The Journal of Korean
Institute of CALS/EC
Vol. 7, No. 2, August 2002

효율적인 e-Procurement를 위한 XML 메시징 시스템 개발

임종선*, 주경수*

Developing of XML Messaging System for efficiently e-Procurement

Lim Jong-Seon, Joo Kyung-Soo

Abstract

Because XML is a W3C standard and has characteristics like platform-independent, it has a critical role in e-commerce. Business rules and procedures should be standardized for efficient B2B integration. But a lot of companies has its own XML documents instead of standard documents. Therefore many organizations try to make standards for e-commerce based on framework. In this paper, XML messaging system is designed and implemented. Client request service in a XML and server return response in XML in this system. So we can more easily and efficiently build e-Procurement system based on this XML messaging system.

KeyWord *: e-Procurement, B2B, XML, Messaging System*

* 순천향대학교 일반대학원 전산학과
** 순천향대학교 정보기술공학부 교수

1. 서론

현재 많은 전자상거래 시스템에서 XML을 이용하고 있으며, 이 전자상거래에서 사용되는 문서는 XML을 이용하여 작성하고 보내어 진다. 그러나 이러한 기업들이 표준화된 XML 문서를 사용하는 것이 아니라 기업의 실정에 맞게 시스템을 구축하여 기업간의 XML 문서 전달에 많은 문제점이 나타나고 있다. 대표적인 예로서 RosettaNet에 따라서 만든 XML 문서를 사용하는 기업과 ebXML에 따라서 만든 XML 문서를 사용하는 기업간의 거래에 XML 문서를 사용한다면 양자 사이에 XML 문서를 변환하는 작업이 필요하다. 이러한 B2B 양자간에 직접적인 거래를 위해서 변환과 문서 전달 기능을 대행해 주는 것이 XML HUB이다. 본 논문에서는 XML 메시징 시스템을 이용하여 효율적인 e-Procurement를 구현하고자 하였다[8].

본 논문에서는 XML로 서비스를 요청하고, 서버가 해당 서비스를 처리하는 시스템을 설계 및 구현하였다. 서비스를 요청할 경우 ebXML을 이용하여 메시지를 작성한 후, 전송하고, 그 메시지를 기반으로 서비스를 처리한다. 본 논문의 2장에서는 관련기술을 소개하고, 3장에서는 XML 메시징 시스템의 구성과 기능에 대하여 설명하며, 4장에서는 메시징 시스템의 분석 및 설계를 설명한다. 그리고, 5장에서는 XML 메시징 시스템을 구현한 화면을 보여주며, 6장에서는 결론 및 향후 연구방향을 소개한다.

2. 관련 연구 및 기술

2.1 관련 연구

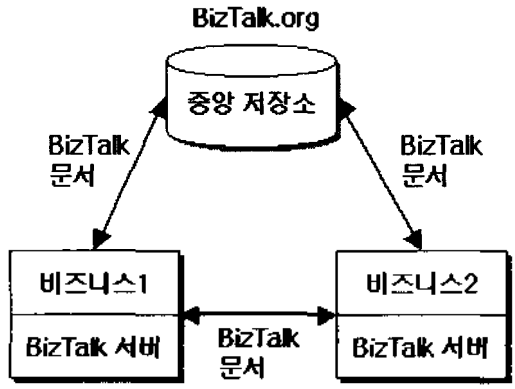
현재까지 XML을 이용한 메시지 교환 시스템은 기존의 전통적인 EDI(Electronic Data Interchange) 시스템을 어떤 방법으로 XML로 메시지를 교환할 것인가에 초점을 맞추어져 있다. EDI/XML 시스템의 성공적인 구축은 '데이터 교환 모델'을 위하여 XML을 사용하고, '모습을 표현하기 위하여' XSL(XML Style Language)을 이용하며, 전통적인 EDI와 쉬운 통합 방안을 지니기 위하여 DTD를 사용하고, 문서중심의 조회와 처리가 가능케 하며, 타 정보 시스템과의 연동이 가능하도록 개발하는 것이다[6]. 이러한 문제를 해결하기 위하여 전자상거래 표준 기술로서 XML이 등장하게 되었으며 광범위한 영역에서 입지를 확고히 해 나가고 있다. 또한 여러 가지 장점을 가지고 있기 때문에 다양한 분야에서 적용되고 있다[7].

RosettaNet은 정보기술 및 전자부품의 SCM(Supply Chain Management)을 위한 XML 기반의 비즈니스 표준을 개발하기 위해 1998년에 결성된 컨소시엄으로, 350여 개 이상의 업체가 참여하고 있다. RosettaNet에서는 비즈니스 프로세스를 정의하고 데이터 교환을 위한 기술규격을 제공하고 있다.

