

SOAP을 이용한 XML Hub 시스템 구현 및 활용

김용수*, 주경수*

An Implementation and Application of XML Hub System using SOAP

Kim Yong-Soo, Joo Kyung-Soo

Abstract

SOAP is a protocol used for information exchange in distributed computing environment and SOAP defines mechanisms for message transferring between distributed system and remote procedure call/response process. This SOAP is able to be used independently on hardware platforms and operating systems and program languages because it uses only a text-based XML for protocol. Therefore ebXML that is a standard for E-Commerce adopts SOAP for message exchange.

In this paper, we implement XML Hub System using SOAP and apply on the system e-Procurement. By this XML Hub System we can more easily exchange messages for B2B. Accordingly we can manage and integrate a lots of business partners.

* 순천향대학교 전산학과

1. 서론

최근 인터넷 기술의 급속한 보급과 발전은 컴퓨팅 패러다임의 변화를 촉진시키고 있다. 이러한 컴퓨팅 패러다임 변화의 중심에는 인터넷을 기반으로 네트워크화되는 e비즈니스가 등장하였다. e비즈니스의 목적은 고객, 회사, 공급사와 협업 체제를 구축하여 신속한 의사 결정을 할 수 있게 한다. 저 비용, 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 ebXML이 등장하였고 ebXML은 모든 거래 당사자들에게 운용 적이며 안전하고 일관성 있는 방법으로 광범위한 e비즈니스가 일어날 수 있도록 개방된 XML 기반의 기반 구조를 제공하고 있다. 이러한 ebXML은 메시지 교환을 위해 SOAP을 이용하고 있다[4].

본 논문에서는 이러한 SOAP을 기반으로 XMLHub 시스템을 구현하였고, e-Procurement에 활용하였다. XML Hub는 서로 다른 거래 당사자들간에 XML 문서를 정확하게 주고받을 수 있도록 하는 XML 메시징(XML Messaging)이다. 이것은 e비즈니스 환경에서 기업들은 다양한 거래 파트너를 관리하게 된다. 이로 인해 보다 효율적인 관리가 필요하며 자주 발생하는 거

래 파트너의 변화에 신속히 대처하기 위해 이를 관리하고 처리해주는 시스템이 필요하다. 또한 거래 파트너와의 메시지 교환에 있어서 메시지 전송에 대한 요청/응답을 담당하고 응답 지연과 실패에 대한 처리가 필요하다. 본 논문의 XML Hub 시스템을 통해 이를 해결하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로써 e비즈니스를 추진중인 표준들에 대하여 설명하고, 관련 기술로 분산 시스템간에 주고받는 메시지의 구조를 정의하는 SOAP에 대해 설명한다. 3장에서는 XML Hub 시스템의 메시지 전송 구조와 기능에 대해 살펴보고, 4장에서는 구현한 시스템의 각각 모듈을 살펴보고 실행결과를 본다. 5장에서는 XML Hub 시스템을 적용하여 웹상에 e-Procurement 시스템을 구현하고 이를 이용한 기업간 전자상거래를 살펴본다. 마지막으로 6장에서는 본 논문의 결론을 맺는다.

2. 관련 연구 및 기술

2.1 관련 연구

현재까지 XML을 이용한 메시지 교환 시스템은 기존의 전통적인 EDI(Electronic Data Interchange) 시스템을 어떻게 XML로 메시지를 교환할 것인가에 초점이 맞추어져 있다. EDI/XML 시스템의 성공적인 구축은 데이터 교환 모델을 위하여 XML을 사용하고, 모습을 표현하기 위하여 XSL(XML Style Language)을 이용하며, 전통적인 EDI와 쉬운 통합 방안을 지니기 위하여 DTD를 사용한다. 또한 문서 중심의 조화와 처리가 가능케 하고, 타 정보 시

시스템과의 연동이 가능하도록 개발하는 것이다. 이러한 문제를 해결하기 위하여 전자상거래 표준 기술로서 XML이 등장하게 되었으며 광범위한 영역에서 입지를 확고히 해 나가고 있다. 그리고 여러 가지 장점을 가지고 있기 때문에 다양한 분야에서 적용이 되고 있다[15].

