

TCP/IP 기반의 XML 메시징 시스템 설계 및 구현

임종선⁺ · 주경수^{**}

요 약

XML은 표준화와 운영체제 중립적이라는 특성 때문에 전자상거래에서 중요한 위치를 차지하며, 이미 많은 전자상거래 시스템에서 이용하고 있다. B2B 전자상거래분야를 위해서는 기업간 비즈니스에 대한 업무 규칙과 절차가 표준화되어야 한다. 그러나 기업들이 표준화 된 XML 문서를 사용하는 것이 아니라 기업의 실정에 맞게 시스템을 구축하여 기업간 XML 문서 전달을 하고 있어, 이에 따라 많은 문제점이 나타나고 있다. 이러한 요구를 기반으로 많은 조직과 기업에서는 XML을 기반으로 하는 전자상거래 표준화 작업을 프레임 워크 형태로 진행하고 있다. 본 논문에서는 TCP/IP 기반의 메시징 시스템을 설계 및 구현하였으며, 시스템을 모듈별로 설계를 하였기 때문에 시스템의 확장성이 용이하다. 본 메시징 시스템을 전자상거래 시스템에 적용하여 효율적으로 이용할 수 있을 것이다.

Design and Implementation XML Messaging System Based on TCP/IP

Lim Jong-Seon⁺ and 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 are its own XML documents instead of standard documents. Therefore many organizations try to make standards for e-commerce based on framework. In this paper, we designed and implemented XML Messaging System based on TCP/IP protocols. This system are designed by module. Because, it is easy to extensibility of system. So we can more easily and efficiently build e-commerce system based on this XML messaging system.

Key words: XML Messaging System, XML, XMS

1. 서 론

현재 많은 전자상거래 시스템에서 XML을 이용하고 있으며, 이 전자상거래에서 사용되는 문서는 XML을 이용하여 작성하고 보내어 진다. 그러나 이러한 기업들이 표준화 된 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 ebXML(e-business XML)에 따라서

만든 XML 문서를 사용하는 기업간의 거래에 XML 문서를 사용한다면 양자 사이에 XML 문서를 변환하는 작업이 필요하다. 이러한 B2B(Business to Business)에서 양자간에 직접적인 거래를 위해서 변환과 문서 전달 기능을 수행해 주는 메시징 시스템이 있어야 한다. 본 논문에서는 이러한 메시징 시스템을 설계 및 구현하였다.

본 논문에서 설계한 메시징 시스템은 TCP/IP 기반의 기본적인 구조를 가지고 있으며, 메시지의 전송, 저장, 백업, 복구등의 기능을 제공한다. 이러한 시스템을 이용하면, 기업간 문서 전달에 있어서 좀더 효율적으로 메시지를 저장, 검색, 관리할 수 있을 것이다. 본 논문의 2장에서는 관련 연구 및 기술을 소개하고, 3장

접수일 : 2002년 9월 30일, 완료일 : 2002년 12월 30일

⁺ 준회원, 순천향대학교 전산학과

^{**} 순천향대학교 전산학과

에서는 XML 메시징 시스템 분석 및 설계에 대하여 설명하며, 4장에서는 메시징 시스템을 구현한 결과를 보여주며, 5장에서는 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[4]. 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 적용이 되고 있다[5].

RosettaNet은 정보기술 및 전자부품의 SCM(Supply Chain Management)을 위한 XML 기반의 비즈니스 표준을 개발하기 위해 1998년에 결성된 컨소시엄으로, 350여개 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. RosettaNet에서 제공하는 표준 중에서 가장 중요한 것은 PIP이며, PIP는 거래 파트너와 인터페이스 할 수 있는 비즈니스 프로세스를 정의하고 있다. PIP는 크게 6개의 클러스터로 구성되어 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 로드 맵 역할을 한다. 각 PIP는 모든 비즈니스 로직, 메시지 흐름, 메시

지 내용을 포함한다[2].

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 있다. BizTalk을 바탕으로 하는 B2B 전자상거래 시스템에서는 거래자들이 비즈니스 문서들을 교환하기 위하여 BizTalk server를 이용한다. 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[3].