RosettaNet에서 정의하고 있는 표준으로는 Dictionary, RNIF(RosettaNet Implementation Framework), PIP(Partner Interface Process)가 있다. RosettaNet Dictionary는 크게 비즈니스 부분과 기술 부분으로 나뉘어진다. 디렉토리는 비즈니스에서 사용되는 공통된 용어

와 속성들을 표준화 한 것으로, 비즈니스를 위한 공통 플랫폼을 제공하여 개별 기업의 중복되는 투자와 노력을 절감하는 역할을 한다. RNIF는 RosettaNet 표준에 대한 시스템의 신속하고 효과적인 개발을 위한 가이드라인과 통신 프로토콜, 보안에 관련된 부분을 명시하고 있다. 즉, XML과 HTML을 사용하여 거래 파트너 사이에 정보를 교환하는 방법을 정의한다. RosettaNet에서 제공하는 표준중에서 가장 중요한 것은 PIP이며, PIP는 거래 파트너와 인터페이스 할 수 있는 비즈니스 프로세스를 정의하고 있다. PIP는 크게 6개의 클러스터로 구성이 되어 있으며, 각 클러스터는 다시 세그먼트 단위로 구분되고, 세그먼트 안에 하나의 PIP가 정의된다. RosettaNet에서는 비즈니스 모델, Dictionary, RNIF가 PIP의 입력이 되며, PIP가 거래 당사자들에게 배포되고, 각 기업에서 해당 소프트웨어를 개발하는 로드맵 역할을 한다. 각 PIP는 모든 비즈니스 로직, 메시지 흐름, 메시지 내용을 포함한다[4].

Microsoft사는 1999년 'BizTalk Initiative'라는 이름을 가지고 XML 기반의 B2B 전자상거래 솔루션을 발표하였다. BizTalk은 XML을 이용하여 기업 내부 또는 기업간 응용 프로그램 통합을 효과적으로 할 수 있는 기반을 제공해 줌으로써 보다 빠르게 전자상거래를 구축할 수 있는 방법을 제시하였다. BizTalk Initiative는 BizTalk Framework, BizTalk.org, BizTalk server의 세 가지 요소로 이루어져 있으며, 그 구성은 <그림 1>과 같다. BizTalk을 바탕으로 하는 B2B 전자상거래 시스템에서는 거래자들이 비즈니스 문서들을 교환하기 위하여 BizTalk server를 이용한다. 기업에서 구매 요청과 같은 비즈니스 이벤트가 발생할 경우 기업의 응용 프로그램은 XML 기반 비즈니스 문서를 작성하기 위하여 이 비즈니스 이벤트에서 참조할 BizTalk Schema를 사용한다.



<그림 1> BizTalk Initiative의 구조

BizTalk은 B2B 전자상거래와 기업 내부나 인터넷을 통하여 다른 기업간 비즈니스 프로세스를 자동화시키는 플랫폼을 제공한다. BizTalk 솔루션을 이용하여 BizTalk 지원 응용 프로그램과 BizTalk server 시스템을 구축하고, 거래자들간에 교환되어질 문서의 양식을 작성하는데 필요한 BizTalk 스키마를 결정함으로써 기업들은 전자상거래 비즈니스를 지원할 수 있게 된다[5].

2.2 ebXML

ebXML(electronic business XML)은 EDI를 표준화한 UN/CEFACT와 기업과 산업 표준 저장소를 운영해 온 OASIS가 주축이 되어 1999년 11월부터 시작하여 18개월 동안 만들어진 표준이다. ebXML은 특히 XML을 이

용한 인터넷 기반 글로벌 전자 상거래가 가능하도록 하기 위한 국제 표준으로, 국내외 거래에 모두 적용될 수 있어 주목을 받고 있다.

ebXML은 ebXML의 전체 구조를 보여주는 Technical Architecture, 거래 절차를 기술하는 Business Process, 거래에 이용되는 공통의 용어와 그 의미를 정의하는 Core Component, 거래에 필요한 정보를 저장하고 검색하는데 필요한 Registry and Repository, 거래에 필요한 합의 사항 및 합의 방법을 규정한 Trading Partner, 거래 문서의 전송을 위한 Transport/Routing and Packing 등에 대해 2001년 5월 공식 표준을 발표했다.

ebXML은 항목과 문서 하나하나를 미리 정해 놓기보다는 체계적인 방법론을 제시함으로써 전 산업에 두루 응용될 수 있도록 하고 있다. 물론 ebXML 표준에 따라 만들어진 전자 문서끼리는 서로 교환할 수도 있다. 또한 표준을 포함하여 전자 거래를 위해서 필요한 각종 컨텐츠를 저장하고 검색할 수 있는 레지스트리를 ebXML에서는 더욱 명확하게 정의하고 있다. 레지스트리의 시스템 인터페이스, 등록과 검색 방법을 포함해서 메타 모델까지 정의함으로써 전 세계적으로 ebXML 레지스트리의 연결과 정보 공유를 보장하고 있다[12].

2.2.1 ebXML의 기반 기술

ebXML에서 필요한 인터넷 응용 기술은 전송 프로토콜과 정보를 표현하는 XML에 관련된 부분들이다. 전송 프로토콜에서는 HTTP, SMTP와 같은 전송 프로토콜, SOAP와 같은 전송 응용 프로토콜이 있다. 그리고 정보를 표현하는 XML은 XML과 연관된 각종 기술들인 XLink, XPath, XML Signature, XML Namespace 등이 활용되고 있다.

ebXML에서 필요한 모델링 기법에는 일반적인 비즈니스와 시스템을 모델링하는 기법이 있다. 거래 절차와 같은 일반적인 비즈니스를 모델링하기 위하여 UMM, UML와 같은 표준화된 모델링 기법을 활용하고 있다. 그리고 정보 모델링에는 XML DTD, XML Schema 등이 활용되고 있다[8][12].