<표 1>은 각각 다른 방향으로 e비즈니스를 위해 표준화 작업을 하고 있는 전자상거래(EC) 프레임워크 비교표이다. 표와 같이 모든 프레임워크들은 XML 기반으로 움직이고 있다는 것을 알 수 있다. 또한 웹서비스, Biztalk, 그리고 최근에는 ebXML에서도 통신 수단으로 HTTP 위에서 SOAP을 기반으로 메시지를 전송하고 있다[3,5,6].

2.2 관련 기술

2.2.1 SOAP

SOAP은 분산 환경에서의 정보 교환에 사용

되는 경량(lightweight) 분산 컴퓨팅 프로토콜이다. SOAP 명세에는 분산 시스템간에 메시지를 전달하는 방법, 원격 프로시저 호출/응답을 처리하는 방법이 정의되어 있다. 그러나 SOAP 명세에는 단지 메시지의 구조에 대해서만 정의할 뿐, 어떤 의미(semantics)의 메시지 인지는 정의하지 않는다. 반면 SOAP을 전송 프로토콜로 사용하는 ebXML과 같은 명세에는 메시지의 의미가 정의되어 있다.

CORBA, RMI, DCOM과 같은 분산 컴퓨팅 프로토콜에서는 이진(binary) 포맷의 메시지를 사용한다. 그러나 SOAP에서는 텍스트 기반(text-based) XML을 프로토콜로 사용한다. 텍스트 기반의 XML이라는 점 덕분에 SOAP은 하드웨어 플랫폼, 운영체제, 프로그래밍 언어, 그리고 네트워크 하드웨어 플랫폼 전 영역에 걸쳐서 상호 운용성이 매우 높다. SOAP은 HTTP를 사용할 수 있기 때문에 기존의 웹 서버, 프록시 서버, 방화벽 같은 인프라를 그

<표 1> 전자상거래(EC) 프레임워크 비교표

	Industry Target	Co	UDDI	Rosettanet	Biztalk
Industry Target	Unspecified	Unspecified	Unspecified	IT & Electronic	Unspecified
Communication protocol	HTTP, SMTP,...	HTTP	HTTP(SOAP)	HTTP/CGI	HTTP(SOAP)/MSMQ
Repository	Distributed	Locally maintained	Distributed	Not addressed	Centralized repository
Message format	XML Document	XML Document	XML Document	XML Document	Biztalk document based on Biztag
Security	S/MIME, Digital Signature	Optional	Authentication in each UDDI Registry	SSL with HTTP; digital certificate and signature	Leverages existing standards
Ontology	Common Business Objects	Common Business Library	NAICS, UN/SPSC	Technical and Business Dictionaries	Collection of Biztag

대로 이용할 수도 있다. SOAP은 HTTP 외에도 SMTP나 JMS와 같은 프로토콜과도 함께 사용할 수 있다.

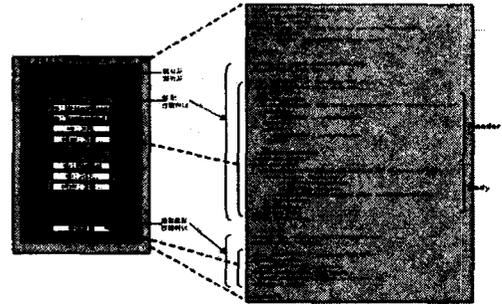
SOAP이 최초의 분산 컴퓨팅 프로토콜은 아니다. 그러나 SOAP은 거의 모든 소프트웨어 및 하드웨어 대형 벤더들로부터 지지를 받고 있는 최초의 프로토콜이다. CORBA, RMI, DCOM과 같은 프로토콜은 이러한 지지를 받지 못했기 때문에 어떤 개발자들은 CORBA만을 사용해서 분산 시스템을 구현하고, 또 어떤 개발자들은 DCOM만을 사용해서 분산 시스템을 구현해야 했다[1,2,7].

이렇게 개발 영역이 나뉘어진 상태에서, 서로 다른 분산 컴퓨팅 프로토콜을 사용하는 소프트웨어 컴포넌트들을 묶어서 하나의 분산 어플리케이션을 개발하기란 참으로 어려웠다. 그러나 SOAP은 서로 다른 프로토콜(CORBA, RMI, DCOM 등)을 사용하는 어플리케이션 위에 존재하여 중계자 역할을 함으로써, 일반적으로는 서로에게 접근할 수 없었던 시스템간의 객체들을 접근 가능하게 하였다. 예를 들어, 각각 CORBA, RMI, DCOM을 사용하는 세 가지 시스템 사이에 SOAP을 추가하면, 각 프로토콜의 세부 내용들을 변경하지 않고도 다른 시스템의 객체를 사용할 수 있는 것이다[9].

