

전자상거래를 위한 Client/Server메시징 시스템 설계 및 구현

고 영승*, 임 종선**, 주 경수***
순천향대학교 전산학과

Design and Implementation Client/Server Messaging System for electronic commerce

Ko Young Seung, Lim Jong-Seon, Joo Kyung-Soo
Dept. of Computer Science, College of Engineering Soonchunhyang University
E-mail: bluekaoru@orgio.net, ronmer@chol.com, gsoojoo@sch.ac.kr

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.

는 결론을 기술한다.

1, 서론

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 기업 간의 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 기업 간의 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, 복구 등의 기능을 제공한다. 이러한 시스템을 이용하면, 기업 간 문서 전달에 있어서 좀더 효율적으로 메시지를 저장, 검색, 관리할 수 있을 것이다. 본 논문의 2장에서는 관련 연구 및 기술을 소개하고, 3장에서는 XML 메시징 시스템 분석 및 설계에 대하여 설명하며, 4장에서는 메시징 시스템을 구현한 결과를 보여주며, 5장에서

2. 관련 연구

현재까지 XML을 이용한 메시지 교환 시스템은 기존

의 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 적용이 되고 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 기술규

격을 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.

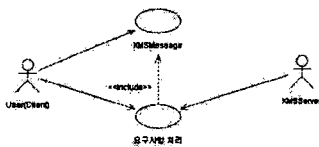
RosettaNet Implementation Framework (RNIF)은 신속하고 효율적으로 PIP를 구현하기 위한 교환 프로토콜을 제공해 줍니다. PIP는 특화된 XML 기반의 시스템과 시스템간의 메시지 교환 절차입니다. 각 PIP 규정은 어휘를 포함하고 있는 비즈니스 문서와 메시지 교환절차를 구성해주는 비즈니스 프로세스로 구성되어 있습니다.[2].

Microsoft사는 1999년 'BizTalk Initiative'라는 이름을 가지고 XML 기반의 B2B 전자상거래 솔루션을 발표하였다. BizTalk은 XML을 이용하여 기업 내부 또는 기업 간 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 기업 간 비즈니스 프로세스를 자동화시키는 플랫폼을 제공한다[3].

3. 시스템 분석 및 설계

3.1 시스템 분석

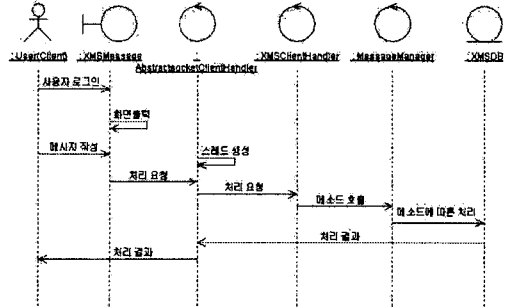
본 시스템을 구성하는데는 크게 서버 측과 클라이언트 측으로 구별하였다. 그 이유는 서버에서는 메시지의 관리와 처리를 전담으로 하고, 클라이언트 측에서는 사용자 인터페이스를 통하여 서버 시스템과의 접속 후, 작업을 처리하는 형태를 취하기 위해서이다. <그림 1>과 <그림 2>는 메시징 시스템의 Use Case Diagram과 Sequence Diagram을 나타낸 것이다. Use Case Diagram은 시스템의 전반적인 처리 과정이 어떻게 이루어지는가를 나타내며, Sequence Diagram은 시간의 흐름에 따라서 프로세스의 처리 과정이 어떻게 이루어지는가를 나타낸다[3].



<그림 1> XMS 시스템의 Use Case Diagram

시스템은 <그림 1>에서 보여주는 것처럼, 클라이언트 측에서는 단지 서버 측에 어떤 처리를 요청하고, 처리된 결과를 얻어오는 형식으로 구성이 된다. 시스템을 이런

방식으로 구성하게 된 이유는 클라이언트 측은 메시징 처리 과정에서의 서버 측의 어떤 정보도 필요로 하지 않기 때문이다. 또한 보안성도 있어야 하므로, 클라이언트 측면에서는 서버의 어떤 부분도 접근하기 어렵게 설계를 하였다.

