Submitted: COMPSAC 2002 Oxford, England, 26-29 August.

Case Study: B2B E-Commerce System Specification and Implementation Employing Use-Case Diagrams, Digital Signatures and XML[*]

Frederick T. Sheldon[1]
School of EECS,
Washington State University
email: sheldon@acm.org

Young-Jik Kwon
School of Computers and Comm.,
Taegu University
email: yjkwon@eecs.wsu.edu

Hong Chung
Dept. of Computer Engineering,
Keimyung University
email: hchung@eecs.wsu.edu

Woo-Hun Kim
School of Computers and Comm,
Taegu University
Email: macshow@taegu.ac.kr

Kshamta Jerath **Orest Pilskalns**
School of Electrical Engineering and Computer Science,
Washington State University
email: kjerath@eecs.wsu.edu | orestp@gemstone.com

Abstract

This paper presents a case study highlighting the best practices in designing and developing a B2B e-commerce system. We developed a remote order and delivery web-based system for an auto-parts manufacturing company. An initial scenario of the system was prototyped and then refined till the users and developers were satisfied. A formalized specification of the requirements employing Use-Case Diagrams and based on event flow was developed, and coded using XML to keep documentation simple and clear. Testing was performed at the component level allowing for feedback to previous steps when errors appeared. Digital signatures were employed for implementing security.

The implementation approach preserved and promoted harmonious communication among the users and the developers. The end product enabled a reduction in the processing time of transactions, reduced processing cost, improved accuracy, efficiency, reliability, and security of transmitted data; and at the same time our strategy shortened the System Development Life Cycle.

Keywords: Use-Case Diagrams, Digital Signatures, XML, Object Oriented Distributed Systems and Electronic Commerce.

[*] Upon translation from Word document to pdf the figures have become undecipherable. Given the last minute occurrence of this situation we are submitting the paper as is. The problem, however, will be fixed for the final version.
[1] Contact Author: Frederick T. Sheldon. Stettener Strasse 50/2, 73732 Esslingen, Germany. (+49 711 174 1339 Office | +49 179 675 9316 Handy | +49 711 1747054 Fax) sheldon@acm.org

1 Introduction and Technology Overview

The Internet has completely changed the way most businesses operate today. E-commerce uses internet-worked computers to create and transform business relationships. Web applications provide business solutions that improve the quality of goods and services, increase the speed of service delivery, and reduce the cost of business operations. However, many ventures into web application development fail because the systems are very complex and the users' requirements are continuously changing. Inefficient communication between the end user and the developer is another contributing factor. To successfully accomplish the development of a web application, one needs to visually model the system's architecture. A visual model helps in coherently grasping the changing user's requirements and effectively communicates them to the development team.

Requirements analysis along with abstraction (i.e., removing unnecessary details) are critical factors in web application development. It is easier and more cost-effective to correct an error at the requirement or design stage than at the implementation or maintenance stage. Further, formal specification provides unambiguous, precise and correct understanding of the user's requirements. Traditional requirement analysis consists of identifying relevant data functions that a software system would support. The data to be handled by the system might be described in terms of entity-relationship diagrams, while the functions might be described in terms of data flows [1]. Indeed, object-oriented analysis techniques offer class, use case, state chart, and sequence diagrams along with other diagrammatic notations for modeling.

Güell et al, [2] present a method that performs requirements gathering, conceptual and navigation design of Web applications, based on Scenarios, Use-Case and User Interaction Diagrams (UIDs). Scenarios are used to validate the requirements and are automatically generated from the use cases obtained from the users. They are also used to describe interface and navigational aspects, especially in the redesign of an existing web site.

This paper describes the development of a B2B e-commerce system using Use-Case diagram and Scenario for requirements analysis [3, 4], and Digital Signatures and XML. Critical success factors including effective communication between users and developers, processing time, process cost, reusability, efficiency, security etc. for building an Electronic Commerce system successfully are considered [5-10].

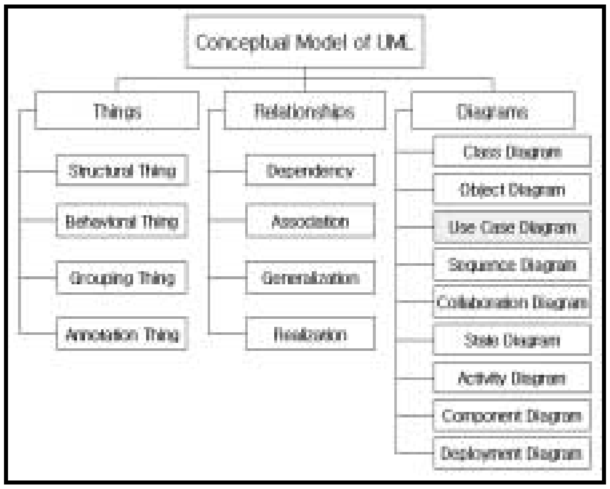
Section 2 provides background information on the areas of UML, Digital Signatures, XML and the critical success factors for building a successful web application. The research methodology employed for the requirements analysis is described in Section 3. Section 4 details the client and server applications and other implementation details for the application. Finally, the conclusions and scope of future work are detailed in Section 5.

2 Background Information and Related Research

Developing a web application requires making decisions and selecting technologies to support those decisions. We developed the B2B system described in this paper using UML and Use Case diagrams for formalization of user requirements; XML for documenting and transmission of data; and digital signatures for security purposes; steered by the critical success factors along the development process. Background information on each of these topics and related research in these areas are presented in this section.

2.1 Related work on Use-Case Diagram

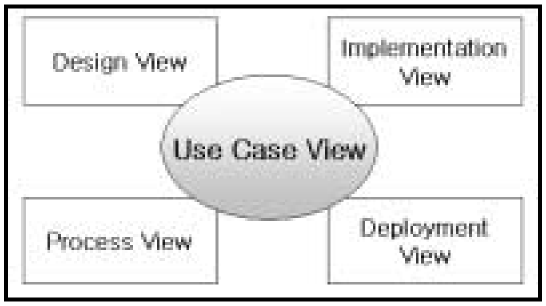
Visual modeling is a way of thinking about problems using modeling organized around real-world ideas [11]. Models are useful for understanding problems, communicating with project team members (customers, domain experts, analysts, designers, etc.), modeling enterprises, preparing documentation, and designing programs and databases. Models promote better understanding of the requirements by filtering out nonessential details and establishing the most suitable architectural basis for design. Most software systems that have been thoroughly modeled tend to be more maintainable systems.