2.2.2 SOAP 메시지의 XML 표현

e비즈니스에서 사용되는 XML 문서를 SOAP 메시지 구조로 변환한 후 다양한 거래 파트너에 정확히 전달하는 기능이 XML Hub의 주된 기능이다. <그림 1>은 이에 사용되는 HTTP 포스트 방식을 이용한 SOAP 메시지의 한 예이다. 원편의 SOAP 메시지 구조와 함께

메시지를 구성하는 요소(elements)들이 있다.



<그림 1> SOAP 메시지 구조

SOAP 메시지는 웰-폼드 XML 문서로서 표현하며, <그림 1>과 같은 구조를 갖는다. SOAP 메시지의 루트 엘리먼트는 Envelope라는 이름의 엘리먼트이며, 이 엘리먼트는 Header, Body라는 두 개의 엘리먼트와 어플리케이션에서 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[12].

Body 엘리먼트는 메시지의 본체가 되는 부분으로서 메시지의 내용을 담는다. Header 엘리먼트는 메시지에 대한 부가적인 정보를 기술하는 부분으로서 권한(authentication) 정보, 트랜잭션 관리 정보, 지불(payment) 관련 정보 등을 기술하는데 사용한다[12].

2.2.3 SOAP 보안

다른 분산 컴퓨팅 프로토콜과는 달리 SOAP

은 전송 프로토콜로서 HTTP를 사용할 수 있다. 대부분 회사의 방화벽은 HTTP를 통해서 들어오는 데이터는 통과시키기 때문에 새로운 종류의 위협에 대비해야 한다. SOAP에서는 HTTP 요청 헤더 필드인 SOAPAction이라는 필드를 새로 정의함으로써 이 문제를 다루고 있다. HTTP 전송을 사용하는 SOAP 클라이언트는 SOAP 요청을 할 때 반드시 이 필드를 사용해야 한다. 이 필드에 들어가는 값은 URI로써 SOAP 메시지의 의도를 명확히 기술하는 역할을 한다. 이 필드 값은 방화벽에서 SOAP 메시지를 적절히 걸러낼 수 있도록 하는 데 사용된다.

2.2.4 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 어플리케이션을 만들어냈으며, 이러한 형태의 체계화가 전자상거래의 일반적인 접근 방식이다. 성공적인 전자상거래 어플리케이션은 다음에 설명하는 것 중 한 가지 이상의 서비스를 제공해야 한다[9,20].

2.2.5 SOAP 메시지와 전자상거래의 관계

SOAP 메시지 구조와 구문은 XML 어휘를 갖는 특정 산업으로 하여금 SOAP 메시지는 필요요건들을 충족시킨다. 이러한 산업에 속하

는 기업들은 이런 노력의 결과로서 다른 산업과의 상호 연동성을 획득 할 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.

만일 회사가 전자 비즈니스 자료를 아직 교환하지 않으면, SOAP은 이 메시지들을 송/수신하는 연결을 만들고 다른 파트너들을 인증하는것, 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 어디에서도 비즈니스 데이터 교환을 위한 하나의 단일화된 프로토콜이다.

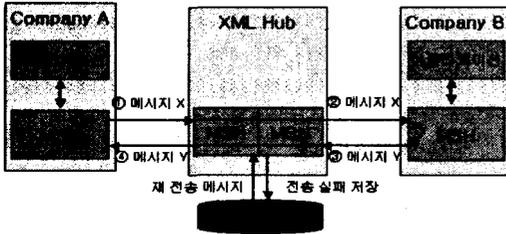
3. XML Hub 시스템

3.1 메시지 전송 구조

SOAP 메시지 요청과 응답 부분을 메시지 서비스 핸들러라고 하는 MSH가 담당한다. 그림 2는 XML Hub를 이용한 두 기업간 메시지 처리 부분을 보여주고 있다.

기업A 에서 메시지를 XML Hub 시스템에 전송하면 Hub 시스템은 다른 기업B 에 메시지를 전송한다. 전송 결과에 대한 성공 메시지를 받을 성공 여부를 기업A 에 전해 주고, 기업B 의 내부 사정으로 메시지 전달에 대한 성공 여부가 도착하지 않을 경우 전송하려는 메시지를 저장시스템에 저장한 후 일정 기간이

지난 후 재전송한다.



