

SOAP을 이용한 XML Hub 시스템 설계

김용수, 주경수,
순천향대학교 전자계산학과
e-mail:xml@.sch.ac.kr

Designing a XML Hub System using SOAP

Kim Yong-Soo, Joo Kyung-Soo
*Dept of Computer Science, Soonchunghang University

요 약

최근 인터넷 기술의 급속한 보급과 발전은 컴퓨팅 패러다임의 변화를 촉진시키고 있다. 이러한 컴퓨팅 패러다임 변화의 중심에는 인터넷을 기반으로 네트워크화 되는 e비즈니스가 등장하였다. e비즈니스 환경에서 기업들은 다양한 거래 파트너를 관리하게 된다. 이로 인해 보다 효율적인 관리가 필요하며 자주 발생하는 거래 파트너의 변화에 신속히 대처하기 위해 이를 관장하고 처리해주는 시스템이 필요하다. 또한 거래 파트너와의 메시지 교환에 있어서 메시지 전송에 대한 요청/응답을 담하고 응답 지연과 실패에 대한 처리가 필요하다. 따라서 본 논문에서는 SOAP(Simple Object Access Protocol)을 이용한 XML Hub 시스템을 설계하여 이를 해결하고자 한다.

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 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[4].

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 운용적이며 안전하고 일관성있는 방법으로 광범위한 e비즈니스가 일어날 수 있도록 개방된 XML 기반의 기반 구조를 제공하고 있다. 이러한 ebXML은 메시지를 교환을 위해 SOAP을 이용하고 있다.

본 연구에서는 이러한 SOAP을 기반으로 XML Hub 시스템을 설계하려고 한다. 이는 e비즈니스 환경에서 기업

들은 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. SOAP 메시지와 e비즈니스의 관계

SOAP 메시지 구조와 구문은 XML어휘를 갖는 특정 산업으로 하여금 SOAP 메시지는 필요조건들을 충족시킨다. 이러한 산업에 속하는 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 [1].

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 매핑

하는 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 접속가능한 세계 어디에서라도 비즈니스 데이터 교환을 위한 하나의 단일화된 프로토콜이다.

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

3. XML Hub 시스템

3.1 XML Hub 시스템 구조

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. 한편의 SOAP 메시지 구조와 함께 메시지를 구성하는 요소(elements)들이 있다.

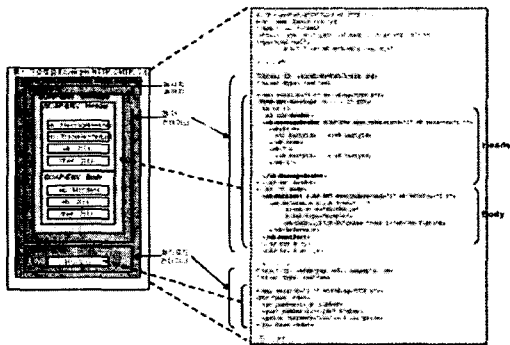


그림 1 SOAP 메시징 구조

SOAP 메시지 요청과 응답부분을 메시지 서비스 핸들러라고 하는 MSH가 담당한다. 그림 2는 XML Hub를 이용한 두 기업간 메시지 처리부분을 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 재전송 한다.

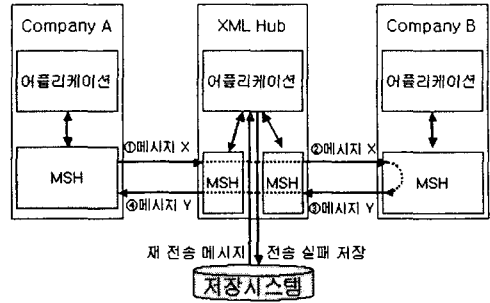


그림 2 XML Hub 구조 다이어그램

3.2 XML Hub 시스템 기능

XML Hub 시스템에서 서버 기능을 수행하며, 받은 메시지에 대하여 헤더처리, 암호화/복호화 및 보안, 예외/예외처리, 라우팅, 호출 5가지 기능을 수행한다[2][3].

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 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.

3.2.3 오류와 예외처리

오류와 예외처리는 XML Hub 시스템이 수행하는 중요한 기능 중 하나다. 기업이 수신한 메시지가 유효하지 않거나, 요청을 수행할 수 없는 경우에는 예외 메시지를 기업에 보내야만 한다. 그리고 XML Hub 시스템의 문제로 인해 서비스를 제공할 수 없는 경우에도 해당 메시지를 클라이언트에 보낸다.

3.2.4 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 할지를 결정한다.