(Src: © Choi, Yong-Lak, UML User Guide, 1999, http://www.uml.co.kr/)

Figure. 2-1 Conceptual Model of UML

Unified Modeling Language (UML) is a language used to specify, visualize, and document the artifacts of an object-oriented system under development. It represents the unification of the Booch, OMT (Rumbaugh), OOSE (Jacobson), and Objectory notations, as well as the best ideas from a number of other methodologists. UML is an attempt to standardize the artifacts of analysis and design: semantic models, syntactic notation, and diagrams. It provides a very robust notation, which grows from analysis into design. Certain elements of the notation (for example, classes, associations, aggregations, inheritance) are introduced during analysis. Other elements of the notation (for example, containment implementation indicators and properties) are introduced during design.



(Src: ©Choi, Yong-Lak, UML User Guide, '99, http://www.uml.co.kr/)

Figure. 2-2 Architectural Framework of UML

The Rational Rose product family is designed to provide the software developer with a complete set of visual modeling tools for development of robust, efficient solutions to real business needs in the client/server, distributed enterprise, and real-time systems environments. Rational Rose products share a common universal standard, making modeling accessible to nonprogrammers wanting to model business processes as well as to programmers modeling applications logic. Although it is one of the leading OO-CASE tools, Rational Rose requires considerable improvements in the support of OO characteristics, prototyping and support for teamwork development [12].

Use-Case Diagrams (Figure 2-1) model the user requirements and their interactions with the system at a very high level of abstraction. They are very useful for early requirements analysis because they enforce the identification of the different users and uses of a system and can be easily understood by customers [13][2]. In addition, Class and Instance Diagrams, Sequence Diagrams, Collaboration Diagrams, Class State Diagrams, Activity Diagrams, Implementation Diagrams are also present. UML can overcome most problems and be used to model most aspects of a system. The activity flow model has been successfully adapted to industrial projects including a leading German organization in the banking sector [14]. The architectural framework of UML is depicted in Figure 2-2. The contents of each architecture view are shown in Table 2-1.

Table 2-1. Contents of each architecture view

Architecture Style	Contents
Use-Case View	Explain System Behavior View for End User, Analyst, Designer, and Tester. Specify factors of concrete system architecture.
Design View	Present system service to End User. Consists of Class, Interface, Collaboration that make problem and solution area.
Process View	Present system ability, flexibility, and capacity. Consists of Threads and Process that make system consistency and synchronize mechanism.
Implementation View	Present shape management of systemic placement. Consists of Component and File that make physical system.
Deployment View	Present distribution, release, and settlement view of system physical part. Consists of Node that make H/W shape.

(Source: © Choi, Yong-Lak, UML User Guide, 1999, http://www.uml.co.kr/)

UML has been used in the development of business information systems based on *business object components* [15], and *business process modeling* [14, 16], to design service components of a telecommunications management system [17] (including component generation in a financial enterprise framework [18]). UML can be used within the context of a service-based architecture and component-based process [19]. Use cases may be employed not to create the architecture but to test it out and as a vehicle for solution delivery. UML notation can also be used to model families of systems [20]. UML supports architecture phase documentation [21] through its Development and

[2] Use cases are written in consultation with the customer. Thus, the use-case employs business terminology understandable by the customer and thus becomes the document that defines the scope of the customer's project (from,"Applying Use Cases: A practical Guide" Schnieder and Winters).

Component Diagrams. Use-Case Diagrams may be enhanced by providing contracts as a formal counterpart [22]. There are two important relationships among Use Cases, namely Uses and Extends which provide an object-oriented specification technique specially designed for formalization [23].

2.2 B2B Electronic Commerce and XML

In this paper, we focus on B2B e-commerce. In the USA, the trend of e-commerce transactions shows that B2B transactions increased from 8 billion U.S dollars in 1997 to $183 billion U.S dollars in 2001 [24]. Also, in Japan, it is predicted that the amount in B2B transactions will reach $680 billion U.S dollars in 2003, while in the USA, it should reach $1650 billion U.S dollars [25]. Accordingly, in the future, only companies that can cope with B2B e-commerce will survive in the global mega-competition.

Electronic Commerce (e-commerce) is the ability to perform business transactions involving the exchange of goods and services between two or more parties using electronic tools and techniques. Electronic commerce can be subdivided into four distinct categories: Business-to-Business (B2B), Business-to-Consumer, Business-to-Administration, and Consumer-to-Administration [26]. B2B e-commerce has its roots in electronic data interchange (EDI) networks established between large buyers and suppliers within a specific industry. E-commerce enables companies to conduct their business from prospecting to order processing and delivery on-line.

B2B e-commerce includes the use of exchanges - internet-based marketplaces in which companies can purchase or sell a variety of products, some generic across industries and others specific to a given industry [27]. Exchange technologies are basically Web sites that use a standard language, XML, to facilitate application-to-application data exchange. XML allows information regarding orders, purchases, payments, and products to be easily understood by other computers and makes the benefits of EDI accessible to organizations of all sizes.

Researchers have conducted several studies and validated the use of the technology acceptance model (TAM) for building Web applications [28]. Security, auditability, non-repudiation of transactions, internet technology for the creation of digital receipt and improved articulation of digital signatures are important to B2B e-commerce. Barclay T. Blair and John Boyer [29] suggested a method that supports them by using XFDL (eXtensible Forms Description Language). David A. Marca and Beth A. Perdue [30] presented a software engineering tool for developing process-oriented Internet applications that implements e-business connections.

In Web application development, technologies like using (1) e-commerce as a Domain for System Development, (2) PC-based Server Software as a Platform, (3) HTML as a Document Design Vehicle, (4) the Common Gateway Interface (CGI), (5) Visual Basic to CGI and, (6) a DBMS as a Live Data Source are desirable [31]. We considered several development tools for building Web Applications suggested by Jim Q. Chen and Richard D [32]. Client-side processing needs Java Applets and ActiveX components. Server-side processing needs CGI, ISAPI, ODBC, Java, JavaScript, VBScript, ActiveX, and CGI-script (Perl, C, C++). Other challenges of the web application development include security, content-rich maintenance, integration with legacy systems, fast development, scalability and load balancing.

