

SOAP을 기반으로 한 XML Hub 시스템 개발

김 용 수[†] · 주 경 수^{††}

요 약

SOAP(Simple Object Access Protocol)은 분산 환경에서의 정보 교환에 사용되는 분산 컴퓨팅 프로토콜로 분산 시스템간에 메시지를 전달하는 방법, 원격 프로시저 호출/응답을 처리하는 방법이 정의하고 있다. 이러한 SOAP은 텍스트 기반의 XML을 이용한 프로토콜이기 때문에 운영체제 및 프로그래밍 언어에 독립적으로 사용할 수 있다. 따라서 전자상거래 표준인 ebXML에서도 메시지 전송을 위해 SOAP을 사용하고 있다. 본 논문에서 SOAP을 기반으로 XML Hub 시스템을 개발하였다. 이에 따라 본 XML Hub 시스템을 기반으로 B2B 메시지 전송이 용이하게 되어, 보다 많은 비즈니스 파트너를 효과적으로 통합관리 할 수 있을 것이다.

Developing XML Hub System based on SOAP

Yong-Soo Kim[†] · Kyung-Soo Joo^{††}

ABSTRACT

SOAP is a protocol used for information exchange in distributed computing environment and defines mechanisms for message transferring between distributed system and remote procedure call/response process. This SOAP is able to be used independently on operating systems and program languages because it is a protocol using text-based XML. Therefore ebXML that is a standard for E-Commerce adopts SOAP for message exchange. In this paper, we developed XML Hub System based on SOAP. By this XML Hub System we can more easily exchange messages for B2B. Accordingly we can manage and integrate a lots of business partners.

키워드 : SOAP, 전자상거래(eBusiness), XML 허브 시스템(XML Hub), XML

1. 서 론

최근 인터넷 기술의 급속한 보급과 발전은 컴퓨팅 패러다임의 변화를 촉진시키고 있다. 이러한 컴퓨팅 패러다임 변화의 중심에는 인터넷을 기반으로 네트워크화 되는 e비즈니스가 등장하였다. e비즈니스의 목적은 고객, 회사, 공급사와 협업 체계를 구축하여 신속한 의사결정을 할 수 있게 한다. 저비용, 고효율을 위한 비즈니스 프로세스 자동화를 이루며 이로 인해 정확하고 신속한 제품/서비스 전달을 위한 생산, 조달시스템의 효율을 높일 수 있다. 또한 고객의 유지, 유치 및 관계 구축을 위한 마케팅 수단으로 이용할 수 있다.

이러한 e비즈니스의 목적으로 기업들은 사업에 e비즈니스를 도입하려 하고 있다. 기업의 사업에 e비즈니스를 도입하기 위해서는 많은 기업들의 통합 환경이 제공되어야 가능하기 때문에 표준안이 중요한 위치를 차지하고 있다. 이로 인해 ebXML과 같은 전자상거래 표준이 등장하였고 ebXML은 모

든 거래 당사자들에게 운용적이며 안전하고 일관성있는 방법으로 광범위한 e비즈니스가 일어날 수 있도록 개방된 XML 기반의 기반 구조를 제공하고 있다. 이러한 ebXML은 메시지 교환을 위해 SOAP을 이용하고 있다[1].

본 논문에서는 SOAP을 기반으로 XML Hub 시스템을 개발하려고 한다. 전자상거래를 위해 기업들은 다양한 거래 파트너를 관리해야 한다. 따라서 보다 효율적인 관리가 필요하며 자주 발생하는 거래 파트너의 변화에 신속히 대처하기 위해 거래 파트너를 관리하고 메시지 전송을 담당해주는 시스템이 필요하다. 또한 거래 파트너와의 메시지 교환에 있어서 메시지 전송에 대한 성공/실패를 담당하여 응답 지연과 실패에 대한 처리가 필요하다. 본 논문에서는 XML Hub 시스템을 통해 전송 실패의 문제를 해결하고자 한다.

2. 관련 연구 및 기술

2.1 관련 연구

현재 XML Hub 시스템을 운영하는 국내 사이트로 GXML Hub라는 이름으로 KTNET에서 서비스 하고 있다. KTNET

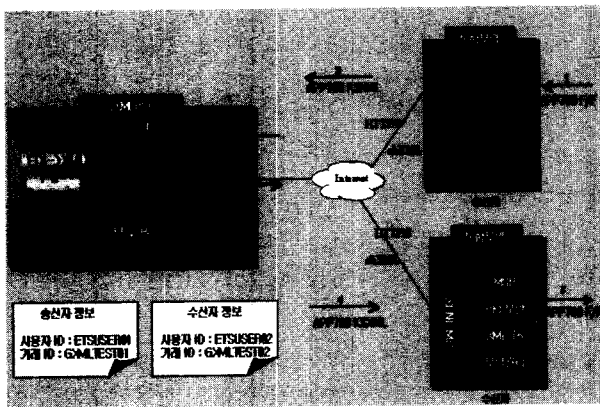
[†] 준 회원 : (주)아이티포럼 근무

^{††} 정 회원 : 순천향대학교 정보기술공학부 교수

논문접수 : 2002년 10월 4일, 심사완료 : 2003년 1월 3일

은 무역자동화 사업을 성공적으로 수행키 위해 국내 최초로 UN/EDIFACT EDI(전자자료 교환) 국제표준을 도입, 실무에 적용하여 사업을 완성시켰으며, 국내 EDI 기술분야에 선구자가 되었다.

