominent.

Minimal investment

EDI's complexity and expense could drive many users toward XML (Linthicum, 1999). By contrast, XML offers a low cost, low learning curve, and low technical demand solution to e-business. These benefits reduce the overall installation and maintenance cost of an XML-based e-business system (GAO, 2002).

XML is a universal, public, and free standard. No single company can make it a private property. XML is augmented by related validation technology (DTD and XML Schema), presentation technology (XSL), and parser and processor technology, which, just like XML itself, are freely available in the public domain. In addition, based on ASCII, XML is hardware and software independent. XML users are not bound to any proprietary system. These features contribute to the low setup cost of XML-based solution.

XML is designed to be easy to use. Simple XML standard obviates special technical support staff. The costs to customize application software, and to develop XML documents, new XML grammar, and XSL presentation instructions are low. Moreover, XML documents are transmitted through the Internet, thus avoiding rental fees of dedicated data lines. Since practically all companies have access to the Internet, an Internet-based solution like XML offers enormous economy of scale, which further drives down any necessary costs.

To help companies around the world realize the advantage of XML in e-business, the United Nations and the Organization for the Advancement of Structured Information Standards (OASIS) are currently sponsoring the development of an XML-based e-business framework called ebXML. ebXML intends to “provide an open XML-based infrastructure enabling the global use of electronic business information in an interoperable, secure and consistent manner by all parties” (ebXML, 2002)¹². While available for companies of all sizes, ebXML is most beneficial to small and medium companies left out by EDI.

Following the tradition of XML-related technologies, ebXML is a nonprofit, and nonproprietary effort. It capitalizes on the benefits of XML as mentioned above, and avoids EDI’s costly path. In particular, ebXML’s design promotes the use of inexpensive, plug-and-play software. It provides a standard method to enables enterprises of any size to exchange business messages, conduct trading relationships, and communicate data in common terms over the Internet. By using freely available XML technology and by taking advantage of the economies of scale presented by the Internet, ebXML offers an economical e-business alternative to EDI. This alternative enables even small and medium enterprises to participate electronically in e-business, which in turn helps EE realize its full benefits.

Flexibility and interoperability

As an EE reaches out to seek new business opportunities, it needs communication standards flexible enough to accommodate unpredicted business needs. When new message components are needed, companies should not be left to wait for changes in industry standards, or wait for proprietary application vendors to revise software coding. XML provides facilities for the user to customize the grammar as required. In contrast, since EDI message definition (i.e. the grammar) is programmed into the proprietary software, which has to be recompiled upon modification, EDI document structure is not serviceable by user.

¹² ebXML’s infrastructure provides a company profile registry that specifies the registrant’s ebXML capabilities and constraints, and supported business processes and business messages. Companies can retrieve each other’s profile, and execute a Collaboration Protocol Agreement (CPA) if profile contents are mutually agreeable.

Unpredicted business needs may require customizing the grammar for the transmission of extra pieces of information. For instance, if two trading partners need to specify multiple delivery addresses depending on the day-of-week of delivery, they should be allowed to redefine the grammar of the purchase order to accommodate such data.

The following illustration (Illustration 7) shows a fragment of an original XML Schema for purchase order. The shipTo element, defined to be of the shipToType, has no facility to carry discretionary delivery information contingent upon the day-of-week. Therefore, such information cannot be communicated before the grammar is modified.

```

<schema>
  <element name="purchaseOrder" type="xpo:PurchaseOrderType"/>
  <complexType name="PurchaseOrderType">
    <all>
      ....
      <element name="shipTo" type="xpo:shipToType"/>
    </all>
  </complexType>
  <complexType name="shipToType">
    <sequence>
      <element name="name" type="string"/>
      <element name="department" type="string" minOccurs="0"/>
      <element name="address" type="string"/>
    </sequence>
  </complexType>
</element>
  ....
</schema>

```

Illustration 7. XML Schema before Redefining

XML provides flexibility through grammar extension. One way to extend or customize an existing grammar is by redefining. The shipToType of the above XML Schema can be redefined to include multiple shipToChoice elements, each specifying a day-of-week and destination option (see Illustration 8a).