ebXML 메시징

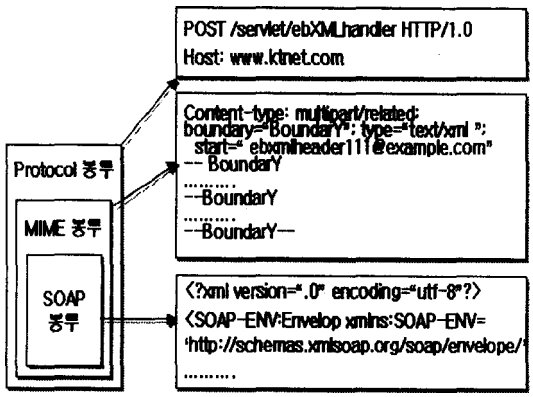
문서를 교환하기 위해서 가장 중요한 프로세서는 메시지를 처리하기 위한 프로세스이다. 즉, 규격화된 메시지를 만들거나 자신에게 들어온 메시지를 처리하여 문서의 내용을 추출하며, 그 내용을 이해하여 업무를 처리하는 프로세서에게 넘겨 줄 수 있는 프로세스가 필요하다. 이러한 프로세서는 메시징 서비스에 속하는 것이며, 메시지 헤더의 처리와 다른 프로세서의 호출, 암호화, 복호화, 에러, 에러 처리, 라우팅, 변환 등의 기능을 수행한다.

ebXML 메시지의 프로토콜 봉투

ebXML 메시지를 전송하기 위한 전송 프로토콜을 선택한 후에 그에 맞는 프로토콜 봉투의 내용을 담아야 한다. 일반적으로 가장 많이 활용되고 있는 HTTP 프로토콜 봉투에서는 기본적으로 POST라는 정보를 기입하여 준다. POST은 서블릿이나 CGI 같은 메소드를 호출하여 메시지를 처리한다. 이렇게 지정된 메소드가 불려지면 그 내부에서는 MIME 봉투를 파싱하고 처리한다.

ebXML 메시지의 MIME 봉투

HTTP 프로토콜와 SMTP 프로토콜은 MIME을 적용한다. 전송 프로토콜에 따라 프로토콜 봉투의 내용은 달라질 수 있지만 MIME 봉투에 담기는 내용은 달라지지 않는다는 것이다. 인터넷의 전자 메일에서는 다양한 형식의 데이터를 상호 주고 받을 수 있다. 이런 다양한 형식을 자동으로 인식하여 처리하려면 컨텐츠의 형식을 지정하는 표준화된 방식이 필요하다. MIME이란 바로 컨텐츠의 형식을 정의하는 방식으로 XML 문서를 첨부한 경우에는 MIME 타입이 "text/xml"이라고 설정되고, 이로 인해 그림이나 영상파일들을 서로 주고받을 수 있다.



<그림 2> ebXML의 메시지 구조와 메시지 내용

ebXML 메시지의 SOAP 봉투

프로토콜 봉투와 MIME 봉투는 나름대로의 규격화된 표기 방식을 가지고 있지만, SOAP 봉투는 봉투 자체도 Header와 Body 부분으로 구성되어있는 XML 형태의 봉투이다. 즉, 봉투 자체가 XML 문서로 이루어졌다. 이런 봉투에서 Header는 SOAP 봉투를 처리하기 위한 Service나 보내는 사람, 받는 사람 등에 대한 정보를 담고 있으며, Body 부분에서는 다음 파트에 오게 되는 내용에 대한 정보를 담고 있다.

2.3 전자상거래(e-Business)

인터넷이 발달하면서 가장 눈여겨볼 부분은 전자상거래의 등장이며, 이는 기존의 전통적인 상거래 시스템을 대체할만한 가치를 가지고 있다. 전자상거래의 등장은 새로운 기술의 등장뿐만 아니라, 그 동안 우리가 사용해 왔던 비즈니스 모델을 바꾸어야 하는 상황을 만들어 내고 있으며, 각 회사들은 그들의 새로운 비즈니스 모델을 만들고 전자상거래의 이점을 활용하고자 많은 노력을 기울이고 있다.

전자상거래는 여러 가지의 카테고리로 나뉘어질 수 있는 서로 다른 애플리케이션을 만들어냈으며, 이러한 형태의 체계화가 전자상거래의 일반적인 접근 방식이다. 성공적인 전자상거래 애플리케이션은 다음에 설명하는 것 중 한가지 이상의 서비스를 제공해야 한다[2].

비즈니스 to 고객 (B2C)

B2C 웹사이트의 보편적인 예로 제품이나 서비스를 고객이 직접 구매할 수 있도록 한 애플리케이션을 들 수 있다. B2C는 중간 마진을 줄이기 위한 노력이며, 기업이 실제 상점과 온라인 상점의 개설 비용 등을 고려하여 선택한 것이다. B2C 애플리케이션은 인터넷에서 상당히 자주 볼 수 있으며, 신용카드 결제를 통해 서적, 음반, 영화 예매 등의 서비스를 제공하는 사이트들이 대표적이다.