GXMLHub는 XML/EDI 상호 변환 및 아시아 7개국의 참여하는 PAA, 일본의 마켓플레이스와의 실시간 연동이 가능한 eAMP Service, XML간 변환서비스, 파트너별 Business Process지원 등, ebXML을 완벽히 소화하는 Vitria의 ebXML 엔진을 기반으로 여러 종류의 XML 및 기존의 EDI와의 변환이 가능한 시스템을 제공한다[13].



(그림 1) GXMLHub 구조도

GXMLHub는 현재 서비스를 하고 있는 단계이므로 본 논문에서 구현한 XML Hub 시스템보다 많은 기능을 가지고 있다. GXMLHub 시스템은 ebXML의 표준을 따르고 있다. 현재 전자상거래 표준은 ebXML 뿐만 아니라 Rosettanet, eCo, Biztalk 등이 있다. 이들은 서로 다른 표준을 가지고 있기 때문에 통합이 필요하다. 따라서 본 논문에서는 단순히 XML 메시지의 형태를 SOAP 봉투화하고, SOAP 메시지의 전송을 담당하는 XML Hub 시스템을 구현하였다.

2.2 관련 기술

SOAP은 확장성이 좋은 일종의 메시지 포맷 프로토콜이며 HTTP, SMTP와 같은 다양한 인터넷 표준 통신 프로토콜을 이용해서 전달할 수 있다. SOAP은 XML 기반 통신 프로토콜로 서로 다른 컴퓨터 사이에서 운영체제와 프로그래밍 환경, 그리고 객체 모델 프레임워크가 달라도 정보 교환이 가능하도록 설계되었다. SOAP에서 정의하고 있는 것은 메시지를 교환하기 위한 단순한 방식과 모듈화된 패키지를 만드는 방법, XML 스키마에 기반을 두고 메시지 내부에 데이터를 인코딩하는 방법이 전부다. SOAP은 의사소통의 메시지 포맷을 정의하고 있는데, 이러한 면에서 SOAP을 가장 쉽게 이해할 수 있는 예는 편지다. 편지를 쓸 때는 항상 주소를 적는 것처럼 SOAP에서도 통신할 컴퓨터의 주소를

적는다. 그리고 편지를 보낼 때 규격봉투에 넣어서 보내야 하는 것처럼 SOAP에서도 데이터 페이로드에 대한 규격봉투를 이용해야 한다. 이러한 의미로 SOAP을 포장하는 부분을 '엔벨로프(envelope)'라고 한다[11].

2.2.1 SOAP의 독립성

SOAP은 XML과 HTTP를 이용해 어떤 인터페이스의 메소드를 호출할 것이며, 이 메소드에 대한 매개변수를 알리는 역할만 담당한다. 결론적으로 말해, SOAP은 JAVA, CORBA, COM 등의 분산 객체 기술에 구애받지 않는 프로토콜이며 심지어 이들 분산 객체 기술을 전혀 사용하지 않는 어플리케이션도 SOAP을 통해 원격 프로시저 호출이나 데이터 전송을 수행할 수 있다. SOAP이 원격 객체의 구현에 대해 언급하고 있지 않으므로, 원격 객체는 어떤 프로그래밍 언어로도 구현할 수 있다[2].

2.2.2 인터넷 적용성

SOAP은 transport로서 HTTP를 사용한다. HTTP를 사용함으로써 얻을 수 있는 장점은 인터넷에서 널리 사용할 수 있다는 점이다. 실제로 SOAP은 인터넷에서 원격 객체를 액세스하기 위해 고안된 프로토콜이다. JAVA, CORBA, COM 등이 사용하는 transport는 고유의 transport로서 RMI, IIOP, DCOM을 사용하지만, 이들 프로토콜은 TCP/IP 포트를 동적 할당하는 메커니즘을 사용하고 있다. 이것 때문에 이들 프로토콜이 방화벽을 통과하는데 문제점이 드러난다[2].

예를 들어 DCOM을 인터넷에서 사용하려면 방화벽에 일정 영역의 TCP/IP 포트를 열어 놓아야 하는데, 이것은 방화벽을 사용하지 않는 것과 다름없게 만들어 버린다. 반면 대부분의 기업 네트워크는 자사의 웹 사이트를 위해 HTTP가 사용하는 포트를 열어 놓고 있다. SOAP은 HTTP를 사용하므로 아무런 문제 없이 방화벽을 통과할 수 있다. 또한 대부분의 방화벽 제품은 HTTP 헤더의 내용을 읽어 필터링을 수행할 수 있으므로 특정 인터페이스의 특정 메소드를 호출하는 SOAP 메시지만을 통과시키도록 설정할 수도 있다. SOAP은 HTTP를 통해 인터넷에 분산된 객체들에 접근할 수 있으며 방화벽이나 HTTPS 등의 인터넷 보안 기술을 그대로 적용받을 수 있는 것이다.