This paper considers factors such as ease of use, speed, accuracy, security and reliability; all essential for building a B2B e-commerce system successfully [26]. We used DTDs (Document Type Definitions) and XML elements to denote the input and output of the service and values.

2.3 Digital Signatures

Digital signatures utilize encryption technology and offer such functions as signer certifications, forgery/falsification identification and transmission as well as repudiation. Figure 2-3 is a brief digital signature algorithm. It first creates a message digest by applying hash functions on a message that the sender will transmit. The hashing function creates a code value of regular length (a value that differentiates each message) by mapping a certain function for a message.

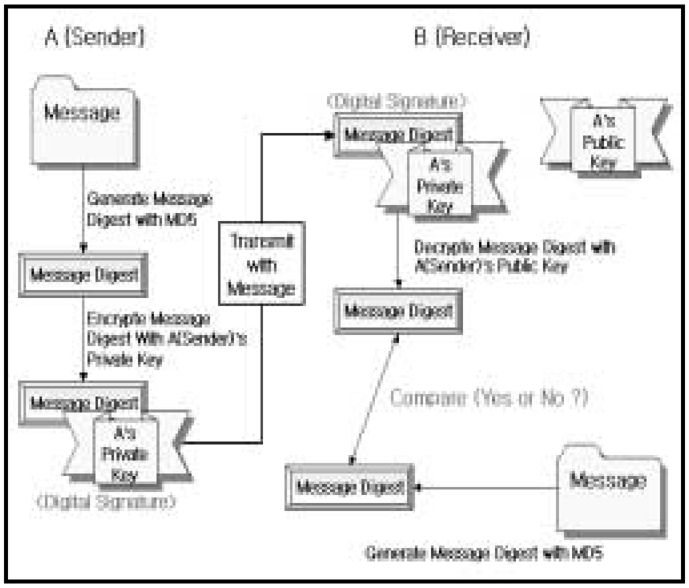


Figure. 2-3 Basic Flow Chart of Digital Signature

The code value is created using a single direction function (simplex) that cannot be used to unencrypt original messages. The sender encrypts the message with its private key and the message digest sends it to the receiver. The receiver using the sender's public key decodes the digital signature. A successful process is the signer's certification and transmission repudiation blockade. A message digest is also created. The receiver creates the message digest in a separate way as hash function MD5 from the original message that is received with digital signatures.

Use of a message digest offers the following advantage: the message digest is encoded instead of encoding all messages by public key encoding methodology, the run time is reduced and the integrity of the message can be confirmed and forgery and/or falsification of messages can be prevented.

2.4 Critical Success Factors for Building B2B e-commerce systems

We address the CSF (Critical Success Factors) for building B2B e-commerce systems successfully in this section. At a minimum, electronic commerce systems should increase the processing speed, accuracy, and efficiency of business and personal transactions. However, B2B e-commerce faces problems like partial solutions, rigid requirements, limited interoperability, insufficient trust and security and a lack of integration with existing business models. Therefore, we must consider factors like security, trust and reputation, legal expert, speed, reliability, accuracy, efficiency of business and transactions for building B2B e-commerce system successfully [26].

A scalable electronic brokerage architecture is required, that can, not only handle the diverse nature of existing and future goods and services but also the heterogeneity of the systems and networks deployed by the various actors (customers, suppliers, brokers, developers) involved in the supply chain [5]. Adaptive business objects and controlled interoperability among business alliances are the key enabling technologies to meet the challenge of integrated value chains [6].

There are two key security requirements: first, to provide users with integrated tools that guarantee privacy, security and fair trade in a framework that protects against criminal behavior and technical failure. Second, to provide suppliers with an environment that enables them to freely change the market model they use for trading digital goods [7]. A role-based access model for e-commerce has been suggested that separates the organization models from applications. Such a model allows for flexible modeling of organizational policies and dynamic authorization requirements in a dynamically changing business world [8].

Collaborative reputation mechanisms can provide personalized evaluations of ratings assigned to users to predict their reliability [9]. In this way, negotiation and trading between unrelated parties can be facilitated. Tan and Thoen [10] presented a legal expert system for electronic commerce that provides on-line explanations and reasoning about the use of trading terms, e.g. types of delivery for traded goods in contracts.

We have established the following CSF for building successful B2B e-commerce systems: (1) harmonious communication between users and developers, (2) reduced processing time of transactions, (3) reduced processing cost of transactions, (4) accuracy of business and transactions data, (5) efficiency of the systems that we are going to implement, (6) shortened systems development life cycle, (7) reliability of transactions data, and (8) security of transmitted data.

3 Empirical Study

A detailed empirical study based on the above stated *critical success factors* is presented. The methodology and process described here employs Use Case diagrams for requirement analysis and forms the basis of our research on best practices contribution.

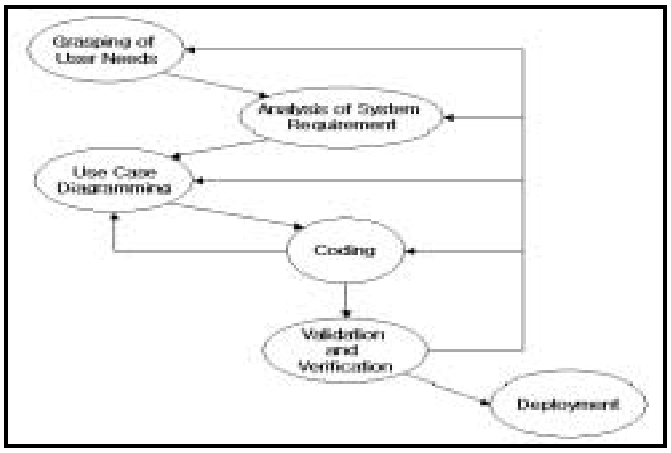


Figure. 3-1 Process Methodology of Research

For that reason, and based on the research of Albert Zündorf [33], we implemented a remote order and delivery web-based system for an auto-parts manufacturing company. The process methodology is shown in Figure 3-1. Myung Shin is a small auto parts manufacturing company (supplier/vendor) that delivers its products to several large automobile companies. Myung Shin set out to build a B2B system that is faster and more convenient. The main goal was to improve the order and delivery process between it's own company and other remote businesses (contractors). First, we drew up a scenario about that captured the existing order and delivery system as shown in Figure 3-2 and based on the study by Jean-Charles Pomerol [3].