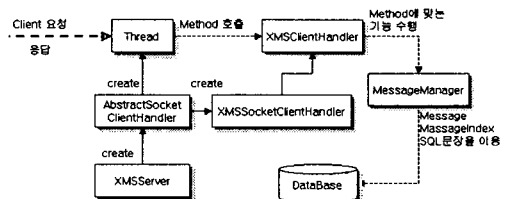


<그림 2> XMS서버의 Sequence Diagram

<그림 2>의 Sequence Diagram은 시간의 흐름에 따라서 서버 측의 각 클래스간에 어떠한 작업을 수행하는지를 보여주는 것이다. 이 Sequence Diagram에서 보면, 클라이언트는 단지 스레드를 통해서만 어떤 작업의 처리를 요청할 수 있으며, 처리 결과를 얻어갈 수 있다.

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 없다는 장점을 가지고 있으며, SMTP처럼 최종 수신자에게 도달하기 전에 다른 서버를 경유하지 않으므로 보안성이 좋아서 현재 실질적으로 많이 사용하는 방법중의 하나이다. 다음의 <그림 3>은 본 시스템의 구조를 나타낸 것이며, 각 클래스간의 생성, 호출 관계를 나타내고 있다. XMSServer 클래스는 메시징 시스템의 서버 측에서 전체를 관리하는 클래스로서, 소켓의 생성과 클라이언트의 요청에 따른 메소드를 선택할 수 있는 각 클래스를 생성시키는 슈퍼 클래스이다.



<그림 3> 서버측 각 클래스간의 구조

다음의 <표 1>, <표 2>, <표 3>는 각 클래스의 기능을 나타낸 것이다. <표 1>은 위에서 설명한 서버 측의 클래스에 대한 세분화된 기능을 나타내며, <표 2>는 클라

이언트 측의 클래스에 대한 세분화 된 기능을 나타낸 것이고, <표 3>은 서버의 관리를 위한 사용자 인터페이스 클래스에 대한 기능 명세이다.

<표 1> XMS server측 클래스의 기능명세

클래스명	기능
XMServer	로그인을 위해 데이터베이스 커넥션, AbstractSocketClientHandler 클래스를 생성.
AbstractSocketClientHandler	스레드를 생성, 클라이언트가 요구하는 요청을 받아서 처리하여 XMS에서 대입되는 메시지를 호출.
XMSocketClientHandler	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	메시지의 인덱스 각 항목에 대하여 get/set 메소드를 포함.

<표 2> XMS Client측 클래스의 기능명세

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
LoginForm	ID와 P/W값을 입력 받아 옳은 값이면 XMSMessage 호출
XMSMessage	자료 저장, 삭제, 검색, 검색 시 XMSResult 호출
XMSResult	키 값(keyID)을 입력 받아 자료를 검색하여 출력

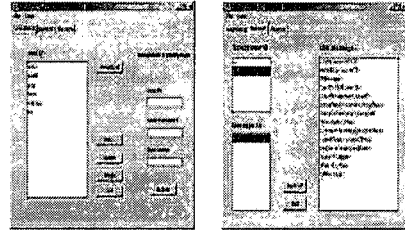
<표 3> UserUI측 클래스의 기능명세

클래스명	기능
XMServer	로그인을 위해 데이터베이스 커넥션, AbstractSocketClientHandler 클래스를 생성.
AbstractSocketClientHandler	스레드를 생성, 클라이언트가 요구하는 요청을 받아서 처리하여 XMS에서 대입되는 메시지를 호출.
XMSocketClientHandler	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	메시지의 인덱스 각 항목에 대하여 get/set 메소드를 포함.