2.2 XML 메시징 시스템

메시징 시스템은 서로 다른 시스템간에 특정 기능을 수행하기 위한 수단으로 헤더(Header)와 페이로드(Payload)로 만들어진 메시지를 사용한다. XML 메시징의 기본 개념은 메시지의 헤더를 XML로 만들고, XML 메시지 브로커(Message Broker)라는 메시징 서버들은 XML로 된 헤더 정보를 읽어서 메시지와 관련된 기능을 수행한다[6].

클라이언트가 XML 문서를 만들어 서비스를 요청하면, XML 메시징 서버는 이를 처리하고 응답 메시지를 만들어 클라이언트에 확인을 해 준다. XML 메시징 서비스는 메일과 같이 헤더 부분과 페이로드 부분으로 구성된다. 헤더 부분은 메시지를 어떻게 처리해야 하는지에 관한 정보가 들어있으며, 페이로드 부분에는 메시지의 내용이 들어가 있다.

2.3 XML 메시징 시스템의 기능

메시지 브로커는 메시징 시스템에서 서버 기능을 수행하며, 받은 메시지에 대하여 헤더처리, 암호화/복호화 및 보안, 에러/예외처리, 라우팅, 호출, 변환의 6가지 기능을 수행한다[1,6].

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 가장 기본적인 조건인 신원확인(authentication), 인증(authorization), 암호화(encryption), 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 여러 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 메소드가 사용돼야 할 지를 결정한다.

호출

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

변환

변환에서는 다른 형태로의 변환이나 매핑을 수행한다. 여기서는 XSLT(Extensible Stylesheet Language Transformation)가 변환을 효과적으로 수행하기 위해 많이 사용된다.

3. 시스템 분석 및 설계

3.1 시스템 분석

본 시스템을 구성하는데는 크게 서버측과 클라이언트 측으로 구별하였다. 그 이유는 서버에서는 메시지의 관리와 처리를 전담으로 하고, 클라이언트 측에서는 사용자 인터페이스를 통하여 서버 시스템과의 접속 후, 작업을 처리하는 형태를 취하기 위해서이다. 그림 1과 그림 2는 메시징 시스템의 유스 케이스 다이어그램과 순차 다이어그램을 나타낸 것이다. 유스 케이스 다이어그램은 시스템의 전반적인 처리 과정이 어떻게 이루어지는가를 나타내며, 순차 다이어그램은 시간의 흐름에 따라서 프로세스의 처리 과정이 어떻게 이루어지는가를 나타낸다[7].

그림 1을 보면, 클라이언트 측에서는 단지 서버측에 어떤 처리를 요청하고, 처리된 결과를 얻어오는 형식으로 구성이 된다. 시스템을 그림 1과 같이 구성하게 된 이유는 메시징 처리 과정에서의 서버측의 어떤 정보도 필요로 하지 않기 때문이다. 또한 보안성도 있어야 하므로, 클라이언트 측면에서는 서버의 어떤 부분도 접근하기 어렵게 설계를 하였다. 단지 요청만 할 수 있도록 하였다.

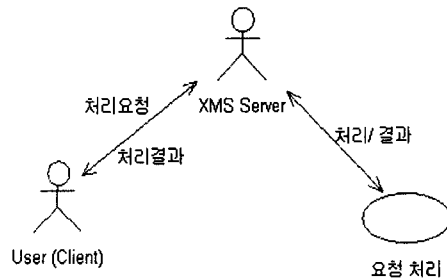


그림 1. XMS 시스템의 유스케이스 다이어그램

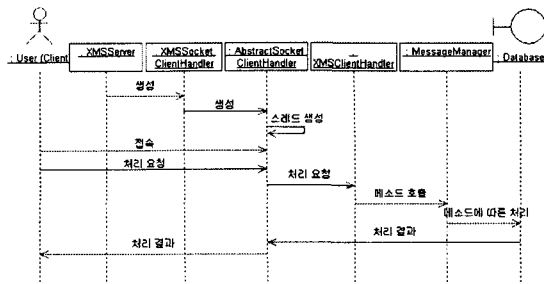


그림 2. 서버측 순차 다이어그램

그림 2의 순차 다이어그램은 시간의 흐름에 따라서 서버측의 각 클래스간에 어떠한 작업을 수행하는지를 보여주는 것이다. 이 순차 다이어그램에서 보면, 클라이언트는 단지 스레드를 통해서만 어떤 작업의 처리를 요청할 수 있으며, 처리 결과를 얻어갈 수 있다.

