

SOAP을 기반으로 한 XML Hub 시스템 개발

김용수* 주경준**, 주경수*
순천향대학교 전산학과*, 골든폰드**

Developing XML Hub System based on SOAP

Kim Young-Soo*, Kyung-Joon Ju**, Joo Kyung-Soo*
Dept. of Computer Science, Soonchunhyang University
EBI Solution Inc**
E-mail : admin@kimys.pe.kr, kju@goldenpond.co.kr gsoojoo@sch.ac.kr

요 약

SOAP(Simple Object Access Protocol)은 분산 환경에서의 정보 교환에 사용되는 분산 컴퓨팅 프로토콜로 분산 시스템간에 메시지를 전달하는 방법과 원격 프로시저 호출/응답을 처리하는 방법 등이 정의되어 있다. 이러한 SOAP는 텍스트 기반 XML을 프로토콜로 사용하고 있기 때문에 하드웨어 플랫폼과 운영체제 및 프로그래밍 언어에 독립적으로 사용할 수 있다. 따라서 전자상거래 표준을 위한 ebXML에서도 메시지 전송을 위해 SOAP을 사용하고 있다.

본 논문에서는 SOAP을 기반으로 XML Hub 시스템을 개발하였다. 이에 따라 본 XML Hub 시스템을 기반으로 B2B 메시지 전송이 용이하게 되어, 보다 많은 비즈니스 파트너를 효과적으로 통합관리할 수 있을 것이다.

1. 서론

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 체제를 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[4,6].

본 연구에서는 이러한 SOAP을 기반으로 XML Hub 시스템을 구현한다. XML Hub는 서로 다른 거래 당사자들간에 XML 문서를 정확하게 주고 받을 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 거래 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.

2. 관련 기술

2.1 SOAP

SOAP은 분산 환경에서의 정보 교환에 사용되는 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 메시지인지는 정의하지 않는다. 반면 SOAP을 전송 프로토콜로 사용하는 ebXML과 같은 명세에는 메시지의 의미가 정의되어 있다[1].

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 상호운용성이 매우 높다. SOAP은 HTTP를 사용할 수 있기 때문에 기존의 웹 서버, 프록시 서버, 방화벽 같은 인프라를 그대로 이용할 수도 있다. SOAP은 HTTP 외에도 SMTP나 JMS와 같은 프로토콜과도 함께 사용할 수 있다.

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 1은 이에 사용되는 HTTP 포스트 방식을 이용한 SOAP 메시지의 한 예이다. 왼편의 SOAP 메시지 구조와 함께 메시지를 구성하는 요소(elements)들이 있다.

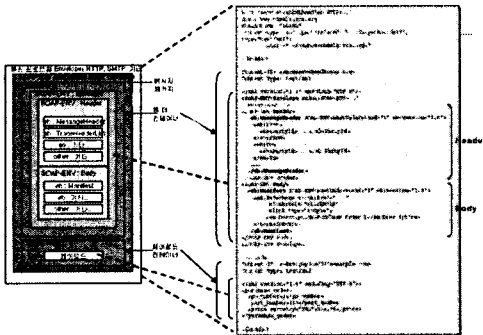


그림 1. SOAP 메시지 구조

SOAP 메시지는 웹-폼드 XML 문서로서 표현하며, 그림 1과 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[7].

Body 엘리먼트는 메시지의 본체가 되는 부분으로서 메시지의 내용을 담는다. Header 엘리먼트는 메시지에 대한 추가적인 정보를 기술하는 부분으로서 권한(authentication) 정보, 트랜잭션 관리 정보, 지불(payment) 관련 정보 등을 기술하는 데 사용한다.

2.3 메시지 전송 구조

SOAP 메시지 요청과 응답부분을 메시지 서비스 핸들러라고 하는 MSH가 담당한다. 그림 2는 XML Hub를 이용한 두 기업간 메시지 처리부분을 보여주고 있다[2,3,5,7].

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 재전송 한다.

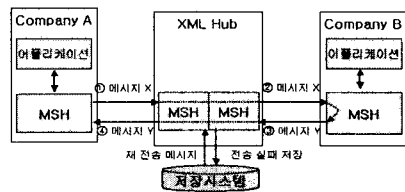


그림 2. 메시지 전송 구조

3. XML Hub 시스템

3.1 XML Hub 시스템 구조

전체 XML Hub 시스템의 구조는 그림 3과 같다. XML Hub 시스템은 송신, 허브시스템, 수신의 각각 3가지 모듈로 이루어져 있으며 각각의 동작 과정을 살펴본다.

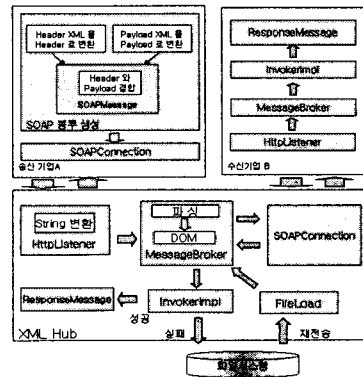


그림 3. 전체 XML Hub 시스템 흐름

3.1.1 송신

송신 기업에서는 사용자가 작성한 XML 문서를 SOAPHeader와 SOAPPayload를 이용하여 헤더와 페이로드를 작성한다. 이렇게 Header와 Payload를 작성한 후 SOAPMessageMake를 이용하여 Header와 Payload를 SOAP 봉투화 한다. 생성된 메시지를

SOAPTransfer 클래스를 이용하여 XML Hub 시스템에 전송한다.