2.2.3 SOAP 메시지와 e비즈니스의 관계

SOAP 메시지 구조와 구문은 XML 어휘를 갖는 특정 산업으로 하여금 SOAP 메시지는 필요 조건들을 충족시킨다. e비즈니스에 속하는 기업들은 SOAP 메시지 교환에 대한 노력의 결과로 다른 산업과의 상호 연동성을 얻을 수 있다. 어떤 기업도 자체 공급사슬 내에서만 독자적으로 비즈니스 거래를 한다고 할 수 없고, 모든 기업들은 그들 산업영역 내에서의 거래뿐만 아니라 밖과의 메시지 교환이 필요로

하게 된다[8, 10].

만일 회사가 전자 비즈니스 자료를 아직 교환하지 않으면, SOAP은 이 메시지들을 송/수신하는 연결을 만들고 다른 파트너들을 인증하는 것, 메시지 내용을 편집하는 것 그리고 내부 시스템에 자료를 매핑하는 것을 의미한다.

인터넷과 이용 가능한 다른 네트워크들의 장점을 취함으로써, SOAP은 전보다 더 넓은 지역에서 더 많은 잠재적 거래 파트너들과의 비즈니스 기회를 제공한다. SOAP은 이러한 네트워크에서 접속 가능한 세계 어디에서라도 비즈니스 데이터 교환을 위한 하나의 단일화된 프로토콜이다.

본 논문에서는 이러한 SOAP을 이용해 e비즈니스를 실현하기 위한 기업의 메시지 전송을 담당하는 XML Hub 시스템을 설계한다.

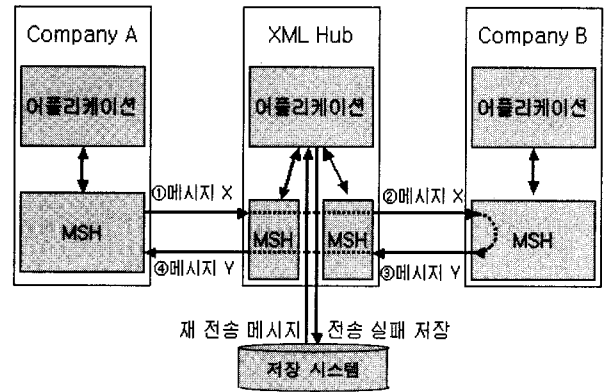
3. XML Hub 시스템

3.1 XML Hub 시스템 구조

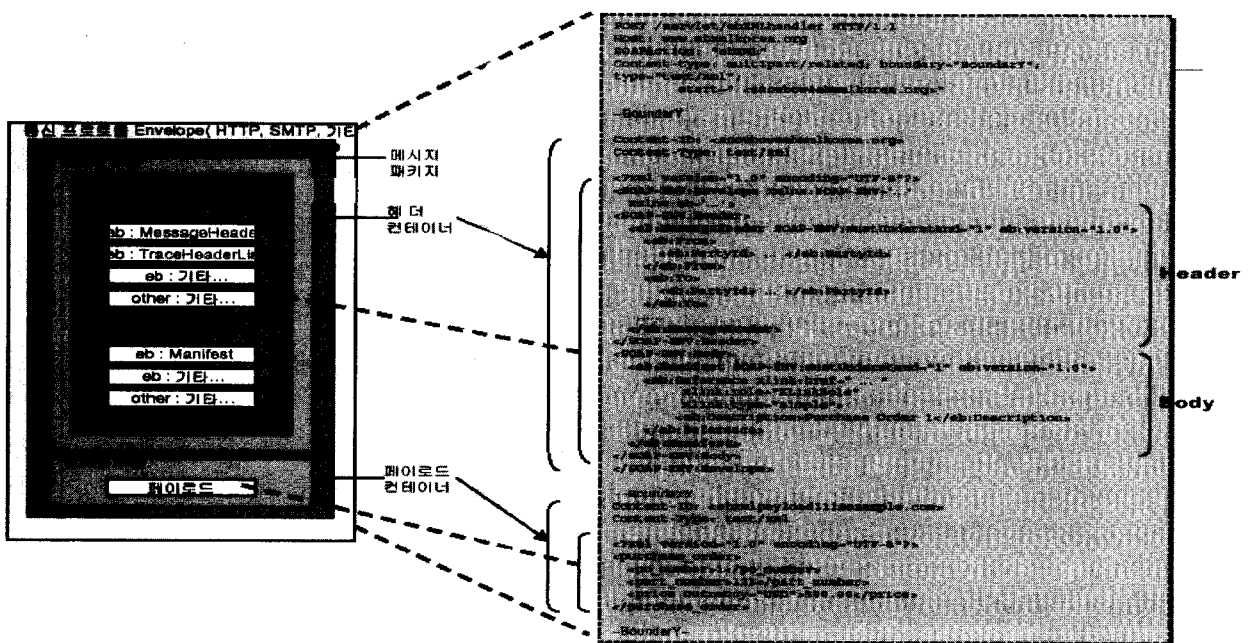
XML Hub 시스템은 e비즈니스에서 사용되는 XML 문서를 SOAP을 이용해 다양한 거래 파트너에게 정확히 전달하는 기능이 주된 기능이다. (그림 2)는 전송에 사용되는 HTTP 포트 방식을 이용한 SOAP 메시지의 한 예이다. 한편의 SOAP 메시지 구조와 함께 메시지를 구성하는 요소(elements)들이 있다.