비즈니스 to 비즈니스(B2B)

비즈니스 사이의 관계를 유지하는 것은 빠르게 변하는 시장에서 살아남기 위해 꼭 필요한 일이 되어가고 있으며, 이러한 B2B 애플리케이션은 새로운 가능성을 제기하고 있다. B2B 구매는 기업에 점차 큰 위치를 차지하고 있으며, 시장 또한 점점 커질 것이다. B2B에서의 이슈는 애플리케이션을 통합 할 때 이익이 되는 점과 손해가 되는 점이 존재하기 때문이다. 또 애플리케이션의 다른 이슈로는 보안, 속도, 안전성, 회사간의 상호 작용을 돕는 여러 가지 요소들이 있다.

비즈니스 to 비즈니스 to 고객(B2B2C)

B2B2C는 전자상거래 분야에서 잘 정의된 부분 중에 하나이다. 이것은 B2B를 사용하여 B2C를 지원하는 회사를 지원하는 것을 의미하며, B2B가 상업적으로 성공했으나 B2C의 기반이 잡혀있지 않은 경우에 반드시 필요하다. B2B는 B2C 회사가 가격을 절감하고 그들의 B2C 서비스를 향상시킬 수 있도록 도와주며, 비즈니스가 서로 교류하여 그 사이에서 여러 가지 이익을 창출하고, B2B의 지원하에서 B2C의 이익을 더 높이는 효과가 있다. 온라인 카탈로그를 서로 연결하여 고객에게 보여주는 사이트들이 B2B2C의 좋은 예라 하겠다.

고객 to 고객, 고객 to 비즈니스 to 고객 (C2C, C2B2C)

C2C는 상거래 시스템의 새로운 분야이며, 고객간의 거래를 회사가 지원해 주는 방식으로 이루어진다. 이때 회사는 약간의 거래 차액, 광고 효과를 얻게 된다. C2C 애플리케이션은 상당한 각광을 받고 있으며, 사람들 사이의 저렴하고 실용적인 거래를 장려한다는 면에서 매우 바람직하다 할 수 있다.

또한 C2B2C는 고객과 고객의 교류하는 과정에서 비즈니스를 매개체로 활용하는 것이다. C2B2C의 예로 중고차 거래 사이트를 들 수 있는데, 여기서는 고객간의 자동차 거래뿐만 아니라 회사에서 고객에게 판매하는 중고차의 목록 또한 볼 수 있다.

2.4 e-Procurement

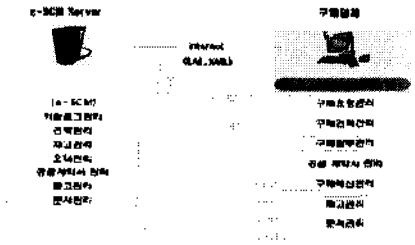
인터넷 환경을 이용하여 구매 요청, 승인, 주문, 운반, 결재 및 인도에 이르는 일련의 프로세스를 전략적으로 관리하는 것을 의미한다. 즉, e-Procurement는 주문에서 인도에 이르는 전체 구매 프로세스를 인터넷 환경하에서 유기적으로 연계하고, 동시에 구매사와 공급사간의 공조를 이루어 구매 업무의 최적화를 도모하려는 전략적 기법이다.

e-Procurement를 통해 기업들이 추구하는 목표는 크게 세 가지로 분류될 수 있다. 첫째, 구매 프로세스의 개선으로 구매 비용 절감과 납기의 단축 등의 목표를 실현한다. 둘째, 자사의 구매시스템과 기존 운영시스템 및 공급사 시스템과의 기능적 통합을 통하여 구매 업무의 효율성을 높인다. 셋째, 구매 활동의 전략적 역량 강화를 통하여 기업 전체 목표에 부응하는 전략적 구매 업무를 수행한다.

또한 e-Procurement 도입은 구매원가 및 구매관리비용 절감, 구매프로세스의 자동화 효율적 재고 관리, 작업현장 임의구매 통제 강화, 구매 이행 주기 단축, 구매 담당직원의 전략적 업무 수행 전환, 투명한 구매 실현 단

기간 내 효율적 ROI 창출, 공급자 관계망 관리 등의 다양한 효과를 거둘 수 있다.

<그림 3>은 e-Procurement의 시스템 구성을 나타낸 것이다. e-SCM 서버는 구매업체 간에 XML, EAI 같은 문서를 인터넷으로 주고받는다. 또한 주고받은 문서는 서버에 일괄적으로 저장되어 보관되며, 별도의 관리가 필요 없다[13].



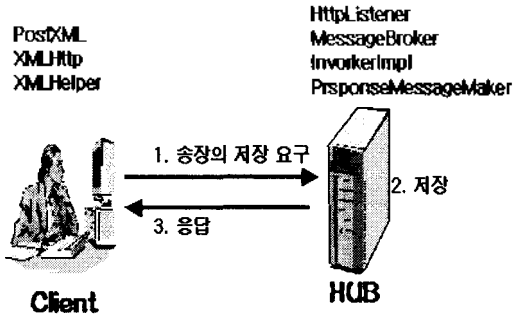
<그림 3> e-Procurement 시스템 구성

3. XML 메시징 시스템

3.1 XML 메시징 시스템의 구성

메시징 시스템은 서로 다른 시스템간에 특정 기능을 수행하기 위한 수단으로 헤더와 페이로드로 만들어진 메시지를 사용한다. XML 메시징의 기본 개념은 메시지의 헤더를 XML로 만들고, XML 메시지 브로커라는 메시징 서버들은 XML로 된 헤더 정보를 읽어서 메시지와 관련된 기능을 수행한다[2].