<그림 2> 메시지 전송 구조

3.2 XML Hub 시스템 기능

XML Hub 시스템에서 서버 기능을 수행하며, 받은 메시지에 대하여 헤더처리, 암호화/복호화 및 보안, 에러/예외처리, 저장, 라우팅, 호출 6가지 기능을 수행한다[10,11].

3.2.1 헤더처리

메시지를 수신했을 때 XML Hub 시스템에서 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.

3.2.2 보안

보안 관점에서 XML Hub 시스템은 보안의 가장 기본적인 조건인 신원 확인(authentication), 인증(authorization), 암호화(encryption), 부인방지(nonrepudiation)를 보장해야 한다. 메시지가 수신되면 메시지 핸들러는 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) 기반의 전자서명을 통해 이루어진다.

3.2.3 오류와 예외처리

오류와 예외처리는 XML Hub 시스템이 수행하는 중요한 기능 중 하나다. XML Hub 시스템이 수신한 메시지가 유효하지 않거나, 최종 수신자에게 전송할 수 없는 경우에는 에러 메시지를 송신자 쪽에 보내야만 한다. 그리고 XML Hub 시스템의 문제로 인해 서비스를 제공할 수 없는 경우에도 해당 메시지를 송신자 쪽에 보낸다.

3.2.4 저장

XML Hub 시스템은 오류처리의 방법으로 전송 실패시 파일 시스템에 저장하고 일정 시간 경과 후 메시지를 재전송한다. 저장 시스템은 관계형 데이터베이스 또는 XML 전용 저장시스템이 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 할지를 결정한다.

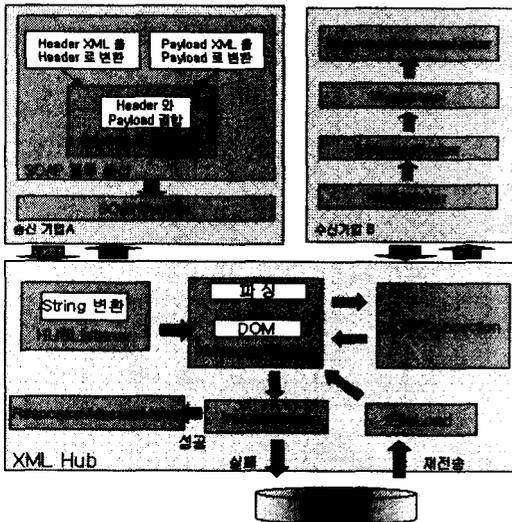
3.2.6 호출

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

4. XML Hub 시스템 구현

4.1 XML Hub 시스템

XML Hub 시스템의 각각 모듈과 전체 메시지 전송 흐름은 <그림 3>과 같다. 각각의 설명은 다음과 같다.



<그림 3> 전체 XML Hub 시스템 흐름

4.1.1 송신 모듈

송신기업에서는 사용자가 작성한 XML 문서를 SOAPHeader와 SOAPPayload를 이용하여 헤더와 페이로드를 작성한다. 이렇게 Header와 Payload를 작성한 후 SOAP-MessageMaker를 이용하여 Header와 Payload를 SOAP 봉투화 한다. 생성된 메시지를 SOAPTransfer 클래스를 이용하여 XML Hub 시스템에 전송한다.

4.1.2 XML Hub 모듈

송신자가 메시지를 만들어서 XML Hub 시스템에 보내게 되면, XML Hub 시스템은 최종 수신기업에 메시지를 전송하게 된다. 여기서 최종 수신자의 메시지 수신 여부에 따라 다음 동작을 하게 된다. 최종 수신자의 응답 메시지가 성공일 경우 전송 성공 여부를 송신자에 보내게 되고, 그렇지 않을 경우 전송하려는 메시지를 파일 시스템에 저장한다.

4.1.3 수신 모듈

HttpListener가 받고 이 메시지로 어떤 작업을 처리해야 할지를 결정하는 MessageBroker, 그리고 헤더에 요청된 작업을 실제로 처리하는 Invoker로 이루어진다. 그 이외에 전송 성공 여부를 되돌려 주는 ResponseMessage-Maker가 있다.