3.1.2 XML Hub 시스템

송신자가 메시지를 만들어서 XML Hub 시스템에 보내게 되면, XML Hub 시스템 최종 수신 기업에 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 화일 시스템에 저장한다.

3.1.3 수신

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 가 있다.

3.2 주요 클래스

3.2.1 SOAPMessageMake

SOAPHeader와 SOAPPayload를 이용하여 작성한 것을 SOAPMessage에서 구성하게 된다. 즉, 두 개의 서로다른 Header 파일과 Payload 파일을 하나로 구성하는 기능을 가진다. 이곳에서 작성한 메시지는 SOAPTransfer 에 넘겨준다.

3.2.2 SOAPTransfer

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

3.2.3 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 MessageBroker에 넘겨준다.

3.2.4 MessageBroker

HUBListener에서 보내준 스트링을 serviceSync()에서 처리한다. serviceSync()는 스트링을 파싱해서

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

3.2.5 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.

3.2.6 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. 구현 환경 및 실행

XML Hub 시스템의 환경으로 운영체제는 Windows 2000 Server, 웹 서버로는 Jakarta Tomcat V3.2.3을 사용하였으며, SOAP API 로는 아파치 SOAP2.2, 와 JAXM을, XML 파서로 XMLParserV2를 이용하여 구축하였다. 다음 그림 4는 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 전달되고 XML Hub 시스템에 의해 최종 수신자에 다시 전송된다.

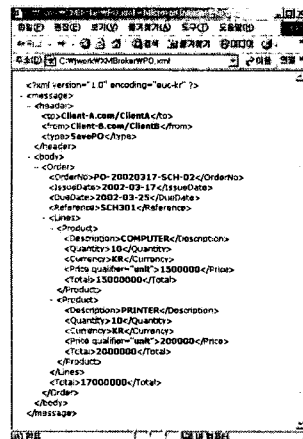


그림 4. sample XML 문서

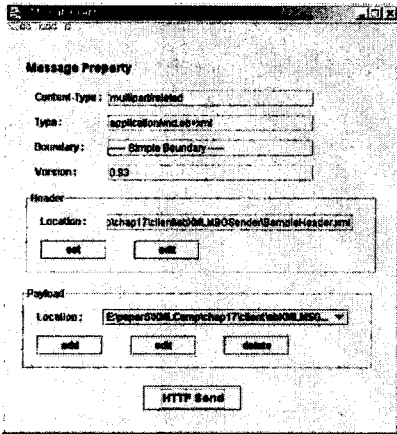


그림 5. 송신 GUI 화면

그림 5는 XML 문서를 Hub 시스템에 전송하는 GUI 프로그램이다. GUI는 JAVA의 Swing 을 이용하여 구현하였다.

그림 6은 수신자가 메시지를 받을 때 나타나는 화면이다. 메시지 전송이 시작되면 Servlet이 시작되고, 전송이 완료되면 Servlet이 끝나며, 전송 결과는 XML Hub 시스템에 전달되어 그림 7와 같은 전송 내용과 결과를 웹서버의 실행 창에 보여 주게된다.

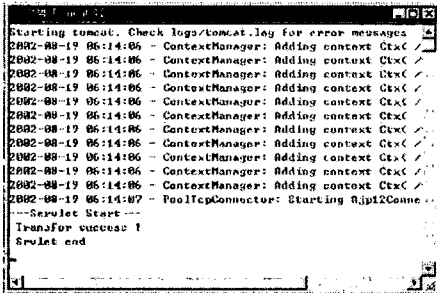


그림 6. 수신측 실행화면

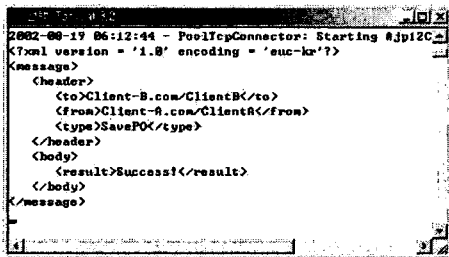


그림 7. XML Hub 시스템 실행화면

5. 결론 및 향후 연구방향

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 XML로 표현하고 전송수단을 HTTP 환경의 SOAP 표준을 이용한다면 지금 까지 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 함으로 전송시 발생하는 문제를 해결할 수 있다.

[참고문헌]

- [1] Simple Object Access Protocol(SOAP)1.1, <http://www.w3.org/TR/SOAP>
- [2] Simple Object Access Protocol(SOAP), http://www.microsoft.com/korea/msdn/workshop/xml/general/SOAP_White_Paper.asp
- [3] Apache SOAP Documentation: User's Guide <http://xml.apache.org/soap/docs/index.html>
- [4] 신동규, "XML/EDI 시스템의 설계 및 구현", 한국정보처리학회 논문지 8-D권 제 2호, 2001
- [5] 김채미, 전문가와 함께가는 XML Camp, 아이트 Press, 2001
- [6] 김채미, 최학열, 글로벌 e비즈니스 리더를 위한 ebXML, 대청미디어, 2001
- [7] 김윤명, XML을 위한 JAVA Programming, 가남사, 2002