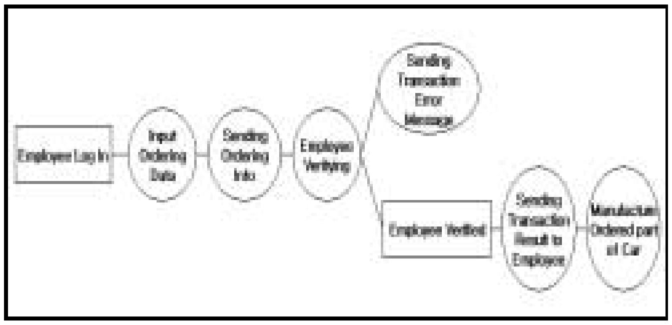

Employee Log In
Input Ordering Data
Sending Ordering Info
Employee Verifying
Sending Transaction Error Message
Employee Verified
Sending Transaction Result to Employee
Manufacturing Ordered part of Car

Figure. 3-2 Scenario for building B2B e-commerce system

3.1 ***Research Process and Methodology***

The first step in the process was to obtain the user's requirements. The users were presented with a scenario, which was iterated until the users were satisfied with the corresponding prototype. All requirements concerning business and transaction data were also developed in this step. The next step included formalizing the user's requirements using the Use-Case Diagram (which is based on event flow) and the Rational Rose tool. These specifications provided easy readability and understandability of the requirements. An iterative methodology was followed to ensure that all requirement-prototypes and designs conformed to the needs of the users. The order and delivery system was implemented using UML, XML and Digital Signatures as shown in Figure 3-3. The critical success factors were the controlling criteria in the implementation scheme[3].

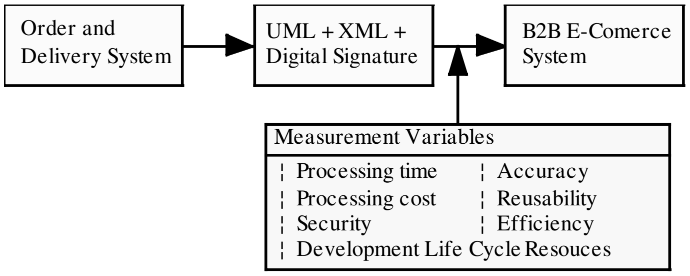


Figure. 3-3 Research Model

3.2 ***User's requirements and Use-Case Diagram***

After developing our order/delivery scenario, we developed Use-Case Diagrams based on the

[3] We use Window 2000 Server, Microsoft Visual C++ 6.0 for programming and Ms-SQL Server 7.0 as the Database. We document using XML (eXtensible Markup Language) and use ASP (Active Server Page) for receiving and ordering data. Also, we use using RSAEuro encode library, to encode/decode exchanged messages. To create a message digest, we use a single hash function MD5. The message switching is created with socket communications using a Microsoft Visual C++ 6.0. Rational Rose tool is used to draw the Use Case diagrams. We use 550Mhz CPU and 256MB memory system for the server and Intel Pentium II 500Mhz CPU and 128MB Memory system for the client.

aforementioned scenario and the elicited user's requirements with Rational Rose as shown in Figure 3-4.

3.2.1 Main Flow of Events

When employees A and B of a business company order something, the order and delivery system is started. The use case is shown in Figure 3-4 and works as follows:

(1) Input ordering data in delivering data input flow is processed, (2) Input data is verified, (3) XML data is sent to the server, (4) Receiver M logs in at the server, (5) Verifying employee verifies the received digital signature, (6) Transaction result at receiving flow of XML documentation is processed, and (7) Close Use Case.

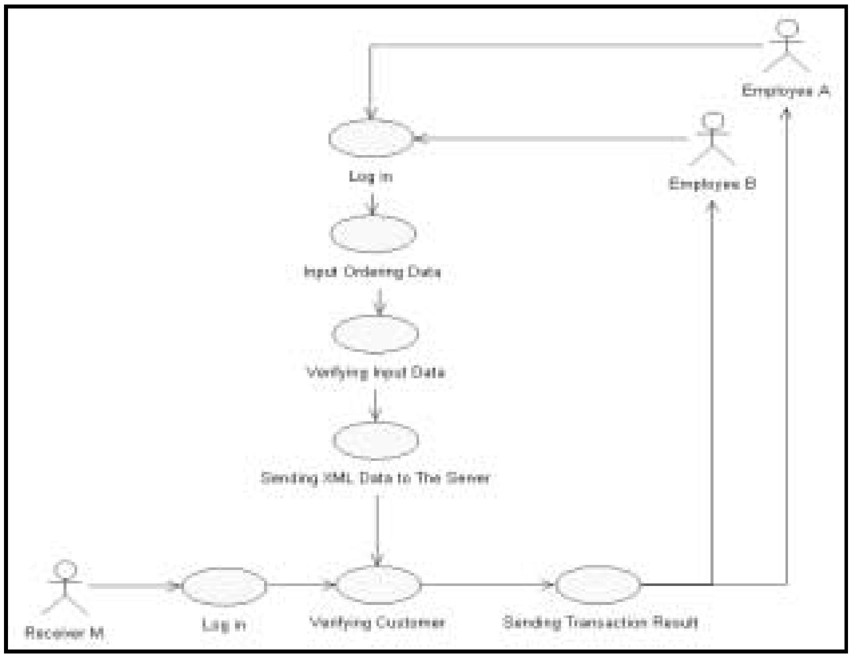


Figure. 3-4 Use-Case Diagram of Auto parts Order and Delivery

3.2.2 Alternative Flow of Events

If an employee omits an essential item in the order then the input ordering data will not be processed. If the server does not properly verify the digital signature that was received, an exception is generated and the employee responsible for verifying each order terminates the order.

3.2.3 Special Requirements

The DTD (Document Type Definition) must exist in a Global Repository to verify XML data. Also, XSL (eXtensible Stylesheet Language) must also exist in the repository to enable expression of XML data.

3.2.4 Preconditions

First, the order and delivery system of employee A, B and Receiver M go ahead of all. Second, Use-Case Diagram needs to maintain Internet connection because of Internet based programs. Third, the database server has to exist to manage transaction data.