그림 3은 XML 문서의 구조를 나타내었다. 이 시스템에서는 간단한 메시징 시스템을 구현하였기 때문에 루트 엘리먼트인 Message에 각각 recID, userID, msgFrom, msgTO 등의 자식 엘리먼트가 있다. 메시지를 전송하기 위하여 필요한 엘리먼트들은 msgTo, msgFrom, title, content, sendTime의 5개의 엘리먼트가 필요하며, 나머지 엘리먼트들은 메시지를 데이터베이스에 저장하거나, 읽었는지 읽지 않았는지의 확인을 위하여 쓰인다.

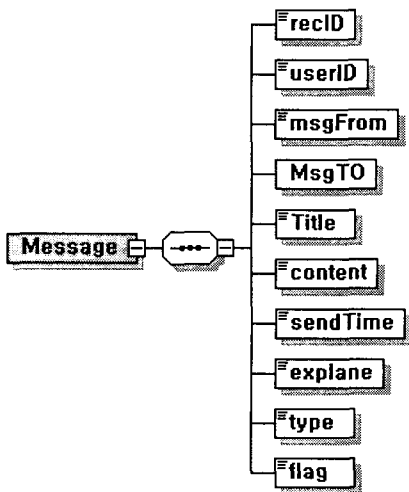


그림 3. XML 문서의 구조

3.2 시스템 설계

본 시스템은 현재 메시징 시스템에서 일반적으로 쓰이는 HTTP/SMTP 형태를 취하지 않고, TCP/IP 레벨의 소켓을 이용하였다. 소켓은 클라이언트가 접속하여 커넥션을 유지하면, 다시 접속할 필요가 없다. 또한 현재 실질적으로 많이 사용하는 방법중의 하나이다. 다음의 그림 4, 그림 5는 본 시스템의 클라이언트와 서버의 구조를 나타낸 것이며, 각 클래스간의 생성, 호출 관계를 나타내고 있다.

클라이언트 측면에서의 기능은 단순한 메시지의 저장, 삭제, 검색하면 관리와 사용자 인증 및 소켓을 연결하는 능만을 가진다. AbstractSocketConnection 클래스

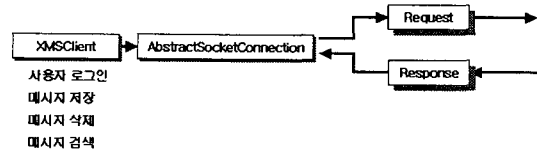


그림 4. 클라이언트측 각 클래스간의 구조

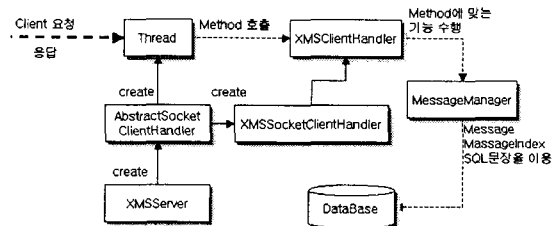


그림 5. 서버측 각 클래스간의 구조

스에서는 Request와 Response 클래스를 생성하여 사용자가 요구한 기능을 수행하기 위하여 서버측과 통신을 위한 소켓을 생성하는 클래스이다.

서버의 측면에서는 소켓 생성요청을 통하여 AbstractSocketClientHandler를 통하여 소켓 처리를 위한 스레드를 생성하고, 스레드를 처리해 주기 위한 XMLClientHandler를 생성시키기 위하여 XMS SocketClientHandler를 생성시켜 준다. 이후, 클라이언트에서 어떤 요청을 발생시키면 MessageManager를 통하여 필요한 기능에 해당되는 SQL을 생성시켜 데이터베이스에 질의를 한다. 이러한 작업을 관리하는 XMSServer 클래스는 메시징 시스템의 서버측에서 전체를 관리하는 클래스이며, 소켓의 생성과 클라이언트의 요청에 따른 메소드를 선택할 수 있는 각 클래스를 생성시키는 슈퍼 클래스이다.