Using the redefined schema, trading partners now can specify within the shipTo element multiple delivery choices depending on the day-of-week. The following XML fragment (Illustration 8b) specifies two delivery locations, one for Saturday and Sunday, and the other for Monday through Friday.

```

<schema>
  <redefine schemalocation="PurchaseOrder.xsd">
    <complexType name="shipToType">
      <sequence>
        <element name="shipToChoice" maxOccurs="unbounded">
          <complexType>
            <sequence>
              <element name="deliveryDayOfWeek"
                type="string" minOccurs="0"/>
              <element name="name" type="string"/>
              <element name="department" type="string"
                minOccurs="0"/>
              <element name="address" type="string"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </complexType>
  </redefine>
</schema>

```

Illustration 8a. Redefining XML Schema

```

<shipTo>
  <shipToChoice>
    <deliveryDayOfWeek>Sat-Sun</deliveryDayOfWeek>
    <name>Citigroup Inc</name>
    <department>Headquarters</department>
    <address>140 58th Street, Brooklyn, NY, 11220</address>
  </shipToChoice>
  <shipToChoice>
    <deliveryDayOfWeek>Mon-Fri</deliveryDayOfWeek>
    <name>Citigroup Inc</name>
    <department>Citigroup Document Services</department>
    <address>399 Park Avenue, New York, NY, 10043</address>
  </shipToChoice>
</shipTo>

```

Illustration 8b. XML purchase order based on redefined XML Schema

In addition to having a flexible grammar, XML enhances interoperability between ERP and non-ERP applications such as e-commerce sites and legacy applications. Traditional ERP integration occurs at the application level, one application at a time. The economic and time cost to integrate all applications is prohibitive. XML integrates at the data level by standardizing the format of exchanged data. Typically disparate systems gain interoperability when they can communicate on a common data medium. As XML standardizes the data format rather than the application, interoperability is not dependent on the application. In other words, XML facilitates information sharing with any application, including

future applications unknown at the current time, as long as these applications adopt the same data format.

In addition, since ASCII XML data is platform-independent and application-independent, computer systems of any vendor can be easily configured to process XML¹³. Some ERP vendors have already adopted XML for describing data. Enterprise Application Integration (EAI) software vendors are adding more integration capabilities via XML¹⁴.

XML's ability to carry structured data increases the efficiency of integration. When data and metadata (that describes the meaning of data) are separated, application programmers have to physically link each data to its metadata before developing programming logics. In contrast, XML relaxes the above limitation by binding data with metadata through a hierarchy of tags. Readable by humans, XML's structured data is optimized for easy programming. Therefore, structured data simplifies the job of making XML-based applications interoperable.

Similarly, XML offers inter-company integrative solution to EE. The main effort occurs at the ebXML project, which is intended to bring e-business capability to all companies¹⁵. As a core technology of ebXML, XML offers businesses of all sizes a common message structure and syntax for exchanging business data over the Internet. Current EDI users can participate in this new mode of data exchange because ebXML defines common data objects that allow companies to interchange standard EDI data with XML vocabularies. By bringing EDI and XML together, ebXML provides EE with a complete cross-enterprise integration solution.

Security and integrity

With communications between business partners taking place electronically, companies have to develop control systems to ensure that transactions are executed properly. XML-related technologies enable these controls. Public Key

¹³ EDI integrates at the data level too, but it is expensive solution because its format is cryptic, complex, and proprietary. EDI puts unreasonable burden on the humans who have to handle EDI data.

¹⁴ These vendors are SAP, Oracle, and PeopleSoft.

¹⁵ Two other initiatives, the XML/EDI Initiative and UDDI (Universal Description, Discovery and Integration) share a similar goal of enabling interoperable e-Business interactions (Webber and Dutton, 2000). XML is used in both.

Infrastructures (PKI) technologies can be applied to a complete XML document to achieve security objectives such as privacy, integrity, and authenticity. In addition, when different elements of an XML document are intended to be for different recipient, security protection can also be applied at the element level. XML security technology such as XML encryption and XML signature are currently being developed for this purpose (Mactaggart, 2001).