3.2.5 Post conditions

Though the Use-Case is closed, the document ordered and delivered off-Line has to be processed continuously. And, an order and delivery business of physical goods must be processed continuously.

4 Implementation and Tangible Benefits

Figure 4-1 is the context diagram of the system. The proposed B2B system exchanges only server programs and encoded files while communicating with other business companies. As shown in Figure 4-1, the relationship between the Web server of XML system and the client based on the transaction model is removed. This method transmits a file of XML documentation by using a socket on the application. The DTD and XSL are saved in a global repository and are used by the web browser's parser for validation based on the information in XML documentation itself that is sent and received as shown in Figure 4-2.

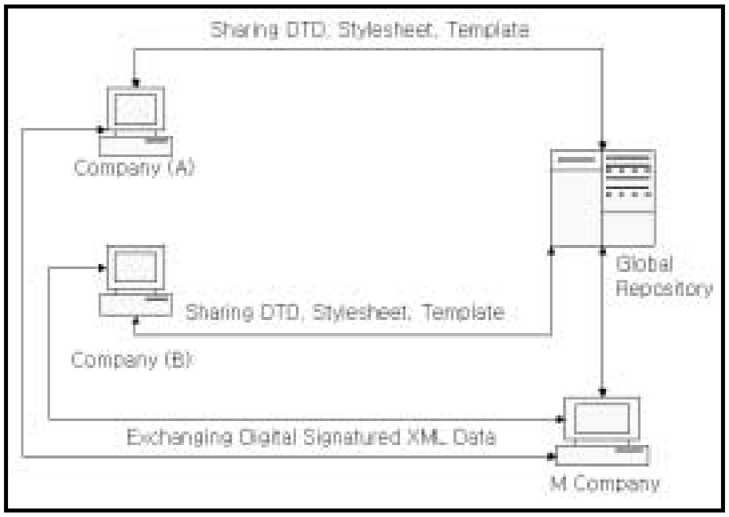
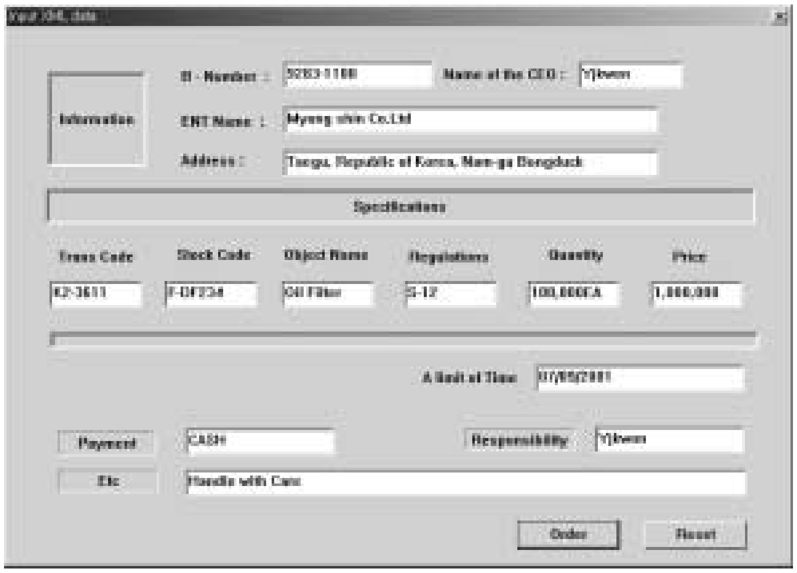

Sharing DTD, Stylesheet, Template
Company (A)
Global Repository
Sharing DTD, Stylesheet, Template
Company (B)
Exchanging Digital Signatured XML Data
M Company

Figure. 4-1 The B2B System Context Diagram.

```
<?xml version='1.0' encoding='    euc-kr'? >
<!DOCTYPEclients SYSTEM ' http ://203.244.152.45/      clients.dtd'  !>
```

Figure. 4-2 Information with which XML documentation itself is preserved

The first row of Figure 4-2 shows that the Korean language is used as the encoding language. The version number is also there indicated. The second row specifies the name of the XML documentation and the location of the global repository where the DTD is saved.



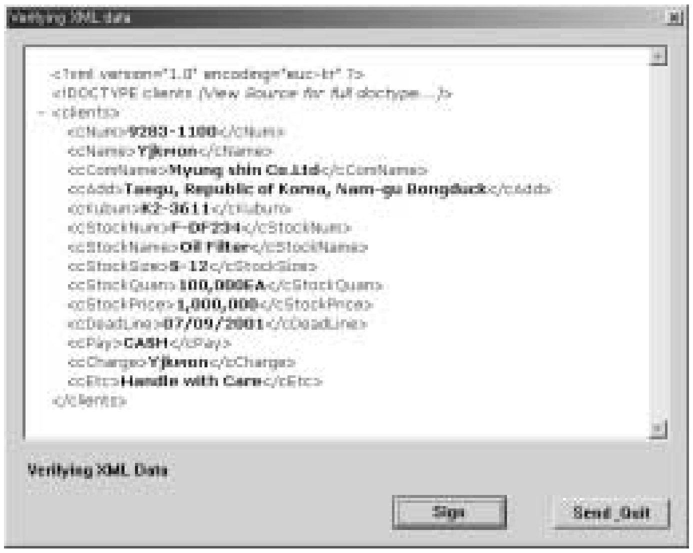
ID - Number :	50783-1108	Name of the CEO :	Y.Jhwan		
ENT Name :	Myung shin Co.Ltd				
Address :	Taegu, Republic of Korea, Nam-gu Bongduck				
Trans Code	Stock Code	Object Name	Regulations	Quantity	Price
K2-3611	F-OF234	Oil Filter	S-12	100,000EA	1,000,000
A limit of Time	07/05/2001				
Payment	CASH	Responsibility	Y.Jhwan		
Etc	Handle with Care				

Figure. 4-3 Delivery data input screen

4.1 Client Application

Only certified employees can execute the client program for security reasons. Figure 4-3 shows the screen to input the data that is to be delivered. The data is stored in XML format on the client when the Order button is clicked and is validated against the DTD.