SOAP 메시지는 웹-폼드 XML 문서로서 표현하며, (그림 2)와 같은 구조를 갖는다. SOAP 메시지의 루트 엘리먼트는 Envelope라는 이름의 엘리먼트이며, 이 엘리먼트는 Header, Body라는 두 개의 엘리먼트와 애플리케이션에서 정의하는 그 밖의 엘리먼트들을 자식 엘리먼트로 가질 수 있다. Header 엘리먼트는 생략될 수 있으나 생략되지 않는 경우 Envelope 엘리먼트의 다른 자식 엘리먼트들보다 앞에 와야 한다. Body 엘리먼트는 생략할 수 없으며 Header 엘리먼트를 제외하고는 Envelope 엘리먼트의 다른 자식 엘리먼트들보다 앞에 와야 한다. Body 엘리먼트는 메시지의 본체가 되는 부분으로서 메시지의 내용을 담는다. Header 엘리먼트는 메시지에 대한 부가적인 정보를 기술하는 부분으로서 신원확인(authentication) 정보, 트랜잭션 관리 정보, 지불(payment) 관련 정보 등을 기술하는데 사용한다[14].

SOAP 메시지 요청과 응답부분을 메시지 서비스 핸들러라고 하는 MSH가 담당한다. (그림 3)은 XML Hub를 이용한 두 기업간 메시지 처리부분을 보여주고 있다.



(그림 3) XML Hub 시스템 구조 다이어그램



(그림 2) SOAP 메시지 구조

기업 A에서 메시지를 XML Hub 시스템에 전송하고 XML Hub 시스템은 기업 A가 최종 보내려고, 하는 기업 B에 메시지를 전송한다. 전송 결과에 대한 성공 메시지를 받을 성공 여부를 기업 A에 전해 주고, 기업 B의 내부 사정으로 메시지 전달에 대한 성공 여부가 도착하지 않을 경우 전송하려는 메시지를 저장 시스템에 저장한 후 일정 기간이 지난 후 재전송 한다.

3.2 XML Hub 시스템 기능

SOAP을 이용하여 XML Hub 시스템을 개발함으로써 다양한 클라이언트 환경과 어플리케이션에 독립적인 수행이 가능하다. 이것은 현재 많은 기업들이 가지고 있는 비즈니스 파트너와 쉽게 메시지 교환을 할 수 있는 시스템이 된다. 따라서 기업 스스로가 다양한 거래 파트너를 모두 관리해야 하는 부담을 줄일 수 있으며, 자주 발생하는 거래 파트너의 변화에 신속히 대처하여 이를 관리하고 처리해주는 기능을 가지고 있다. 본 논문에서 구현한 XML Hub 시스템은 메시지 전송능력을 가지고 있으면 또한 수신을 확인하고 수신에 대한 응답이 없을 경우 메시지 전송을 다시 함으로써 전송시 발생하는 문제를 해결할 수 있다.

XML Hub 시스템에서 서버 기능을 수행하며, 받은 메시지에 대하여 헤더처리, 암호화/복호화 및 보안, 에러/예외처리, 저장, 라우팅, 호출 6가지 기능을 수행한다.

3.2.1 헤더 처리

메시지를 수신했을 때 XML Hub 시스템에서 가장 먼저 수행되는 기능 가운데 하나이다. 헤더 처리는 수신된 메시지의 헤더 영역을 확인하고, 거기에 포함된 기능 수행을 포함한다. 헤더 처리에 특정번호를 헤더에 추가하여 추적 가능하게 하거나 헤더 정보가 유효하게 구성됐는지를 검증할 수 있다. XML 메시지 내의 수신자가 적절함을 처리과정에서 사전에 검사할 수 있다.

3.2.2 보 안

보안 관점에서 XML Hub 시스템은 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 XML Hub 시스템은 메시지에 포함된 기능이나 처리에 인증을 받는다. 이러한 과정이 완료되면 메시지를 대칭키 혹은 비대칭키 알고리즘을 이용해서 암호화 또는 복호화하게 된다. 현재 이러한 보안과 관련된 전반적인 기능은 PKI(공개키 기반구조) 기반

의 전자서명을 통해 이루어진다.

3.2.3 오류와 예외 처리

오류와 예외 처리는 XML Hub 시스템이 수행하는 중요한 기능 중 하나다. XML Hub 시스템이 수신한 메시지가 유효하지 않거나, 최종 수신자에게 전송 할 수 없는 경우에는 에러 메시지를 송신자 쪽에 보내야만 한다. 또한 XML Hub 시스템의 문제로 인해 서비스를 제공할 수 없는 경우에도 해당 메시지를 송신자 쪽에 전송한다.