<그림 4>는 메시징 시스템의 간단한 구조를 보여준다. 클라이언트가 XML문서를 만들어 서비스를 요청하면, XML 메시징 서버는 이를 처리하고 응답 메시지를 만들어 클라이언트에 확인을 해 준다.



<그림 4> XML 메시징 시스템 구조

3.2 XML 메시징 시스템의 기능

메시지 브로커는 메시징 시스템에서 서버 기능을 수행하며, 받은 메시지에 대하여 헤더 처리, 암호화/복호화 및 보안, 에러/예외처리, 라우팅, 호출, 변환의 6가지 기능을 수행한다[2][8].

헤더처리

메시지를 수신했을 때 메시지 브로커에서 가장 먼저 수행되는 기능 가운데 하나이다. 헤더처리는 수신된 메시지의 헤더 영역을 확인하고, 거기에 포함된 기능 수행을 포함한다. 헤더처리에 특정번호를 헤더에 추가하여 추적 가능하게 하거나 헤더정보가 유효하게 구성됐는지를 검증할 수 있다. XML 메시지 내의 수신자가 적절한지를 처리 과정에서 사전에 검사할 수 있다.

보안

보안 관점에서 메시지 브로커는 보안의 가

장 기본적인 조건인 신원확인(authentication), 인증(authorization), 암호화(encryp tion), 부인 방지(nonrepudiation)를 보장해야 한다. 메시지가 수신되면 메시지 브로커는 우선 디렉토리 서비스나 데이터베이스에 저장된 자료로 신원을 확인한다. 해당 자격을 가진 사용자로 확인이 되면 메시지 브로커는 메시지에 포함된 기능이나 처리에 인증을 받는다.

오류와 예외처리

오류와 예외처리는 메시지 브로커가 수행하는 중요한 기능 중 하나다. 클라이언트가 수신한 메시지가 유효하지 않거나, 요청을 수행할 수 없는 경우에는 애러 메시지를 클라이언트에 보내야만 한다. 그리고 메시지 브로커 시스템의 문제로 인해 서비스를 제공할 수 없는 경우에도 해당 메시지를 클라이언트에 보낸다.

라우팅

메시지 라우팅은 두 단계로 이루어진다. 하나는 헤더 라우팅으로, 수신된 메시지가 어느 응용 프로그램에서 처리돼야 할지를 결정한다. 다른 하나는 페이로드 라우팅으로 해당 응용프로그램에서 어떤 프로세스나 메쏘드가 사용돼야 할 지를 결정한다.

호출

호출 단계에서는 실제 수신 메시지에 있는 페이로드의 자료를 가지고 메쏘드를 호출하게 된다. 여기서는 메쏘드 호출을 통해 브로커에서 클라이언트로 반환할 수행 결과가 만들어질 수도 있다.

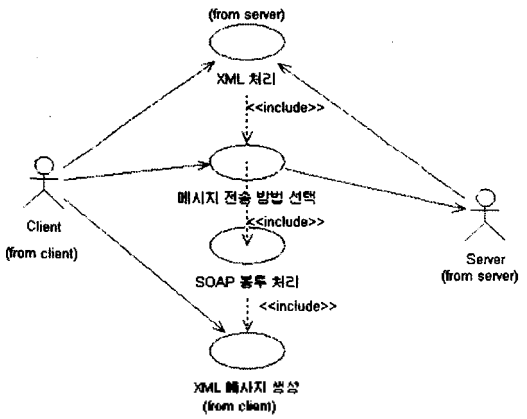
변환

변환에서는 다른 형태로의 변환이나 매핑을 수행한다. 여기서는 XSLT가 변환을 효과적으로 수행하기 위해 많이 사용된다.

4. 메시징 시스템 분석 및 설계

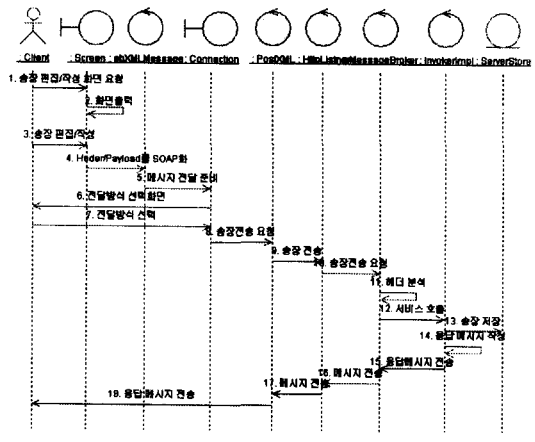
4.1 Use Case 다이어그램

<그림 5>는 유스 케이스 다이어그램을 나타냈다. 사용자는 XML 문서를 작성하고, 작성된 문서를 토대로 SOAP 봉투 처리를 한 다음, 메시지를 전송하는데 HTTP 방식으로 할 것인지, SMTP 방법으로 할 것인지를 선택한다. 이렇게 생성된 XML 문서를 메시징 서버에 전송해서 어떠한 처리를 요구한다. XML 문서 처리를 하기 위해서는 XML 문서가 있어야만 가능하기 때문에, XML 처리 유스 케이스는 생성 유스케이스를 반드시 필요로 하는 관계가 <<include>>로 표현된다.