다음의 표 1, 표 2, 표 3, 표 4는 각 클래스의 기능을 나타낸 것이다. 표 1, 표 2는 위에서 설명한 서버측의 클래스에 대한 세분화된 기능을 나타낸다. 표 3, 표 4는 클라이언트 측면의 클래스에 대한 기능 명세이다. 각 표에서는 XMS 시스템에서 사용되는 각각의 클래스 기능에 대한 상세한 기술이 되어있으며, 각 클래스가 해야 할 일에 대하여 명시를 하였다.

4. 메시징 시스템 구현

본 시스템의 구현 환경으로는 Windows 2000 server, JDK 1.3 이상, MS-SQL server 2000, JBuilder 5.0을 이용하여 구현하였다. 그림 6은 사용자 관리 화

표 1. XMS server측 클래스의 기능

클래스명	기능
XMSServer	로그인을 위해 데이터베이스 커넥션, AbstractSocketClientHandler 클래스를 생성.
AbstractSocketClientHandler	스레드를 생성, 클라이언트가 요구하는 요청을 받아서 처리하여 XMS에서 매칭되는 메시지를 호출.
XMSSocketClientHandler	XMSClientHandler를 생성.
XMS	XMSClientHandler 클래스의 인터페이스.
XMSClientHandler	XMS에서 호출된 메시지를 수행, 주로 MessageManager 인터페이스를 호출.
MessageManager	메시지를 관리하는 인터페이스 클래스
MessageManagerImpl	MessageManager에서 호출된 메시지 수행, 주로 데이터베이스 커넥션을 통해 데이터베이스 처리 작업을 수행.
Message	메시지를 정의한 클래스, 메시지의 각 항목에 대하여 get/set 메소드를 포함.
MessageIndex	메시지 인덱스를 정의한 클래스, 메시지 인덱스 각 항목에 대하여 get/set 메소드를 포함.

표 2. Admin UI 클래스의 기능

클래스명	기능
AdminUI	관리자 UI 클래스, 데이터베이스 커넥션을 설정, UserSetup 클래스를 호출
UserSetup	세계의 탭으로 구성 <ul style="list-style-type: none"> • 사용자 관리 <ul style="list-style-type: none"> - 사용자 ID를 수정, 삭제, 삽입해주는 기능을 가짐. 삽입시 AddUser 호출 - 사용자 리스트를 보기 위하여 UserList 호출 • 백업 <ul style="list-style-type: none"> - 지정된 자료를 백업, 사용자ID, messageTo에 의해서 원하는 자료 선택. - 자료 백업을 위하여 BackupFileFrame 호출 • 리스토어 <ul style="list-style-type: none"> - 지정된 파일에 저장되어 있는 자료를 데이터베이스로 복구 - 리스토어를 위하여 Restore FileFrame을 호출
AddUser	사용자의 ID, PW, 이름을 데이터베이스에 저장.
UserList	사용자의 리스트를 리스트박스로 출력
BackupFileFrame	선택된 레코드를 파일로 백업 - 문서 형태는 XML, 확장자는 제한없음.
RestoreFileFrame	선택된 파일을 데이터베이스의 레코드로 백업.