3.2.4 저 장

XML Hub 시스템은 오류 처리의 방법으로 전송 실패시 파일 시스템에 저장하고 일정 시간 경과 후 메시지를 재전송 한다. 저장 시스템은 관계형 데이터베이스 또는 XML 전용 저장 시스템이 있지만 XML 문서를 저장하고 다시 XML 문서로 변환하는 것이 쉬운 일이 아니므로 파일 시스템을 이용하여 잠시 보관하는 형태가 적당하다.

3.2.5 라우팅

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

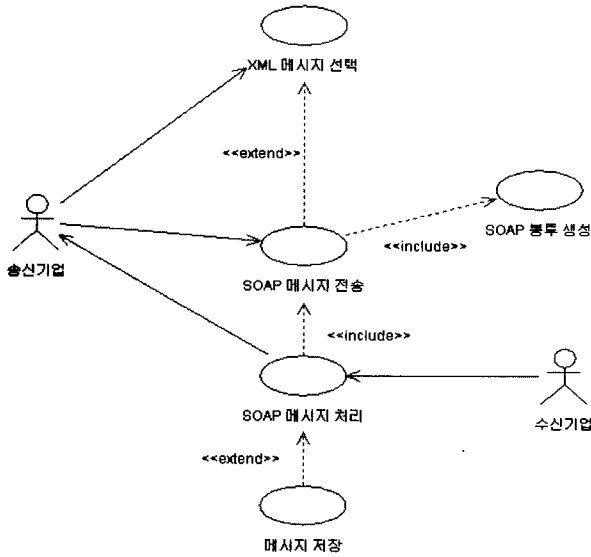
3.2.6 호 출

호출 단계에서는 실제 수신 메시지에 있는 페이로드의 자료를 가지고 메소드를 호출하게 된다. 여기서는 메소드 호출을 통해 XML Hub 시스템에서 송신 기업으로 반환할 수행 결과가 만들어진다.

4. XML Hub 시스템 설계

4.1 유즈케이스 다이어그램

(그림 4)는 유즈케이스 다이어그램을 나타낸다. 사용자는 XML 문서를 선택하고, 선택한 문서를 SOAP 봉투 처리를 하고, SOAP 메시지를 전송한다. 수신 기업은 받은 SOAP 메시지를 처리하고 결과를 송신 기업에 전송한다. SOAP 메시지를 전송하는데 SOAP 봉투 생성은 반드시 필요하므로 <<include>>로 표현하였고, XML 메시지 선택은 단일 작업이 아니므로 <<extend>>로 표현하였다. SOAP 메시지 처리에서 SOAP 메시지가 전송되어야 처리가 가능하므로 <<include>>로 표현하였고 메시지 저장은 상황에 따라 저장하기도 하고 그렇지 않을 수도 있기 때문에 <<extend>>로 표현된다[7].