<그림 5> 유스 케이스 다이어그램

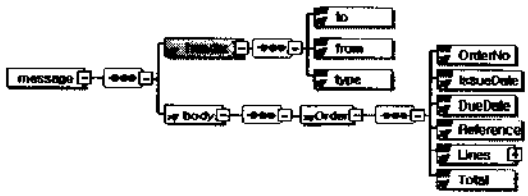
<그림 6>은 클라이언트가 XML 문서를 생성하고, 서버가 클라이언트의 요청을 처리하는 순차 다이어그램을 나타낸 것이다. 클라이언트는 사용자 인터페이스를 통하여 요청 서비스를 포함하는 XML 문서를 작성하고 XMLMessage를 통하여 SOAP 봉투를 작성한다. 이렇게 작성된 문서를 사용자가 HTTP나 SMTP를 통해 문서를 전송할 것인지를 선택한다. 그리고 서버에게 문서를 보내 관련된 처리를 요청한다. 서버에서는 HTTP 요청을 받아서, 이를 처리할 Mes sageBroker에 전달한다. Message Broker는 먼저 헤더를 분석해서 클라이언트가 요청한 서비스를 확인한다. 그 다음에 해당되는 서비스를 호출하는데, 현재는 송장의 저장만을 대상으로 하므로 InvokerImpl가 호출되고, 이 클래스는 저장소에 저장한 후, 처리가 성공했음을 알리는 XML 메시지를 만든다. 작성된 메시지는 역방향으로 사용자에게 성공/실패 메시지를 전달하게 된다.



<그림 6> 유스 케이스 - 순차 다이어그램

4.2 메시지의 구조

<그림 7>은 본 논문에서 사용될 XML 메시지의 구조를 나타낸다. 이 XML 문서는 헤더 부분과 바디 부분으로 나뉘어져 있다. 헤더 부분은 전달할 목적지와 발신자의 정보를 저장하고 있으며, 바디 부분에는 송장에 관한 여러 가지 정보를 입력하였다. XML 메시지에서 정보를 저장하는 부분이 바디 부분이므로 이곳에 주문 내역에 대한 정보를 입력하였다.

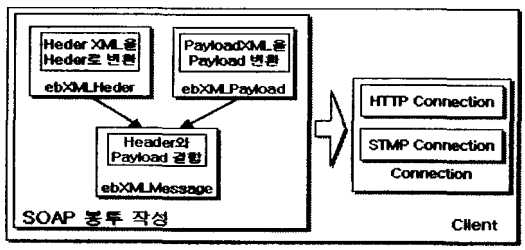


<그림 7> 메시지의 구조

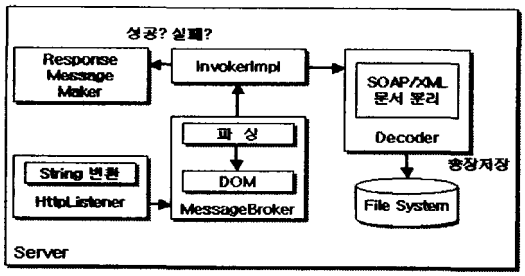
4.3 XML 메시징 서버

메시징 서버에서 처리과정은 <그림 8>, <그림 9>, <그림 10>과 같다. 클라이언트 측에서는 사용자가 작성한 XML 문서를 ebXMLHeder와 ebXMLPayload를 이용하여 헤더와 페이로드를 작성한다. 또한 이렇게 Heder와 Payload를 작성한 후 ebXML Message를 이용하여 Header와 Payload를 SOAP 봉투화 한다. 이렇게 작성된 메시지를 어느 방식으로 보낼 것인가를 결정하는 Connection을 이용하여 HTTP나 STMP로 보낼 것인지를 묻는 사용자 인터페이스가 나타나 둘중 하나를 선택하게 된다. 이렇게 클

라이언트가 메시지를 만들어서 XML 메시지 서버에 보내게 되면, 메시지 서버는 HTTP로 전송되어 온 메시지를 수신하는 HttpListener와 이 메시지로 어떤 작업을 처리해야 할지를 결정하는 MessageBroker, 그리고 헤더에 요청된 작업을 실제로 처리하는 Invoker로 이루어진다. 그 이외에 부수적인 Response Message Maker가 있다.



<그림 8> 클라이언트 측 메시지 처리 과정



<그림 9> 서버 측의 XML 메시지 처리과정

4.3.1 ebXMLMessage

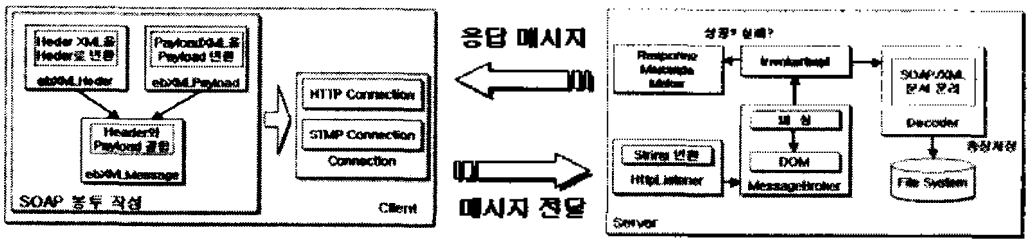
ebXMLHeader와 ebXMLPayload를 이용하여 작성한 것을 ebXMLMessage에서 구성하게 된다. 즉, 두 개의 서로 다른 Header 파일과 Payload 파일을 하나로 구성하는 기능을 가진다. 이곳에서 작성한 메시지는 connection에게 넘겨준다.