The screen in Figure 4-4 confirms validation by web browsers on the client application by referring to the DTD that was saved in the global repository. If the input data wasn't in accord with the DTD the error is displayed in Figure 4-4. By clicking the Sign button in Figure 4-4, digital signature files are created that are translated to XML data (Figure 4-5) for transmission.



```
  <?xml version="1.0" encoding="euc-kr" ?>
  <!DOCTYPE clients (View Source for full doctype...)>
- <clients>
    <cNum>9283-1100</cNum>
    <cName>Yjkwon</cName>
    <cComName>Myung shin Co.Ltd</cComName>
    <cAdd>Taegu, Republic of Korea, Nam-gu Bongduck</cAdd>
    <cKubun>K2-3611</cKubun>
    <cStockNum>F-OF234</cStockNum>
    <cStockName>Oil Filter</cStockName>
    <cStockSize>S-12</cStockSize>
    <cStockQuan>100,000EA</cStockQuan>
    <cStockPrice>1,000,000</cStockPrice>
    <cDeadLine>07/09/2001</cDeadLine>
    <cPay>CASH</cPay>
    <cCharge>Yjkwon</cCharge>
    <cEtc>Handle with Care</cEtc>
  </clients>
```

Figure. 4-4 The screen that confirms input data in client

Figure 4-6 is a digital signatures file that applies the digital signatures algorithm (MD5 hash algorithm) of the RSAEuro encoded library and adopts a 1024-bit password key to a message digest file. Figure 4-6 shows a screen that is opened as a text editor. It is formalized as a length of 64byte. If we click the Send-Quit button (Figure 4-4), XML data and the digital signature are transmitted to the server.

4.2 Server Application

On the server side, we receive an XML file and a digital signature file simultaneously from the client through the course of Figure 4-5. Decoding the digital signature file with the public key of sender (obtained from an earlier exchange) produces the message digest. After decoding the message digest and the digital signatures that were created from XML data, we compare these message digests with each other. If the compared results are the same, the transaction is certified.

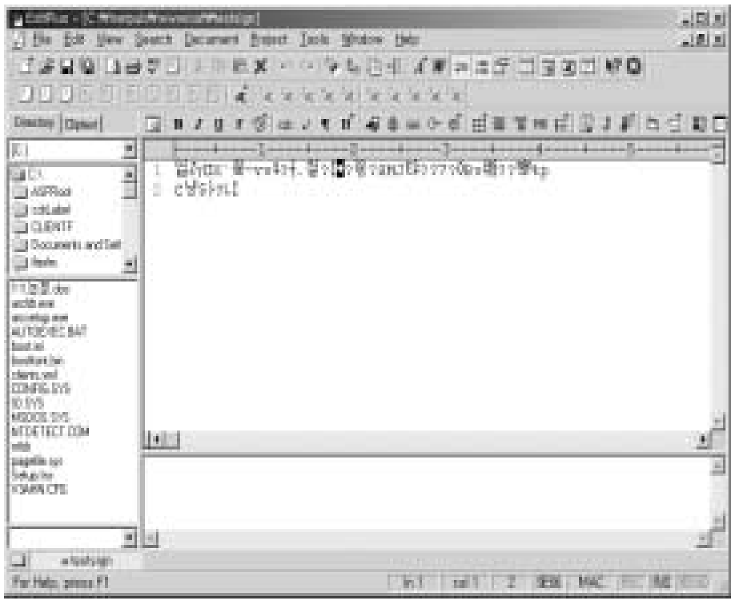
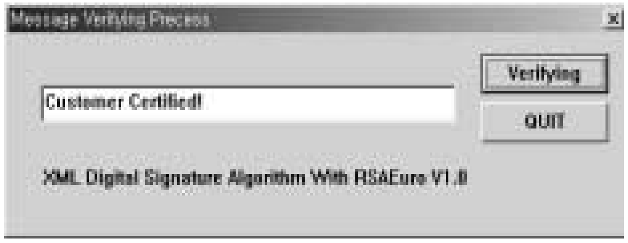


Figure. 4-5 Digital signature of 64-byte length

Figure 4-6 is the screen to confirm the digital signature.

Figure 4-6 confirms the digital signature by the algorithm that is in the RSAEuro library, and when the received digital signature is inconsistent, an error message is displayed while at the same time the operation is stopped. Figure 4-7 is the employee's certification screen that enables access to the global repository which can then be used to identify the received XML data at the server.



Customer Certified!

XML Digital Signature Algorithm With RSAEuro V1.0

Figure. 4-6 Digital signature certification screen

Figure 4-8 is the screen that displays the XML documentation that was delivered, after the digital signature has been verified and the data validated against the DTD and the XSL in the global repository.

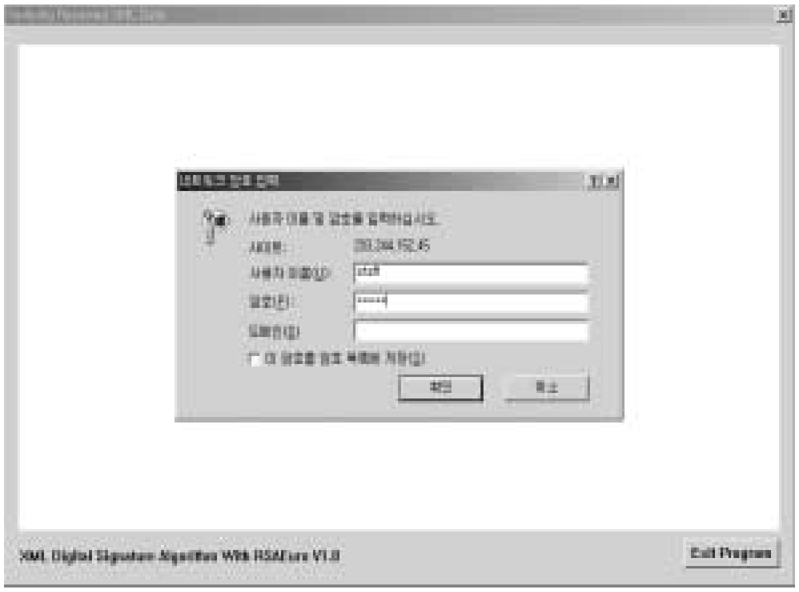
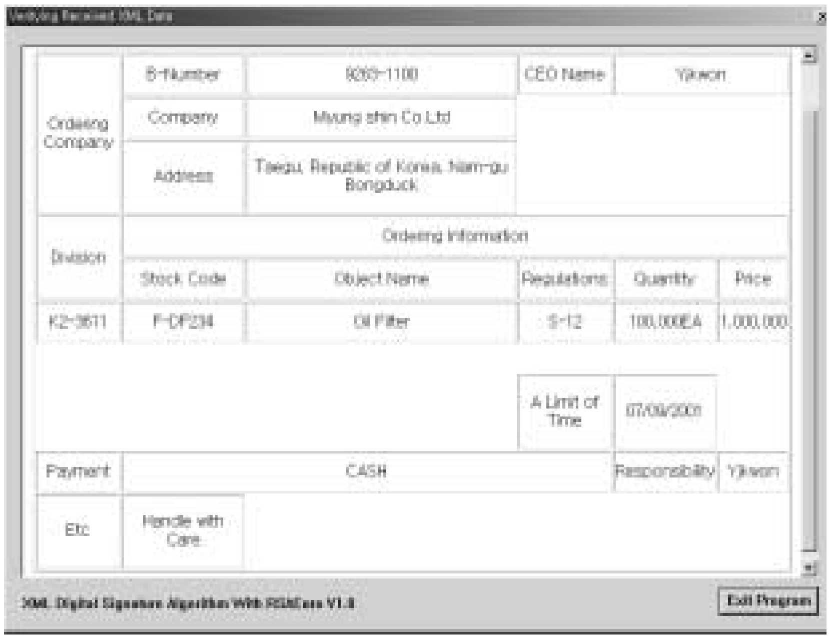