(그림 4) 유즈케이스 다이어그램

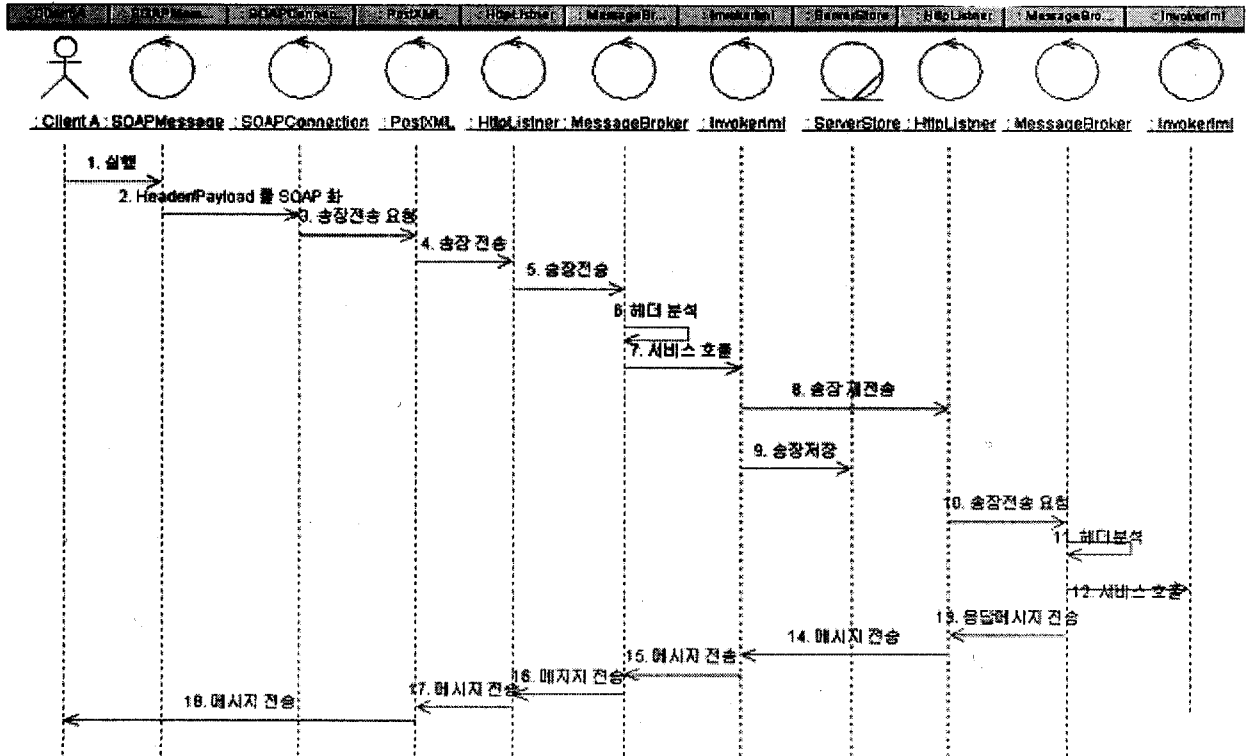
4.2 순차 다이어그램

(그림 5)는 기업 A가 XML 문서를 생성하여 XML Hub 시스템에 전송하고, XML Hub 시스템은 다시 기업 B에 XML 문서를 전송하여 처리 결과를 응답하는 순차 다이어그램을 나타낸 것이다. 기업 A는 사용자 인터페이스를 통하여 요청 서비스를 포함하는 XML 문서를 작성하고 XMLMessage를 통하여 SOAP 봉투를 작성한다. 이렇게 작성된 문서

를 사용자가 HTTP를 통해 문서를 전송한다. 그리고 XML Hub 시스템에서는 다시 XML 문서를 다른 기업에 재전송하고 기업으로부터 관련된 처리를 요청한다. 최종 전송된 기업에서 HTTP 요청을 받아서, 이를 처리할 MessageBroker에 전달한다. MessageBroker는 먼저 헤더를 분석해서 송신 기업에서 요청한 서비스를 확인한다. 그 다음에 해당되는 서비스를 호출하는데, 현재는 XML 문서의 저장만을 대상으로 하므로 InvokerImpl가 호출되고, 이 클래스는 저장소에 저장한 후, 처리가 성공했음을 알리는 XML 메시지를 만든다. 작성된 메시지는 Broker 시스템에 성공/실패 메시지를 전달하고 XML Hub 시스템 다시 송신 기업에 성공/실패 메시지를 전달한다.

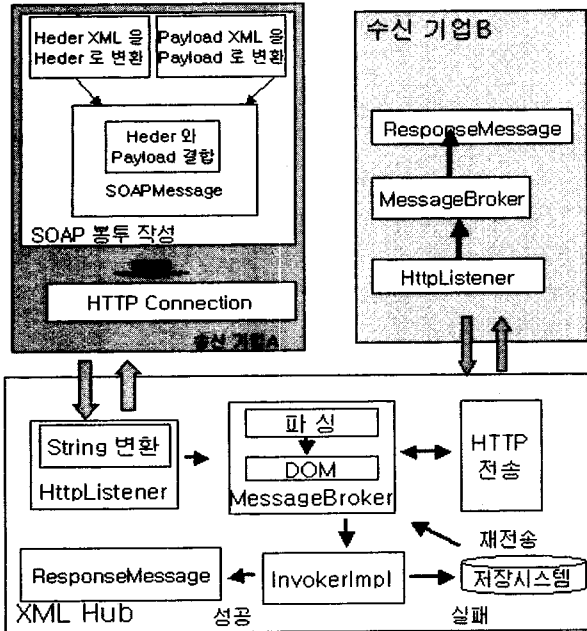
4.3 XML Hub 시스템

XML Hub 시스템의 처리과정은 (그림 6)과 같다. 송신 기업 A에서는 사용자가 작성한 XML 문서를 SOAPHeader와 SOAPPayload를 이용하여 헤더와 페이로드를 작성한다. 또한 이렇게 Heder와 Payload를 작성한 후 SOAPMessage를 이용하여 Heder와 Payload를 SOAP 봉투화 한다. 이렇게 작성된 메시지를 HTTP를 이용하여 전송한다. 이렇게 송신자가 메시지를 만들어서 XML Hub 시스템에 보내게 되면, XML Hub 시스템 다시 수신 기업에 전송하여 Http Listener가 받고 이 메시지로 어떤 작업을 처리해야 할지를 결정하



(그림 5) 순차 다이어그램

는 MessageBroker, 그리고 헤더에 요청된 작업을 실제로 처리하는 Invoker로 이루어진다. 그 이외에 전송 성공 여부를 되돌려 주는 ResponseMessage가 있다[3, 5, 6].



(그림 6) XML Hub 시스템

4.3.1 SOAPMessage

SOAPHeader와 SOAPPayload를 이용하여 작성한 것을 SOAPMessage에서 구성하게 된다. 즉, 두 개의 서로 다른 Heder 파일과 Payload 파일을 하나로 구성하는 기능을 가진다. 이곳에서 작성한 메시지는 connection에게 넘겨준다.

4.3.2 SOAPConnection

SOAPMessage에서 받은 파일을 HTTP를 이용하여 전송하게 된다. 전체 메시지를 구성하는 각각의 메시지의 특성들을 Content-Type과 Type, Bounder, Version 등을 통하여 설정할 수 있도록 구성한다.