The following example depicts how encrypting an XML element enables secure transmission of transaction information. In the scenario described in Figure 1, some XML documents flowing across the EE may contain sensitive information. Dell, for instance, needs to send out a purchase order to Sony as outlined in Illustration 9a, but it may desire to keep its cost information secret. Therefore, Dell needs to encrypt the content of the <price> element to protect price information against eavesdropping.

```
<purchaseOrder>
  ....
  <supplier>
    <name>SONY CORPORATION OF AMERICA</name>
  ....
</supplier>
  ....
  <itemList>
    <item>
      <productNum>SDH-M51D</productNum>
      <productName>DELLICEPPO Series 15" Flat Panel LCD</productName>
      <quantity>1</quantity>
      <price>199.00</price>
    ....
  </item>
</itemList>
</purchaseOrder>
```

Illustration 9a. XML Purchase order before encryption

W3C's (2002a) candidate recommendation on XML Encryption Syntax and Processing specifies a process for encrypting data and representing the result in XML. Encryption can be applied on an XML document, an XML element, or XML element content. When Dell encrypts the content of the <price> element of the purchase order to Sony, the original content of 199.00 will be replaced by an <EncryptedData> element (see Illustration 9b). This <EncryptedData> element provides the encoded price, and specifies the algorithm and the name of the key necessary for decoding.

```

<purchaseOrder>
  ....
  <supplier>
    <name>SONY CORPORATION OF AMERICA</name>
    ....
  </supplier>
  ....
  <itemList>
    <item>
      <productNum>SDM-M51D</productNum>
      <productName>DELUXEPRO Series 15" Flat Panel LCD</productName>
      <quantity>1</quantity>
      <price>
        <EncryptedData xmlns="http://www.w3.org/2001/04/xmlenc#"
          Type="http://www.w3.org/2001/04/xmlenc#Content"/>
          <EncryptionMethod
            Algorithm=
              "http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
          <ds:KeyInfo
            xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
            <ds:KeyName>Sony Private Key A3</ds:KeyName>
          </ds:KeyInfo>
          <CipherData>
            <CipherValue>A23B45C56</CipherValue>
          </CipherData>
        </EncryptedData>
      </price>
    </item>
  </itemList>
</purchaseOrder>

```

Illustration 9b. Encrypting Sensitive Information

XML uses constraints to safeguard the integrity of data. Constraints are specified in either a DTD or a XML schema. An XML document containing data not conforming to a named DTD or XML Schema will fail to be validated by such DTD or XML Schema. An XML document that fails the validation process will be denied further processing. An EE using XML as a data medium therefore can rely on XML constraints to provide assurance of data integrity as long as appropriate constraints are spelled out in a DTD or XML Schema. For instance, the XML Schema fragment in Illustration 10 specifies allowable values for a Stock Keeping Unit (SKU).

```

<simpletype name="SKU">
  <restriction base="string">
    <pattern value="[A-Z]{4}-[A-Z]{2}-\d{6}-[A-Z]{3}"/>
  </restriction>
</simpletype>

```

Illustration 10. XML Schema fragment specifying data constraints

7.2. Limitations of XML

Critics challenge XML's semantics capacity. Kimbrough (2000) argues that XML, while offering a clear lexicon and a fully rigorous grammar, lacks the semantics for business messages. Madnick (2001) points to XML's limitations in handling rich semantics¹⁶. However, Kimbrough (2000) and Madnick (2001) believe that XML's semantic capacity can be enhanced by using a FLBC (formal language for business communication) to formally express business messages and by using semantic metadata¹⁷.

XML is a new technology with a short implementation record. Vendors rushing to unleash XML products are testing different implementation philosophies, but eventually only a few may survive. Therefore, an EE starting to embrace XML may assume additional business risk when adopting XML products.

Technical shortcomings of XML are attributable to its verbosity and its ASCII format. XML's opening and closing tags (and also attributes names) often take up sizable storage space and transmission bandwidth, undermining the efficiency of EE's information system. In addition, enabling only ASCII format means XML cannot be represented as binary and cannot transport binary data, such as multimedia data, between EE partners.

Fortunately, the open nature of XML promotes collaboration among multiple research and design groups to resolve these limitations. In particular, efficient taxonomy design can reduce verbosity of XML tags. New XML-related technologies such as XLink and WBXML are being developed to integrate XML with the binary environment¹⁸.

¹⁶ Madnick (2001) points to the semantic confusion of XML tag such as <P/E Ratio>, which may have multiple definitions depending on whether quarterly, yearly, historical, or forecasted earnings is used.

¹⁷ For instance, Kimbrough (2000) argues that XML cannot formally express the business message: "s requests of a that a pay s for item 789." A main emphasis of FLBC is to model speech acts such as promising, requesting, accepting, asserting, etc., which are common in business communication.

¹⁸ XML Linking Language (XLink) allows elements to be inserted into XML documents in order to create and describe links between resources. See <http://www.w3.org/TR/xlink/> for details.