표 3. Client 클래스의 기능

클래스명	기능
XMSClient	Login Form 호출
XMSConnection	서버에 요청할 요구 사항들을 정의 AbstractSocketConnection을 상속받음
AbstractSocketConnection	소켓 연결 Request, Response 클래스 호출
Request	명령을 서버에 전송
Response	결과값을 전송 받음

표 4. ClientUI 클래스의 기능

클래스명	기능
LoginForm	ID와 P/W값을 입력 받아 옳은 값이면XMLMessage 호출
XMLMessage	자료 저장, 삭제, 검색 검색 시 XMLResult 호출
XMLResult	키 값(redID)을 입력 받아 자료를 검색하여출력

면을 나타낸 것이다. 여기에서 사용자 등록, 삭제, 수정 등을 할 수 있으며, 사용자의 ID는 변경 불가능하게 하였다. 또한 그림 7은 현재 시스템에 저장되어 있는 메시지를 파일로 백업 받게 하였다. 파일의 형태는

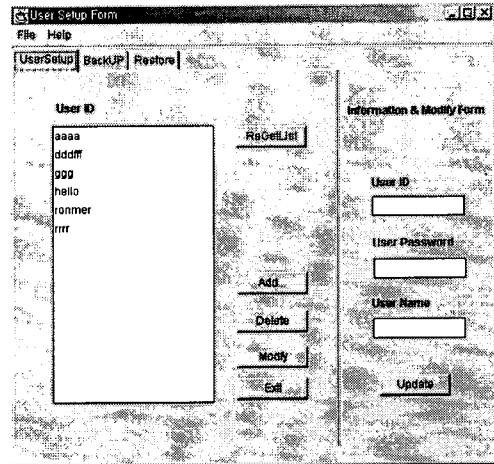


그림 6. 사용자 관리 화면

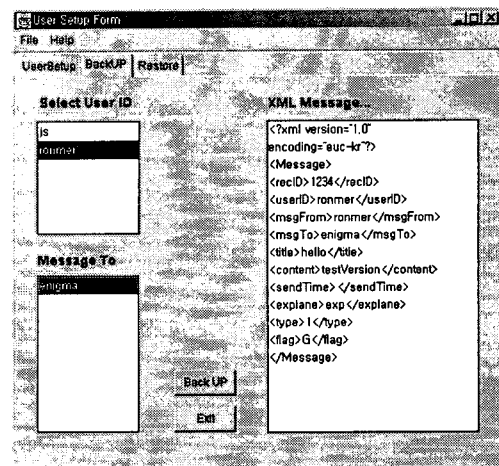


그림 7. 데이터 백업 화면

XML의 구조를 가지게 하였으며, 마지막에 주석을 붙여 언제 백업했는지를 나타나게 하였다.

또한, XML 문서를 파싱 및 생성하는 과정은 JDOM을 이용하였다. 이 시스템에서는 XML 문서의 구조가 간단하기 때문에 특별한 알고리즘을 사용하지 않았다. 단지 XML 문서의 내용 부분만을 인코딩 함으로써, 정보의 보안성을 첨가하였다.

그림 8은 백업한 데이터를 복구하는 과정이다. 백업한 파일을 선택하여 복구할 수 있도록 하였다. 선택된 파일은 파싱하여 데이터베이스 테이블에 저장된다. 다음의 그림 9는 클라이언트의 사용자 인터페이스를 나타낸 것이다. 사용자가 메시지를 검색, 삭제, 전송을 할 수 있으며, 사용자가 메시지 전송을 요구할 경우