4.3.2 HttpListener

HttpListener은 송신자가 POST 메소드로 넘어온 XML 메시지를 문자열로 받는 서블릿 클래스이다. HttpListener는 (그림 5)의 메시지 구조를 이용하여 작성한 XML 문서를 Stream 값으로 받은 다음, Reader로 변환한 후, 스트링으로 읽고, 그 코드를 MessageBroker에 넘겨준다.

4.3.3 MessageBroker

HttpListener에서 얻어온 값을 이용하여, 파싱해서 DOM을 만들고, invoke()로 DOM을 넘겨서 XML 문서에서 <type> 태그에 있는 텍스트 값을 읽는다. 현재의 구현은 <type> 태그에는 SavePO의 값을 가지고 있어 최종 수신자에게 전

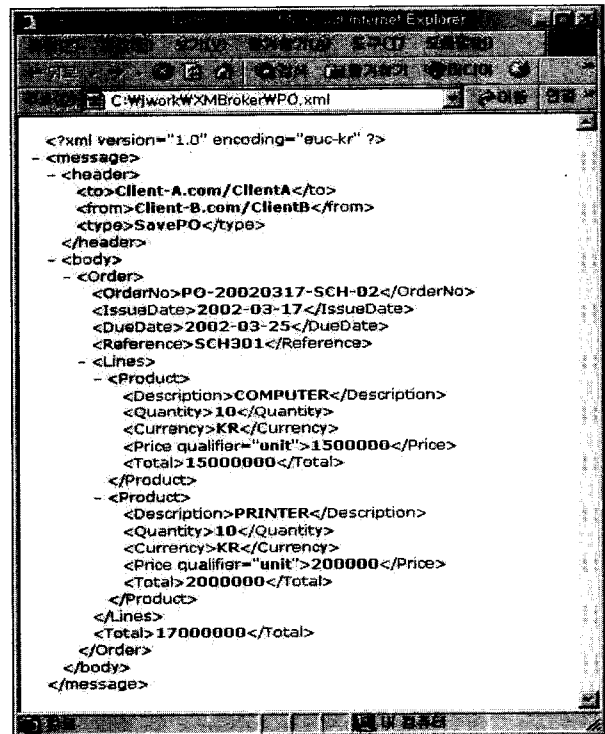
달된 XML 메시지를 저장하게 된다. <type>의 형태에 따라 ERP 시스템과 직업 연계되어 생산 계획을 수립하거나 송장을 전송하는 기능등 여러 형태로의 추가가 가능하다. 만일 <type> 태그가 없을 경우에도 기본적으로 XML 문서를 저장하게 되어있다.

4.3.4 InvokerImpl

Invoker 인터페이스를 구현한 것인 InvokerImpl 클래스이다. 여기에서는 파일 시스템에 송장을 저장하고, ResponseMessageMaker에서 응답 메시지를 만든다. ResponseMessage Manager는 요청이 성공적으로 이루어 졌을 때는 송신자에 보낼 '성공' 메시지나, 실패했을 경우의 '에러' 메시지를 XML 문서로 만든다.

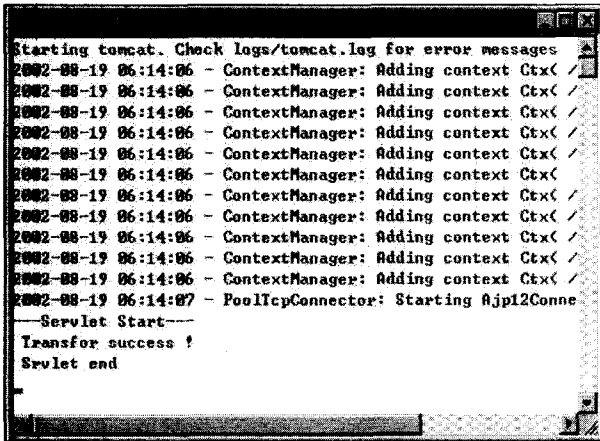
5. XML Hub 시스템 구현

XML Hub 시스템의 환경으로 운영체제는 Windows 2000 Server, 웹 서버로는 Jakarta Tomcat V3.1.3을 사용하였으며, 아파치 SOAP2.2, XML Parser V2를 이용하여 구축하였다. 다음 (그림 7)은 XML 문서를 이용하여 작성된 송장을 나타낸다. 송장에서 헤더 부분과 페이로드 부분으로 나뉘는 것을 볼 수 있을 것이다. OrderNo를 이용하여 각 송장에 일련번호를 부여하였으며, 송장은 XML Hub 시스템에 전달되어 저장 시스템에 저장되고 최종 수신자에게 다시 전송된다.