4. 메시징 시스템 구현

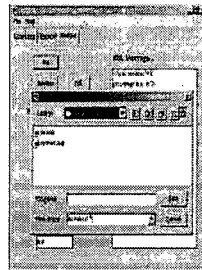
본 시스템의 구현 환경은 Windows 2000 server와 MS-SQL server 2000이며, JBuilder 5.0을 이용하여 구현하였다. <그림 4>은 사용자 관리 화면을 나타낸 것이다. 여기에서 사용자 등록, 삭제, 수정 등을 할 수 있으며, 사

용자의 ID는 변경 불가능하게 하였다. 또한 <그림 5>은 현재 시스템에 저장되어 있는 메시지를 파일로 백업 받게 하였다. 파일의 형태는 XML의 구조를 가지게 하였으며, 마지막에 주석을 붙여 언제 백업했는지를 나타내게 하였다



<그림 4> 사용자 관리화면 <그림 5> 데이터 백업화면

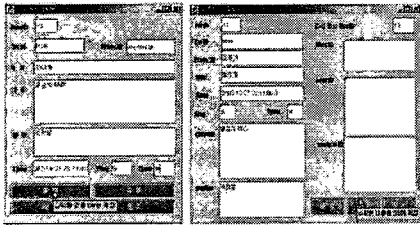
<그림 6>은 백업한 데이터를 복구하는 과정이다. 백업한 파일을 선택하여 복구할 수 있도록 하였다. 선택된 파일은 파싱되어 데이터베이스 테이블에 저장된다.



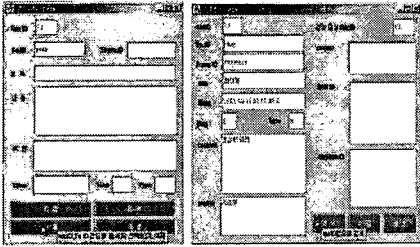
<그림 6> 데이터 복구화면

다음의 <그림 7>, <그림 8>, <그림 9>, <그림 10>은 클라이언트의 사용자 인터페이스를 나타낸 것으로서, <그림 7>은 사용자가 메시지를 입력하여 전송하는 화면이고, <그림 8>은 이미 작성된 내용을 수정하여 다시 입력하는 화면이며, <그림 9>는 RecID와 ToID값을 입력받아 사용자가 원하는 값을 삭제하는 화면이고, <그림 10>은 RecID값을 입력받아 사용자가 원하는 메시지에 대한 정보를 검색하는 화면이다.

여기서 전송되는 메시지 중에서 내용에 해당하는 부분은 Base64 인코딩을 하여 전송을 해 주고, 메시지를 받을 때 역시, 인코딩 된 내용을 Base64 디코딩을 해서 보게 되는데 이는 서버관리자가 사용자의 개인적인 메시지 내용에 대해서 볼 수 없도록 메시지의 보안적인 측면을 강화하기 위한 것이다.



<그림7> 메시지 전송화면 <그림8> 메시지 수정화면



<그림9> 메시지 삭제화면 <그림10> 메시지 검색화면

- [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] 김채미, 전문가와 함께가는 XML Camp, 아이트Press, 2001
- [5] 조완수, UML 객체지향 분석·설계, 홍릉과학출판사 2000
- [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

5. 결론

B2B에서 기업 간의 거래에서 문서를 서로 주고받을 때, 각 기업은 상대방의 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 기업 간의 XML 문서 전달에 많은 문제점이 나타나고 있다. 이러한 문제점을 해결하기 위해서는 B2B(Business to Business)에서 양자간에 직접적인 거래를 위해서 변환과 문서 전달 기능을 수행해 주는 메시징 시스템이 있어야 한다.

본 논문에서 구현된 XML 메시징 시스템은 기업 간 문서전달의 방식을 기존의 HTTP/SMTP방식이 아닌 TCP/IP 방식에서 동작하도록 구현하였다. 이러한 방식은 현재 활발히 사용중이거나 시스템을 구축하는 대표적인 시스템으로서, 클라이언트가 접속하여 커넥션을 유지하면, 다시 접속할 필요가 없다는 장점과 메일(SMTP)과는 달리 클라이언트가 다른 서버들을 경유하지 않고 직접 서버에 접속하여 메시지를 전달하기 때문에 시스템의 보안성을 향상시킬 수 있다는 장점을 가지고 있다. 또한 시스템을 모듈별로 설계를 하였기 때문에 차후 확장에 용이하다.

참고문헌