WAP Binary XML (WBXML) Encoding Specification defines a compact binary representation of the Extensible Markup Language. This binary XML content format reduces the transmission size of XML documents and allows more effective use of XML data on narrowband communication channels. See <http://www.w3.org/TR/wbxml/> for details.

8. SUMMARY AND CONCLUSIONS

The EE paradigm of doing business is expected to increase the rate of innovation and productivity through the pooling of talents and expertise of different organizations. It will also reduce costs and increase speed to market. These improvements are enabled by advances in technology and business practices. However, there are still challenges to be overcome before EE can achieve its claimed potential. Among these is the challenge of developing or identifying the right types of IT. In this paper we discuss this challenge and analyze XML's suitability as an IT for EE.

9. REFERENCES

BALDWIN, C. (2000): "Data Integration Software Speeds Web", *Network World*, November 27, 2000, vol. 17, n. 48: 59.

BOSAK, J. (1997): *XML, Java, and the future of the Web*. <http://www.ibiblio.org/pub/sun-info/standards/xml/why/xmlapps.htm>

BOWEN, T. (1998): XML eases linking ERP apps. *Open Applications Group pushes XML for application integration*. <http://iwsun3.infoworld.com/cgi-bin/displayStory.pl?features/980928xml-erp.htm>

BRYAN, M.; MARCHAL, B.; MIKULA, N.; PEAT, B.; WEBBER, D. (1998): *Guidelines for using XML for Electronic Data Interchange*. <http://www.eccnet.com/xmledi/guidelines-styled.xml>

CAP GEMINI ERNST & YOUNG. (2002): *From EAP to EEA: Extending the enterprise*. <http://www.cgey.com/solutions/erpeea/media/CGEYPOVnchrt.pdf>

CHOU, C. (2002): "Getting XBRL from ERP - The Analysis, Design and Implementation of XBRL (XBRL Mapping Wizard) Applying Design Pattern and Java Technology", *Collected Papers of the Eleventh Annual Research Workshop on Artificial Intelligence and Emerging Technologies in Accounting, Auditing and Tax*, San Antonio, Texas, August 13, 2002.

COLEMAN, P. (2001): *ERP Integration Options*. http://eai.ebizq.net/erp/coleman_1.html

COOPER, R. (1994): Yokohama Corporation, Ltd. (A): *The Yokohama Production System*. Harvard Business School Case 9-195-070.

DUFFY, J. (2000): “Knowledge Management: What Every Information Professional Should Know”, *Information Management Journal* (July), vol. 34, n. 3: 10-16.

DYER, J. (1996): “How Chrysler Created an American Keiretsu”, *HBR*, July-August, 1996.

ECONOMIST (2002): *A Survey Of The Real-Time Economy*. February 2, 2002.

EDIDEV LLC (2002): XML vs. EDI. <http://www.edidev.com/index.htm>

FOSTER, T. (2000): *Needed: Quicker, Easier, Web-Friendly ERP. Supply Chain e-Business*. <http://www.supplychainebusiness.com/archives/8.31.00.ERP.htm?adcode=65>

GAO (2002): *Challenges to Effective Adoption of the Extensible Markup Language: Report to the Chairman, Committee on Governmental Affairs, U.S. Senate* (April). Washington, DC: GAO. <http://www.gao.gov/new.items/d02327.pdf>

GELINAS, U.; SUTTON, S.; ORAM, A. (1999): *Accounting Information Systems*, Cincinnati, OH: South-Western College Publishing.

HART, P.; SAUNDERS, S. (1997): “Powers and Trust: Critical Factors in the Adoption and Use of Electronic Data Interchange”, *Organization Science*, vol. 8, n. 1: 23-42.

HART, P.; SAUNDERS, S. (1998): “Emerging Electronic Partnership: Antecedents and Dimensions of EDI Use from the Supplier’s Perspective”, *Journal of Management Information Systems*, vol. 14, n. 4: 87-111.

IMTI INC (2000): *Integrated Manufacturing Technology Roadmapping Project: An Overview of the IMTR Roadmaps*, 24 July 2000. www.imti21.org/Documents/Roadmaps/IS/ISSummary.pdf

JUREWICZ, J. (2001): *EDI vs. XML: The technologies are actually completely compatible*. http://www.asptip.com/edi_vs.htm

KANTER, M. (1999): “Change is Everyone’s Job: Managing the Extended Enterprise in a Globally Connected World”, *Organizational Dynamic* (Summer), vol. 28, n. 1: 7-23.