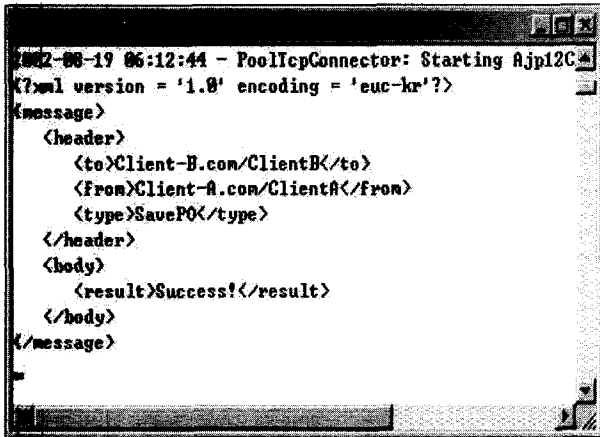


(그림 7) sample XML 문서

(그림 8)은 송신자가 메시지를 받을 때 나타나는 화면이다. 메시지 전송이 시작되면 Servlet이 시작되고, 전송이 완료되면 Servlet이 끝나며, 전송 결과는 XML Hub 시스템에 전달되어 (그림 9)와 같은 전송 내용과 결과를 웹 서버의 실행 창에 보여 주게된다.



(그림 8) 수신측 실행 화면



(그림 9) XML Hub 시스템 실행 화면

본 논문에서 구현한 SOAP을 기반으로 한 XML Hub 시스템은 XML 메시지의 전송을 담당하는 중계 기관이다. 본 논문에서는 기업과 기업간의 직접적인 전송도 가능하겠지만 이렇게 할 경우 하나의 기업에서 B2B를 실현하고자 하는 많은 기업을 모두 관리해야 한다. 따라서 본 논문에서는 중계 역할을 담당하는 XML Hub 시스템을 구현하였고, XML Hub 시스템은 메시지 전송의 성공/실패를 확인할 수 있으며 실패시 재전송 할 수 있도록 하였다.

6. 결론 및 향후 연구 방향

e비즈니스를 위해 다양한 분야의 기업간 협력이 필요하다. 그러나 실질적으로 기업간의 의사소통은 온라인상의 문

서전달을 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 것이 XML이다. 데이터를 XML로 표현하고 전송수단을 HTTP 환경의 SOAP 표준을 이용한다면 지금까지 가지고 있던 여러 가지 문제를 한번에 해결 수 있는 것이다.

본 연구는 SOAP을 이용하여 XML Hub 시스템을 개발함으로써 다양한 클라이언트 환경과 어플리케이션에 독립적으로 수행 가능한 시스템이 된다. 많은 기업들은 다양한 거래 파트너를 모두 관리해야 하는 부담을 줄일 수 있게 된다. 따라서 보다 효율적인 관리가 필요하며 자주 발생하는 거래 파트너의 변화에 신속히 대처하여 이를 관리하고 처리해주는 시스템이 된다. 본 연구에서는 XML Hub 시스템을 통해 이를 해결하고자 한다. XML Hub 시스템은 메시지 전송능력을 가지고 있으면 또한 정확한 수신을 확인하고 수신에 대한 응답이 없을경우 메시지 전송을 다시 함으로써 전송시 발생하는 문제를 해결할 수 있다.

참 고 문 헌

- [1] ebXML, <http://www.ebXML.org>.
- [2] Simple Object Access Protocol(SOAP), http://www.microsoft.com/korea/msdn/workshop/xml/general/SOAP_White_Paper.asp.
- [3] Microsoft, "Biztalk Framework 1.0 Independent Document Specification," <http://www.biztalk.org>.
- [4] Simple Object Access Protocol(SOAP)1.1, <http://www.w3.org/TR/SOAP>.
- [5] Apache SOAP Documentation : User's Guide <http://xml.apache.org/soap/docs/index.html>.
- [6] Bequet, Henry. Professional Java SOAP, Wrox, 2001.
- [7] DAVID CARLSON, MODELING XML APPLICATIONS WITH UML, Wesley, 2001.
- [8] 신동규, "XML/EDI 시스템의 설계 및 구현", 정보처리학회 논문지D, 제8-D권 제2호, 2001.
- [9] 김채미, 전문가와 함께가는 XML Camp, 아이트Press, 2001.
- [10] 김채미, 최학열, 글로벌 e비즈니스 리더를 위한 ebXML, 대청미디어, 2001.
- [11] 정지훈, 웹 서비스, 한빛미디어, 2002.
- [12] 한국 ebXML로럼, <http://www.ebxmlkorea.org>.
- [13] Single Global e-Trade Service, <http://gxmlhub.com>.
- [14] 김윤명, XML을 위한 JAVA Programming, 가남사, 2002.



김용수

e-mail : yskim@it4web.com

2001년 순천향대학교 전산학과(학사)

2003년 순천향대학교 일반대학원 전산학과
(공학석사)

2003년~현재 (주)아이티포웹 근무

관심분야 : Database Systems, EJB, 전자
상거래, B2B



주경수

e-mail : gssoojoo@sch.ac.kr

1980년 고려대학교 이과대학 수학과(학사)

1985년 고려대학교 일반대학원 전산학과
(석사)

1993년 고려대학교 일반대학원 전산학과
(박사)

1986년~현재 순천향대학교 정보기술공학부 교수

관심분야 : Database Systems, System Integration, Object-
oriented Systems.