4.3.2 Connection

ebXMLMessage에서 받은 파일을 HTTP Send 버튼과 SMTP Send 버튼을 두어 각각의 HTTP와 SMTP 프로토콜로 보내도록 구성되어진다. ebXML의 전체 메시지를 구성하는 각각의 메시지의 특성들을 Content-Type과 Type, Bounder, Version 등을 통하여 설정할 수 있도록 구성한다.

4.3.3 HttpListener

통신 프로토콜을 한정해서, POST 메소드로 넘어온 XML 메시지를 문자열로 MessageBroker에 넘겨준다. HttpListner는 <그림 5>의 메시지 구조를 이용하여 작성한 XML 문서를 Stream 값으로 받은 다음, Reader로 변환한 후, 스트링으로 읽고, 그 코드를 MessageBroker에 넘겨준다.



<그림 10> 서버와 클라이언트 간의 메시지 전송

4.3.4 MessageBroker

HttpListner에서 얻어온 값을 이용하여, 파싱해서 DOM을 만들고, invoke()로 DOM을 넘겨서 XML 문서에서 <type> 태그에 있는 텍스트 값을 읽는다.

4.3.5 InvokerImpl

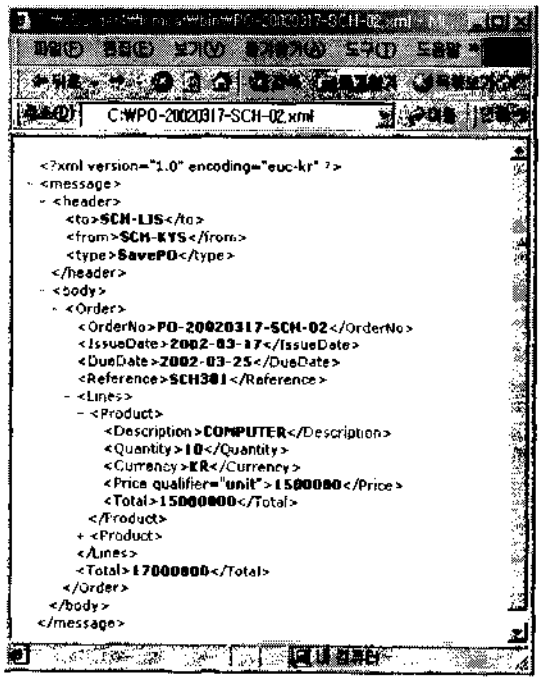
Invoker 인터페이스를 구현한 것인 InvokerImpl 클래스이다. 여기에서는 파일 시스템에 송장을 저장하고, Response Message Maker에서 응답 메시지를 만든다. Response MessageManager는 요청이 성공적으로 이루어졌을 때는 클라이언트에 보낼 '성공' 메시지나, 실패했을 경우의 '에러' 메시지를 XML 문서로 만든다.

5. XML 메시징 시스템 구현

XML 메시징 시스템의 환경으로 운영체제는 Windows 2000 Server, 웹 서버로는 Jakarta Tomcat V3.1.3을 사용하였으며, JDK1.3.1, XML Parser V2를 이용하여 구축하였다. 다음 <그림 11>은 XML 문서를 이용하여 작성된 송장을 나타낸다. 송장에서 헤더 부분과 페이로드 부분으로 나뉘는 것을 볼 수 있을 것이다. OrderNo를 이용하여 각 송장에 일련번호를 부여하였으며, 서버 측에서는 OrderNo와 동일한 파일이 생성된다.

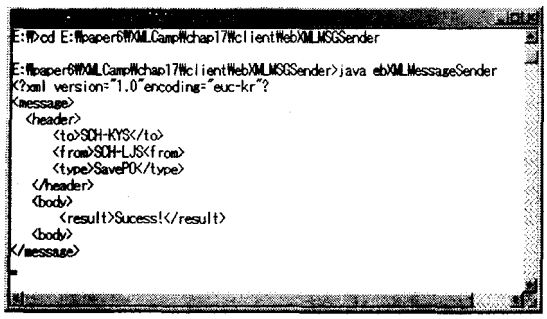
<그림 12>는 서버 측에서 메시지를 받을 때 나타나는 화면이다. 메시지 전송이 시작되면 Servlet이 시작되고, 전송이 완료되면 서블릿이 끝난다. 메시지를 잘 받았다는 것을 커맨드 상으로 확인하는 과정이다.

<그림 13>은 클라이언트 측에서 송장을 전송 시 송장의 헤더 부분이 결과물로 출력이 된다. 송장의 헤더 부분 출력되는 것은 송/수신자를 확인할 때 사용한다. 또한 'Success!'라는 메시지를 출력을 해주는데, 이 메시지는 송장의 전송이 완료되었을 경우에 보여지게 된다.