Figure. 4-7 Employee's certification screen

4.3 Tangible Benefits

First, communication between the developers and the users was harmonious. That is, by producing conceptually and physically visualized and specified output using Use Case Diagram and XML, the communications between developers and employees were improved.


Verifying Received XML Data
XML Digital Signature Algorithm With RSAEuro V1.0
Exit Program

Ordering Company	B-Number	9203-1100	CEO Name	Yikwon	
	Company	Myung shin Co Ltd			
	Address	Taegu, Republic of Korea, Nam-gu Bongduck			
Division	Ordering Information				
	Stock Code	Object Name	Regulations	Quantity	Price
K2-3611	F-DF234	Oil Filter	S-12	100,000EA	1,000,000
			A Limit of Time	07/09/2001	
Payment	CASH			Responsibility	Yikwon
Etc	Handle with Care				

Fig. 4-8 Received delivery data screen

Second, processing time of the business and employee's transaction data was reduced. In the past, employees managed order and delivery documents by writing directly on paper. Moving the order and delivery system onto the Internet reduced the time needed to process business/employee's transaction data.

Third, the processing cost was reduced. This conclusion is based on the annual salary of an employee. Namely, we compared between the times when the B2B e-commerce system wasn't and was implemented. There was a cost savings of roughly $12,000 USD per month.

Fourth, the accuracy of transaction data was improved resulting in less rework. The system was more responsive (i.e., interactive) and therefore employees were less apt to commit errors (the user interface provided a more intuitive environment). For these reasons, the accuracy of transaction data input to the system (e.g., auto parts codes were tabulated) was greatly improved.

Fifth, efficiency of the system was improved. Unit components were designed by separating concerns into functions. In this way, the system designed was extensible and reusable. Perfective and corrective maintenance were greatly simplified by this component based object-oriented approach. Consequently our system evolved into a more usable and ultimately more efficient system.

Sixth, the development life cycle for the system was shortened by applying our development methodology, a combination of SDLC (System Development Life Cycle) and PDLC (Prototyping Development Life Cycle).

Seventh, the reliability of the system was improved. Errors were reduced because employees retrieved the relevant auto parts (and other information) from tables stored in memory and without input through keyboard when they worked on an order/delivery.

Eighth, we ensured security of transmitted data by using digital signatures while transmitting data. This enabled authentication of identity and repudiation of forgery/falsification.

5 Conclusions and Future Work

In this paper, we have described the implementation of a B2B e-commerce system for the order and delivery of auto parts. Requirements analysis was carried out using scenarios and formalized using Use Case diagrams. Digital signatures were employed for implementing security. Order and delivery documentation was made simple and clear through the use of XML. Eight critical success factors were used as controlling parameters while building the application.

The implementation approach preserved and promoted harmonious communication between the users and developers. The end product achieved a reduction in the processing time of transactions, reduced processing cost including improved accuracy, efficiency and security of transmitted data. Also, the strategy seemingly shortened the System Development Life Cycle.

In general, the typical B2B e-commerce characteristic is heterogeneity, especially in the types of product information that is needed. Strategies for the successful implementation of such systems depend on both the standardization and the accommodation of such heterogeneity. For future work, the tools that we are considering for accomplishing this include XML, DTD (Document type definitions), ICE (Information and Context Exchange), and CBL (Common Business Library) [34]. Further research on the convergence of XML technologies and software engineering will also be done.

Wolfgang Emmererich et al, [35] claim that the strengths of middleware and markup languages are complementary. They expect this combination to be used in the future for distributed systems where complex data structures need to be transmitted between distributed off-the-shelf components and semantic transformations performed. Soon-Kyeong Kim and David Carrington [36] present a formal basis for syntactic structures and semantics of core UML class constructs, and also provide a basis for reasoning about UML class diagrams in their paper. They translate UML class constructs to Object-Z constructs as being based on this formal description.

Dupuy et al, [37] present RoZ, an automated tool for generating a Z formal specification An XML web environment for projecting integrated formal models (TCOZ: Integrated model of state-based Object-Z and event-based Timed CSP) to UML diagrams and several ways of using UML for designing effective software architectures have been suggested [38].

In the future, we plan on building a B2B e-commerce system using Advanced Visual Modeling Technique and Object-Z. Study of a natural language processing technique for semantic modeling of user's requirements will also be undertaken.