4.2 주요 클래스

4.2.1 SOAPMessageMaker

SOAPHeader와 SOAPPayload를 이용하여 작성한 것을 SOAPMessage에서 구성하게 된다. 즉, 두 개의 서로 다른 Header 파일과

Payload 파일을 하나로 구성하는 기능을 가진다. 이곳에서 작성한 메시지는 SOAPTransfer에 넘겨준다.

4.2.2 SOAPTransfer

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

4.2.3 HUBListener

HUBListener 은 송신자가 POST 메소드로 넘어온 XML 메시지를 문자열로 받는 서블릿 클래스이다. HUBListener는 작성한 XML 문서를 Stream 값으로 받은 다음, Reader로 변환한 후, 스트링으로 읽고, 그 코드를 Message Broker에 넘겨준다.

4.2.4 MessageBroker

HUBListener에서 보내준 스트링을 service-Sync()에서 처리한다. serviceSync()는 스트링을 파싱해서 DOM을 만들고, invoke()로 DOM을 넘겨서 XML 문서에서 <type> 태그에 있는 텍스트 값을 읽는다.

4.2.5 InvokerImpl

Invoker 인터페이스를 구현한 것인 Invoker-Impl 클래스이다. 여기에서는 파일 시스템에 송장을 저장하고, ResponseMessage-Maker에서 응답 메시지를 만든다. ResponseMessage Manager는 요청이 성공적으로 이루어졌을 때는 송신자에 보낼 '성공' 메시지나, 실패했을 경우의 '에러' 메시지를 XML 문서로 만든다.

4.2.6 ResponseMessageMaker

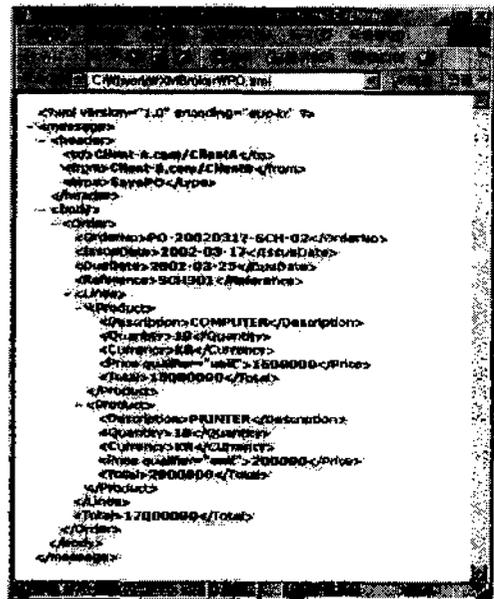
서비스 요청이 성공적으로 처리됐을 경우에 최초 송신자에게 보낼 '성공' 메시지나, 실패했을 경우의 '에러' 메시지를 XML 문서로 만든다. InvokerImpl에서 값을 받아 응답 XML 문서를 생성한다.

4.3 구현 환경 및 실행

XML Hub 시스템의 환경으로 운영체제는 Windows 2000 Server, 웹 서버로는 Jakarta Tomcat V3.2.3을 사용하였으며, SOAP API로는 아파치 SOAP2.2, 와 JAXM을, XML 파서로 XML Parser V2를 이용하여 구축하였다.

4.3.1 메시지 전송 과정

다음 <그림 4>는 XML 문서를 이용하여 작



<그림 4> sample XML 문서

성된 송장을 나타낸다. 송장에서 헤더 부분과 페이로드 부분으로 나뉘는 것을 볼 수 있을 것이다. OrderNo를 이용하여 각 송장에 일련 번호를 부여하였으며, 송장은 XML Hub 시스템에 전달되어 저장 시스템에 저장되고 최종 수신자에게 다시 전송된다.