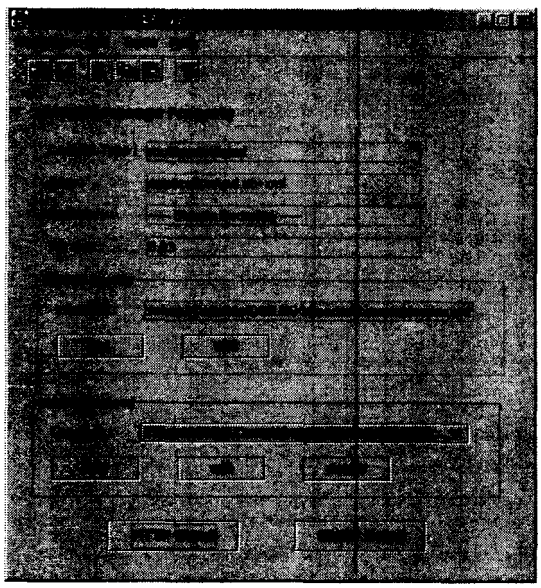
<그림 11> XML 메시징 서버에 저장된 송장

<그림 12> 서버 측 메시지 수신화면



<그림 13> 클라이언트 측 메시지 전송 성공화면

<그림 14>는 클라이언트 측의 GUI 인터페이스이다. 여기에서는 Heder와 Payload를 선택하고, 어떤 방식으로 보낼 것인가를 선택해 주는 화면이다.



<그림 14> 클라이언트 측 메시지 전송 GUI

6. 결론 및 향후 연구방향

공급사슬관리를 위해서는 다양한 분야의 기업간 협력이 필요하다. 그러나 실질적으로 기업간의 의사소통은 온라인상의 문서전달을 이용하여 이루어지고 있기 때문에, 기업간의 의사소통을 위해서는 온라인 메시지를 전달할 수 있는 기능을 포함한 시스템이 필요하다. 또한, 기업간 거래는 여러 가지 종류의 교류와 예측하기 힘든 양의 데이터 변화를 필요로 한다. 따라서 많은 플랫폼과 시스템은 서로 중립적인 데이터 교류에 필요한 표준을 필요로 하는데, 이 요구를 만족시켜 줄 수 있는 기술 중 하나가 바로 XML이며, 이는 W3C에 의해 표준으로 자리잡았다. 또한 XML은 표준화와 운영체제 중립적이라는 특성 때문에 기업간 거래에서 중요한 위치를 차지하며, 이미 많은 전자상거래 시스템에서 이용하고 있다.

본 논문에서는 현재 다양한 방향으로 연구가 진행중인 XML 문서 교환용 메시징 시스템을 설계 및 구현하였다. 이 메시징 시스템에서 전송되는 메시지의 형태는 ebXML에서 정의한 메시지 봉투를 적용하였다. 즉, 프로토콜 봉투, MIME 봉투, SOAP 봉투를 적용하여 메시지에 보안성을 첨가하였다. 이를 이용하면 기업간의 문서를 전달하는데 좀더 효율적인 방법을 제공한다. 향후 연구방향으로는 ebXML 메시징 시스템을 좀더 발전시켜 HUB 시스템을 구현하고자 한다. XML HUB 시스템은 XML 메시징 전달, 메시지 변환, 저장, 비동기전송 등을 지원하며, 기업간의 메시지 전송에서 가장 핵심이 되는 부분으로 발전할 것이다.

참고문헌

[1] Alexander Nakhimovsky, Tom Myers, Professonal Java XML Programming, WORX, 2000
[2] Subrahmanyam Alluamaraju, Professional Java E-Commerce, WORX, 2001
[3] Elliotte Rusty Harold , W. Scott Means, XML XML in a Nutshell - A Desktop Quick Reference, O'reilly, 2001
[4] RozettaNet, http://www.rosettanet.org/
[5] MicroSoft, "Biztalk Framework 1.0 Independent Document Specification"' http://www.biztalk.org
[6] 이태웅, "전자상거래 표준화 기술로서의 XML", 추계국제학술대회 논문집, Vol.0, No.0, 1999
[7] 신동규, "XML/EDI 시스템의 설계 및 구현", 한국정보처리학회 논문지 8-D권 제 2호, 2001
[8] 김채미, 전문가와 함께가는 XML Camp, 아이트Press, 2001
[9] 조은정, 초보자를 위한 XML, 가남사, 2001
[10] 조완수, UML 객체지향 분석·설계, 홍릉과학출판사, 2000
[11] 지영수, 객체지향 Rational Rose 2000, 홍릉과학출판사, 2001
[12] 김채미, 최학열, 글로벌 e비즈니스 리더를 위한 ebXML, 대청미디어, 2001
[13] 케이큰닷컴, "e-Procurement", 2001 see http://www.chemizeninfo.com/chem mart/sw/eprocurement.asp

저자 소개

임종선 (E-mail : ronmer74@hotmail.com)
1997 년 청운대학교 전산학과 졸업(학사)
1999 년 순천향대학교 일반대학원 졸업(석사)
2002 ~ 현재 순천향대학교 일반대학원 전산학과 재학중(박사과정)
관심분야 : XML, XML Schema, UML.

주경수 (E-mail : gsoojoo@asan.sch.ac.kr)
1980년 고려대학교 이과대학 수학과 졸업(학사)
1985년 고려대학교 일반대학원 전산학과 졸업(석사)
1993년 고려대학교 일반대학원 전산학과 졸업(박사)
1986년 ~현재 순천향대학교 정보기술공학부 교수
관심분야 : Database Systems, System Integration, Object-oriented Systems.