3.2.5 호출

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

4. XML Hub 시스템 설계

4.1 Use Case 다이어그램

그림 3은 유스 케이스 다이어그램을 나타냈다. 사용자는 XML 문서를 작성하고, 작성된 문서를 토대로 SOAP 봉투 처리를 하고 이렇게 생성된 XML 문서를 XML Hub 시스템에 전송하여 어떠한 처리를 요구한다. XML 문서 처리를 하기 위해서는 XML 문서가 있어야만 가능하기 때문에, XML 처리 유스 케이스는 생성 유스케이스를 반드시 필요로 하는 관계인<<include>>로 표현된다.

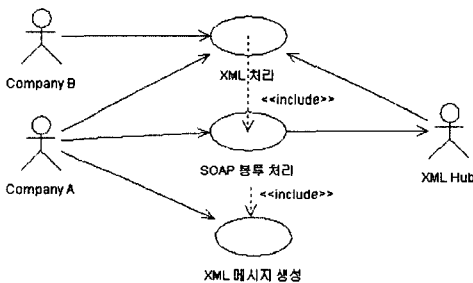


그림 3 Use Case 다이어그램

4.2 순차 다이어그램

그림 4는 기업A가 XML 문서를 생성하여 XML Hub 시스템에 전송하고, XML Hub 시스템은 다시 기업B에 XML 문서를 전송하여 처리 결과를 응답하는 순차 다이어그램을 나타낸 것이다. 기업A는 사용자 인터페이스를 통하여 요청 서비스를 포함하는 XML 문서를 작성하고 XMLMessage를 통하여

SOAP 봉투를 작성한다. 이렇게 작성된 문서를 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.

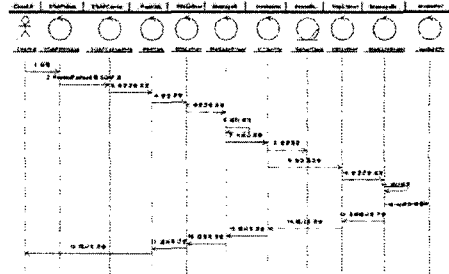


그림 4 순차 다이어그램

4.3 XML Hub 시스템

XML Hub 시스템의 처리과정은 그림 5과 같은 흐름을 갖는다. 송신 기업A에서는 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 HTTP 를 이용하여 전송한다. 이렇게 클라이언트가 메시지를 만들어서 XML Hub 시스템에 보내게 되면, XML Hub 시스템 다시 수신 기업에 전송하여 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[4][5].

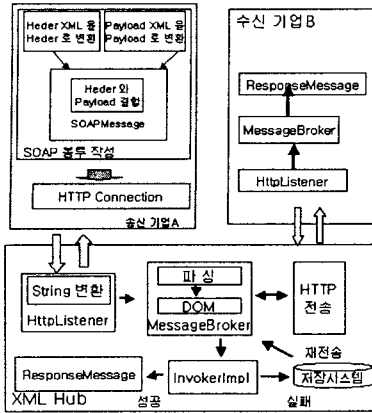


그림 5 XML Hub 시스템

4.3.1 SOAPMessage

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

4.3.2 SOAPConnection

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

4.3.2 HttpListener

HttpListener 은 클라이언트에서 POST 메소드로 넘어온 XML 메시지를 문자열로 받는 서버릿 클래스이다. HttpListener는 그림 5의 메시지 구조를 이용하여 작성한 XML 문서를 Stream 값으로 받은 다음, Reader로 변환한 후, 스트링으로 읽고, 그 코드를 MessageBroker에 넘겨준다.

4.3.3 MessageBroker

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

4.3.4 InvokerImpl

Invoker 인터페이스를 구현한 것인 InvokerImpl 클래스이다. 여기에서는 파일 시스템에 송장을 저장

하고, ResponseMessageMaker에서 응답 메시지를 만든다. ResponseMessage Manager는 요청이 성공적으로 이루어 졌을 때는 클라이언트에 보낼 '성공' 메시지나, 실패했을 경우의 '에러' 메시지를 XML 문서로 만든다.

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 설계함으로 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 대처하기 위해 이를 관리하고 처리해주는 시스템이 필요하다. 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. 향후 본 시스템을 구현하려고 한다.

참고문헌

- [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] Bequet, Henry. Professional Java SOAP, Wrox, 2001
- [4] 김채미, 최학열, 글로벌 e비즈니스 리더를 위한 ebXML, 대청미디어, 2001
- [5] 김채미, 전문가와 함께가는 XML Camp, 아이트Press, 2001