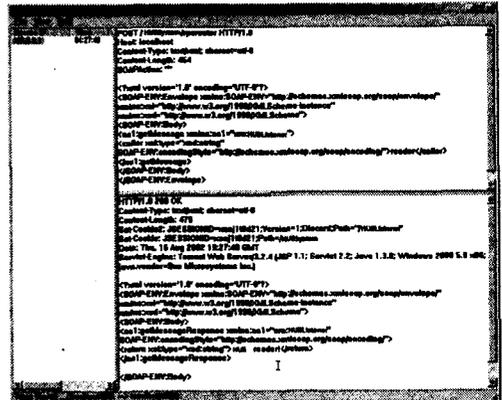
다음은 SOAP 메시지를 생성하는 부분으로 전송하려는 XML 문서에 SOAP 메시지를 만들게 되는 소스 코드이다. <그림 5>를 보면 SOAPMessageMaker를 이용하여 <그림 4>의 XML 문서를 SOAP 봉투화 하게 된다.

```
public SOAPMessage SOAPMessageMaker(String requestNo,
String reason) throws Exception {
    MessageFactory mf = MessageFactory.newInstance();
    SOAPMessage message = mf.createMessage();
    SOAPPart sp = message.getSOAPPart();
    SOAPEnvelope envelope = sp.getEnvelope();
    // 헤더 엔트리 추가
    SOAPHeader header = envelope.getHeader();
    SOAPHeaderElement headerElement=header.addHeaderElement
(envelope.createName("to", "from", "type"));
    SOAPElement idElement
= headerElement.addChildElement(envelope.createName("id"));
    SOAPElement passwordElement=
headerElement.addChildElement
(envelope.createName("password"));
    idElement.addTextNode(id);
    passwordElement.addTextNode(password);
    // 바디 엔트리 추가
    SOAPBody body = envelope.getBody();
    SOAPBodyElement bodyElement =body.addBodyElement
(envelope.createName("OrderNo", "IssueDate", "DueDate",
"Reference"));
    SOAPElement requestNoElement
=bodyElement.addChildElement(envelope.createName
("requestNo"));
    SOAPElement reasonElement
=
bodyElement.addChildElement(envelope.createName("reason"));
    requestNoElement.addTextNode(requestNo);
    reasonElement.addTextNode(reason);
    return message;
}
```

<그림 5> SOAP 메시지 생성 소스코드

<그림 4>의 코딩을 통해 <그림 5>의 XML 문서는 그림 6과 같이 SOAP 봉투화 하게 된다. <그림 6>은 SOAP 메시지가 생성되어 전송되는 메시지를 야파치 SOAP에서 제공

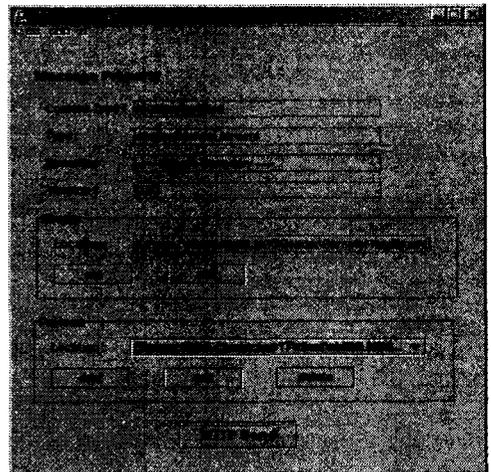
하는 tcpTunnel을 이용하여 전송결과를 확인할 수 있다.



<그림 6> SOAP 메시지 생성

4.3.2 실행 화면

<그림 7>은 XML 문서를 XML Hub 시스템에 전송하는 GUI 프로그램이다. GUI는 JAVA의 Swing 을 이용하여 구현 하였다.



<그림 7> 송신 GUI 화면

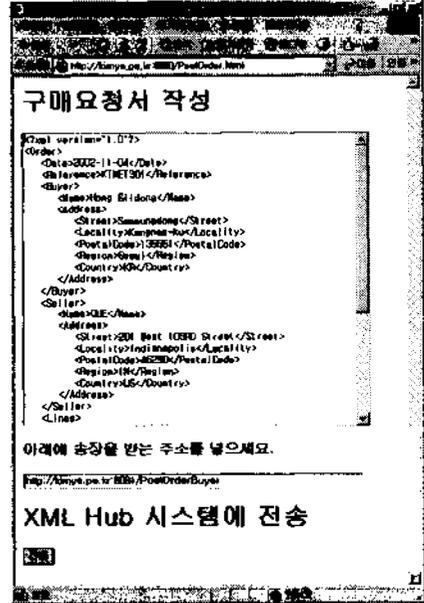
e-Procurement는 인터넷을 통한 물품 및 서비스의 B2B 거래를 의미한다. XML Hub 시스템은 송·수신 모듈과 XML Hub 모듈을 각각 구현하였는데 이를 인터넷 기반에 적용되는 e-Procurement 구조는 <그림 9>와 같다.