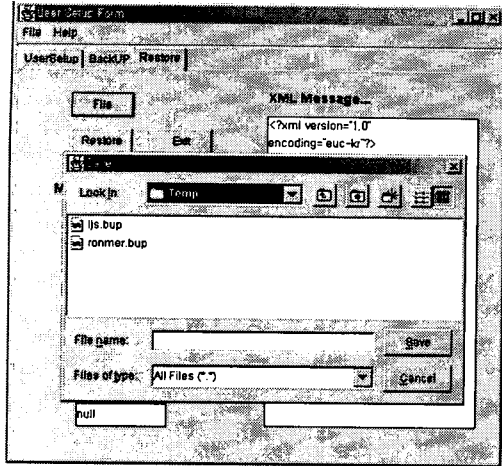


그림 8. 데이터 복구 화면

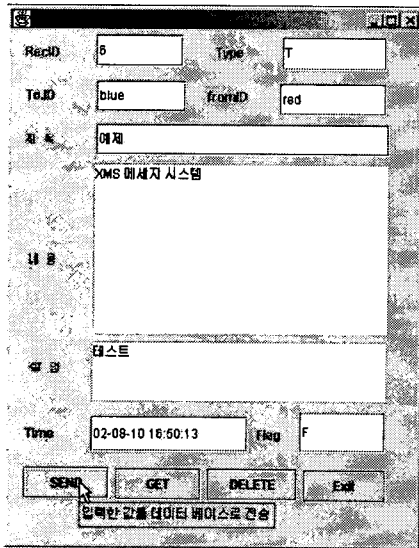


그림 9. 메시지 전송 화면

기본적으로 Base64 Encoding을 하여 전송을 해 주고, 메시지를 수신하여 읽을 경우에는 Base64 Encoding을 한 메시지 내용을 다시 디코딩을 하여 보여주게 된다. 이는 기본적인 암호화를 해주기 위함이다.

5. 결 론

기업간의 거래에서 문서를 서로 주고받을 때, 기업의 실정에 맞게 문서를 변형해야 한다. 이 요구를 만족시켜 줄 수 있는 기술 중 하나가 바로 XML이며, 이는

W3C에 의해 표준으로 자리잡았다. 또한 XML은 표준화와 운영체제 중립적이라는 특성 때문에 전자상거래에서 중요한 위치를 차지하며, 이미 많은 전자상거래 시스템에서 이용하고 있다. 이 전자상거래에서 사용되는 문서는 XML을 이용하여 작성하고 보내어 진다. 그러나 이러한 기업들이 표준화 된 XML 문서를 사용하는 것이 아니라 기업의 실정에 맞게 문서를 구축하여 기업간의 XML 문서 전달에 많은 문제점이 나타나고 있다. 이러한 흐름을 반영하여, 기업간의 문서 전달 시스템에서, XML 문서를 이용하면 효과적이다.

본 논문에서 구현된 XML 메시징 시스템은 기업간 문서전달의 방식을 현재 다양한 솔루션으로 개발되고 있는 TCP/IP 방식에서 동작하도록 구현하였다. 이러한 방식은 현재 활발히 사용중이거나 시스템을 구축하는 대표적인 시스템이다. 이 시스템은 메일과는 달리 클라이언트가 직접 서버에 접속하여 메시지를 전달하기 때문에 시스템의 보완성을 향상시킬 수 있다. 또한 소켓을 이용하여 서버 시스템에 지속적으로 접속이 되어있으므로 재 접속에 사용되는 시간을 줄일 수 있다. 이 시스템은 모듈별로 설계를 하였기 때문에 차후 확장에 용이하며, 다른 시스템과의 연계를 위한 보조 시스템으로 사용할 수도 있다. 본 시스템을 이용하여 약간의 수정 및 보완을 거치게 되면 기존의 B2B 거래에서 유용하게 쓰일 전망이다.

참 고 문 헌

- [1] Subrahmanyam Alluamaraju, Professional Java E-Commerce, WORX, 2001.
- [2] RozettaNet, <http://www.rosettanet.org/>
- [3] MicroSoft, "Biztalk Framework 1.0 Independent Document Specification" <http://www.biztalk.org>
- [4] 이태용, "전자상거래 표준화 기술로서의 XML", 추계국제학술대회 논문집, Vol.0, No.0, 1999.
- [5] 신동규, "XML/EDI 시스템의 설계 및 구현", 한국정보처리학회 논문지 8-D권 제 2호, 2001.
- [6] 김채미, 전문가와 함께가는 XML Camp, 아이트 Press, 2001.
- [7] 조완수, UML 객체지향 분석·설계, 홍릉과학출판사, 2000.



임 종 선

1997년 청운대학교 전산학과 졸업(학사)
1999년 순천향대학교 일반대학원 졸업(석사)
2002~현재 순천향대학교 일반대학원 전산학과 재학중(박사과정)

관심분야 : Database System, B2B, XML DBMS 등이다.
E-mail : ronmer@chol.com



주 경 수

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 System, XML, System Integration, Object Oriented System 등이다.

E-mail : gsoojoo@sch.ac.kr

교신저자

임 종 선 336-745 충남 아산시 신창면 읍내리 646
순천향대학교 멀티미디어관 M601