References

1. Mylopoulos, J., et al., *Exploring Alternatives during Requirements Analysis.* IEEE Software. **18**(1): p. 92-96.
2. Guell, N., D. Schwabe, and P. Vilain. *Modeling Integrations and Navigation in Web Application*. in *ER 2000 Workshop*. 2000: Springer Verlag. LNCS 1921. p. 115-127.
3. Pomerol, J.-C., *Scenario development and practical decision making under uncertainty.* Decision Support Systems, 2001. **31**(2): p. 197-204.
4. Whittle, J. and J. Schumann. *Generating Statechart Designs From Scenarios*. in *ICSE*. 2000. Limerick, Ireland: ACM. p. 314-323.
5. Hands, J., et al., *An Inclusive and Extensible Architecture for Electronic Brokerage.* Decision Support Systems, 2000. **29**: p. 305-321.
6. Papazoglou, M.P., P. Riebbers, and A. Tsalgatidou, *Integrated Value Chains and their Implications from a Business and Technology Standpoint.* Decision Support Systems, 2000. **29**(4): p. 323-342.
7. Rohm, A.W. and G. Pernul, *COPS: A Model and Infrastructure for Secure and Fair Electronic Markets.* Decision Support Systems, 2000. **29**(4): p. 343-355.
8. Cheng, E.C., *An Object-Oriented Organizational Model to Support Role-Based Access Control in Electronic Commerce.* Decision Support Systems. **29**(4): p. 357-369.
9. Zacharia, G., A. Moukas, and P. Maes. *Collaborative Reputation Mechanisms for Electronic Marketplaces*. in *HICCS' 1999*. 1999. Maui, Hawaii: IEEE Computer Society. p. 371-388.
10. Tan, Y.-H. and W. Thoen, *INCAS: A Legal Expert for Contract Terms in Electronic*

Commerce. Decision Support Systems, 2000. **29**(4): p. 389-411.

11. Quatrani, T., Visual Modeling with Rational Rose and UML. 2 ed. 1998, Boston: Addison Wesley Longman Inc. 240.
12. Post, G. and A. Kagan, *OO-CASE tools: an evaluation of Rose.* Information and Software Technology, Elsevier Science, 2000. **42**(6): p. 383-388.
13. Bergner, K., A. Rausch, and M. Sihling. *A Critical Look upon UML 1.0.* in *The Unified Modeling Language - Technical Aspects and Applications*. 1998: Physica Verlag. p. 79-92.
14. Wolf, M., R. Burkhardt, and I. Philippow. *Software Engineering Process with the UML.* in *UML Workshop*. 1997. Mannheim, Germany: Physica Verlag. p. 271-280.
15. Korthaus, A. *Using UML for Business Object based Systems Modeling*. in *The Unified Modeling Language - Technical Aspects and Applications*. 1998: Physica Verlag. p. 220-237.
16. Korthaus, A. and S. Kuhlins. *BOOSTER*Process, A software development Process Model Integrating Business Object Technology and UML.* in *The UML'98*. 1998. France: Springer Verlag. p. 215-239.
17. Kande, M.M. *Applying UML to Design an Inter-domain Service Management Application.* in *The UML'98*. 1998. France: Springer Verlag. p. 200-214.
18. Dykman, N., M. Griss, and R. Kessler. *Nine Suggestions for Improving UML Extensibility.* in *The UML'99*. 1999. Colorado, USA: Springer Verlag. p. 236-248.
19. Allen, P. *A Practical Framework for Applying UML.* in *The UML'98*. 1998. France: Springer Verlag. p. 419-433.
20. Gomaa, H. *Object Oriented Analysis and Modeling for Families of Systems with UML.* in *ICSR*. 2000. Vienna, Austria: Springer Verlag. p. 89-99.
21. Kivisto, K. *Considerations of and Suggestions for a UML-Specific Process Model.* in *The UML'98*. 1998. France: Springer Verlag. p. 294-306.
22. Back, R.-J., L. Petre, and I.P. Paltor. *Amalysing UML Use Cases as Contracts*. in *The UML'99*. 1999. Colorado, USA: Springer Verlag. p. 518-533.
23. Overgaard, G. and K. Palmkvist. *A Formal Approach to Use Cases and Their Relationships.* in *The UML'98*. 1998. France: Springer Verlag. p. 406-418.
24. Menasce, D., A., *Web Performance Modeling Issues.* The International Journal of High Performance Computing Application, 2000. **14**(4): p. 293.
25. Hosoi, K., *Advanced B2B Procurement on the Internet.* FUJITSU Science Journal, 2000. **32**(2): p. 226-231.
26. Papazoglou, M.P. and A. Tsalgatidou, *Business to business electronic commerce issues and solutions.* Decision Support Systems, 2000. **29**(4): p. 301-304.
27. Tumolo, M., *Business-to-business exchanges.* Information Systems, 2001. **18**(2): p. 54-62.
28. Lederer, A.L., et al., *The technology acceptance model and the World Wide Web.* Decision Support Systems, 2000. **29**(3): p. 269-282.
29. Blair, B.T. and J. Boyer, *XFDL: Creating Electronic Commerce Transaction Records Using XML.* Computer Network - The International Journal of Computer and Telecommunication

Networking, 1999. **31**(11-16): p. 1611-1622.

30. Marca, D.A. and B.A. Perdue. *A Software Engineering Approach and Tool Set for Developing Internet Applications*. in *ICSE*. 2000. Limerick, Ireland: ACM. p. 738-741.

31. Denton, J.W., *Using Web-based Projects in a Systems Design and Development Course.* Journal of Computer Information Systems, 2000. **40**(3): p. 84-87.

32. Chen, J.Q. and R.D. Heath, *Building Web Applications.* Information Systems, 2001. **18**(1): p. 68-79.

33. Zundorf, A. *From Use-Case to Code - Rigorous Software Development with UML*. in *ICSE*. 2001. Toronto, Canada: ACM. p. 711-712.

34. Ng, W.K., G. Yan, and E.P. Lim, *Standardization and Integration in Business-to-Business Electronic Commerce.* IEEE Intelligent Systems, 2001. **16**(1): p. 12-14.

35. Emmererich, W., E. Ellmer, and H. Fieglein. *TIGRA-An Architecture Style for Enterprise Application Integration*. in *ICSE*. 2001. Toronto, Canada: IEEE Computer Society, ACM. p. 567-576.

36. Kim, S.-K. and C. David. *Formalizing the UML Class Diagram Using Object-Z*. in *The UML'99*. 1999. Colorado, USA: Springer Verlag. p. 83-98.

37. Dupuy, S., Y. Ledru, and M.C. Peccoud. *An Overview of RoZ: A Tool for Integrating UML and Z Specifications*. in *International Conference on Advanced Information Systems Engineering*. 2000. Sweden. p. 417-430.

38. Dong, J.S. *State, Event, Time and Diagram in System Modeling*. in *ICSE*. 2001. Toronto, Canada: IEEE Computer Society, ACM. p. 733.