구매자는 XML Hub 시스템에 등록된 판매자 정보와 판매 상품을 인터넷을 통해 검색하고 구매 하려는 물품에 해당하는 구매 요청서를 작성한다. 구매 요청서가 접수되면 SOAP에 의해 판매자가 호출되고 구매 요청서가 전송된다. 접수된 구매요청서를 기준 하여 송장을 작성하고 송장을 구매자에게 SOAP 봉투화하여 전송한다. <그림 9>의 구조와 같이 XML Hub 시스템의 접근은 웹브라우저만 있으면 접근이 가능하며 XML Hub 시스템에서 전송하는 구매 요청서와 송장은 SOAP을 이용함으로 구매자와 판매자의 환경에 상관없이 구매 요청서와 송장을 전송할 수 있다.

5.2 시스템 실행

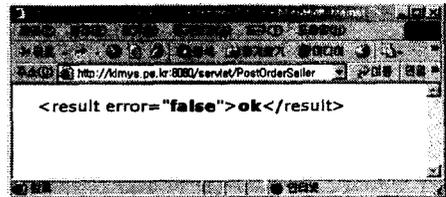
5.2.1 구매요청서 전송

XML Hub 시스템을 통한 B2B 전자상거래를 수행하는 과정을 보인다. XML Hub 시스템은 인터넷이 가능한 환경이며 모두 접근이 가능하도록 웹을 이용해 구현하였다. 구매 요청서는 웹브라우저에서 수정이 가능하도록 하였으며 구매 요청서에 따른 송장을 받게 되는 URL 주소를 입력하도록 하였다. <그림 11>을 통해 구매 요청서를 전송하는 화면을 볼 수 있다.



<그림 11> 구매요청서 전송

구매 요청서가 작성이 되고 전송을 클릭 하면 이것은 XML Hub 시스템에 전송된다. 전송된 결과는 <그림 12>와 같이 나타난다.



<그림 12> 구매 요청서 전송 결과

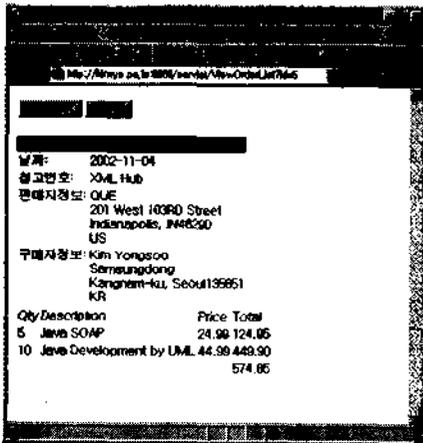
5.2.2 송장 전송

판매 기업은 전송된 구매 요청서를 XML Hub 시스템에서 SOAP을 통해 전달받고 XML Hub 시스템을 접속하여 접수된 구매 요청서를 확인할 수 있다. <그림 13>은 접수된 구매 요청서의 리스트이다.



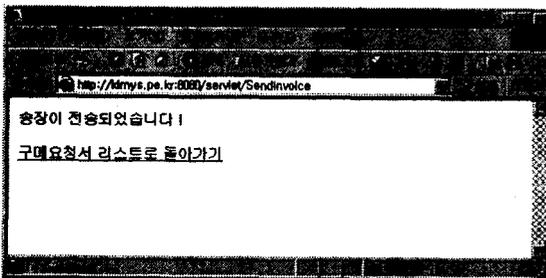
<그림 13> 구매요청 리스트

구매 요청 중 하나를 클릭하여 송장을 전송하면 구매 기업에게 송장이 전송된다. <그림 14>는 주문의 상세 내역과 송장 전송 화면이다.



<그림 14> 송장 전송

<그림 15>는 송장 전송이 완료된 것을 알려 주며 다시 구매 리스트로 돌아 갈 수 있다.



<그림 15> 송장 전송 결과

5.3 기대 효과

지금까지 XML Hub 시스템을 활용한 e-Procurement의 일부를 구현하였다. 전자 상거래에서 e-Procurement를 이용함으로써 다음과 같은 장점을 가지고 있다. 첫째, 구매 프로세스의 개선으로 구매 비용 절감과 납기의 단축 등의 목표를 실현한다. 둘째, 자사의 구매 시스템과 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.