KIMBROUGH, S. (2000): *EDI, XML, and the Transparency Problem in Electronic Commerce*. University of Pennsylvania working paper.

KING, J. (2000a): “Who Can You Trust?”, *Computerworld*, Nov. 13, 2000, vol. 34, n. 46: S10-S11.

KING, J. (2000b): “Quietly, Private e-markets Rule”, *Computerworld*, Sep. 4, 2000; v34n36: 1,16.

LINTHICUM, D. (2001): *B2B Application Integration: e-Business-Enable Your Enterprise*. Reading, MA: Addison-Wesley.

LINTHICUM, D. (1999): *When do you adopt XML?* <http://www.devx.com/upload/free/features/entdev/1999/06jun99/mm0699/mm0699.asp>

MACTAGGART, M. (2001): *An introduction to XML encryption and XML signature*. <http://www-106.ibm.com/developerworks/xml/library/s-xmlsec.html/index.html>

MADNICK, S. (2001): *The misguided silver bullet: what xml will and will not do to help information integration*. <http://ebusiness.mit.edu/research/papers/111%20SMadnick%20Silver%20Bullet.pdf>

MARION, L. (1998): *The limitations of extended ERP*. http://itmanagement.earthweb.com/entdev/print/0,,11979_605981,00.html

MCLAUGHLIN, B. (2000): *Java and XML*. Sebastopol, CA: O'Reilly.

MCLAUGHLIN, B. (2001): *ebXML: Not just another acronym — How ebXML plays in global trade*. <http://www.javaworld.com/javaworld/javaone00/j1-00-ebxml.html>

MURPHY, T. (1999): “Jeep Equipment Pushes DC’s One-stop Extended Enterprise”, *Ward’s Auto world* (February), vol. 35, n. 2: 49-50.

OASIS (2000): *ebXML Technical Architecture Specification*. http://www.ebxml.org/specs/index.htm#technical_specifications

PAUL, L. (2000): “The Biggest Gamble Yet”, *CIO* (April 15, 2000), vol. 13, n. 13: 144-156.

RADDING, A. (2000): A New Approach to Integration. *Informationweek*, August 28, 2000, n801: 91-102.

ROSS, J.; DEMERS, D.; OLSEN, G. (1998): *The Extended Enterprise and the Integrated Supply Chain*. http://www.commerce.virginia.edu/CMIT/activities/a_fall98.htm

SHANK, J.; GOVINDARAJAN, V. (1993): *Strategic Cost Management*. New York: Free Press.

SSA CONSULTANTS (2000): *EDI, database, and XML. The XML Zone*. http://www.reims.net/resource_zones/xml/xml_resources/XML_EDI_Database.html#data

TOMKINS, C. (2001): "Interdependencies, Trust and Information in Relationships, Alliances and Networks", *Accounting, Organizations and Society*, vol. 26, n. 2: 161-191.

W3C (1999): *WAP Binary XML Content Format: W3C NOTE*, 24 June 1999. <http://www.w3.org/TR/wbxml/>

W3C (2000): *Extensible Markup Language (XML) 1.0 (Second Edition): W3C Recommendation*, 6 October 2000. <http://www.w3.org/TR/REC-xml>

W3C (2001): *XML Linking Language (XLink) Version 1.0: W3C Recommendation*, 27 June 2001. <http://www.w3.org/TR/xlink/>

W3C (2002a): *XML Encryption Syntax and Processing: W3C Candidate Recommendation*, 04 March 2002. <http://www.w3.org/TR/xmlenc-core/>

W3C (2002b): *XML-Signature Syntax and Processing: W3C Recommendation* 12 February 2002. <http://www.w3.org/TR/xmldsig-core/>

WEATHERBY, D. (2001): *XML and EDI: Competitors or Partners. e-centre*. http://xml.e-centre.org.uk/download/xml_and_edi_partnership.pdf.

WEBBER, D.; DUTTON, A.. (2000): *Understanding ebXML, UDDI, XML/EDI. XMLGlobal*, http://www.xml.org/feature_articles/2000_1107_miller.shtml

WILLIAMS, K. (2001): *Professional XML Databases*. Wrox: Birmingham, UK.

XBRL.ORG (2001): *Extensible Business Reporting Language (XBRL) 2.0 Specification*. <http://www.xbrl.org/TR/2001/XBRL-2001-12-14.htm>