본 논문에서 구현한 XML Hub 시스템을 활용한 e-Procurement는 HTTP 위에서 SOAP을 사용하므로 기업의 환경에 영향을 받지 않고 적용이 가능하다. 또한 XML Hub 시스템이 가지고 있는 애러처리 능력으로 전송의 실패를 방지할 수 있다. 본 논문에서는 e-Procurement의 일부만을 구현했지만 이를 더욱 확장한다면 e-Procurement가 가지고 있는 모든 장점을 수용하게 된다.

6. 결론

e비즈니스를 위해 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 것이 XML 이다. 데이터를 XML 로 표현하고 전송수단을 HTTP 환경의 SOAP 표준을 이용한다면 지금 까지 가지고 있던 여러 가지 문제를 한번에 해결 수 있다.

본 논문에서는 SOAP을 이용하여 XML Hub 시스템을 구현함으로써 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 처리해 주는 시스템이 된다. 이러한 XML Hub 시스템을 e-Procurement에 활용하였고, 이것은 e-Procurement 시스템의 장점과 XML Hub 시스템이 가지고 있는 장점을 모두 수용할 수 있는 시스템이 된다.

참고문헌

- [1] Apache SOAP Documentation: User's Guide [HTTP://xml.apache.org/soap/docs/index.html](http://xml.apache.org/soap/docs/index.html)
- [2] Bequet, Henry. Professional Java SOAP, Wrox, 2001
- [3] MicroSoft, "Biztalk Framework 1.0 Independent Document Specification" <http://www.biztalk.org>
- [4] OASIS - Technical Committees - ebXML Messaging Services, <http://www.oasis-open.org/committees/ebxml-msg/#documents>
- [5] RozettaNet, <http://www.rosettanel.org/>
- [6] Simple Object Access Protocol(SOAP), http://www.microsoft.com/korea/msdn/workshop/xml/general/SOAP_White_Paper.asp
- [7] Simple Object Access Protocol(SOAP) 1.1, <http://www.w3.org/TR/SOAP>
- [8] Subrahmanyam Alluamaraju, Professional Java E-Commerce, WORX, 2001
- [9] 김용수, 임종선, 주경준, 주경수, "공급사슬관리를 위한 XML 메세징 시스템 개발", 한국인터넷정보학회 춘계학술발표대회 논문집 제3권 1호, p141-144, 2002
- [10] 김용수, 주경준, 주경수, "SOAP 기반의 XML 메세징 전송 시스템 개발", 한국멀티미디어학회 춘계학술발표대회 논문집, 제5권 1호, p649-654, 2002
- [11] 김용수, 주경수, "SOAP 기반의 메시지 Broker 시스템 개발" 한국정보과학회 추계학술발표대회 논문집, Vol.30 No.1, 2002
- [12] 김윤명, XML을 위한 JAVA Programming, 가남사, 2002
- [13] 김채미, 전문가와 함께하는 XML Camp, 아이트Press, 2001
- [14] 김채미, 최학열, 글로벌 e비즈니스 리더를 위한 ebXML, 대청미디어, 2001
- [15] 신동균, 신동일, "XML/EDI 시스템의 설계 및 구현", 정보처리학회 논문지, 제 8-D권, 제 2호, 2001, pp.181-192.
- [16] 임종선, 주경수, "B2B 전자상거래를 위한 공급체인관리 시스템 설계 및 구현", 한국멀티미디어학회 춘계학술발표대회 논문집, 제5권 1호, p643-648, 2002
- [17] 임종선, 주경수, "효율적인 e-Procurement를 위한 XML 메세징 시스템 개발", 한국전자거래학회 논문집, 제7권 2호, 2002

저자 소개

김용수 (E-mail : admin@kimys.pe.kr)

2001년 순천향대학교 컴퓨터공학과 졸업

2003년 순천향대학교 전산학과 석사과정 졸업

2003년 (주)IT4web

연구분야 : XML, CBD, EJB

주경수 (E-mail : gsoojoo@asan.sch.ac.kr)

1980년 고려대 이과대학 수학과 졸업(학사)

1985년 고려대 일반대학원 전산학과 졸업(석사)

1993년 고려대 일반대학원 전산학과 졸업(박사)

1998년 University of North Carolina Visting Professor

1986년 ~ 현재 순천향대 정보기술공학부 교수

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