

## B2B를 위한 ebXML 메시징 서비스 엔진의 설계 및 구현

강태원<sup>o</sup> 강필구 박상현 채진석 신승호

인천대학교 컴퓨터공학과

{k83tw<sup>o</sup>, kpg2976, tank1862, jschae, shin0354}@incheon.ac.kr

### A Design and Implementation of ebXML Message Service Engine for B2B

Taewon Kang<sup>o</sup>, Pilgu Kang, SangHyun Park, Jinseok Chae, SeoungHo Shin

Dept. of Computer Science and Engineering, University of Incheon

#### 요 약

EDI는 기업과 기업간의 거래에서 좋은 방법이지만 중소기업이나 소기업에서는 EDI의 도입이 기술적으로나 비용적으로 어려운 게 사실이다. 이 경우 ebXML을 이용하여 EDI를 대신할 수 있다. ebXML을 이용하여 B2B환경을 구축할 경우 ebXML문서를 서로 주고 받는데 메시징 서비스 엔진이 필요하게 된다. 본 연구에서는 ebXML기반의 B2B환경에서 활용할 수 있는 메시징 서비스 엔진을 설계하고 구현하였다. 또한 실제 ebXML메시지를 전송할 때 발생하는 오류를 해결하기 로그를 기록하여 분석할 수 있도록 개발하였다.

#### 1. 서 론

기업과 기업간의 거래에서 좋은 방법의 하나는 바로 EDI(Electronic Data Interchange)를 이용하는 것이다. 실제로도 공공기관이나 대기업에서 EDI를 사용하여 많은 효과를 얻고 있다. 하지만 중소기업이나 소기업에서는 EDI의 도입이 기술적으로나 비용적으로 어려운 것이 사실이다. 또한 EDI를 이용한 기업 간 거래에 있어서도 융통성이나 확장성을 보장하지 못한다. 특히 한 기업의 전자시스템과 다른 기업의 전자시스템이 전자문서 교환을 하려고 한다면 둘 중 하나는 새로 만들거나 고쳐야 한다. 그 이유는 고정된 메시지 포맷 형식과 상이한 프로그래밍 언어, 이질적인 플랫폼 등을 들 수 있다[1].

EDI의 한계를 극복할 수 있는 대안이 바로 ebXML이다. ebXML은 플랫폼에 독립적이며, XML 기반의 SOAP(Simple Object Access Protocol)메시지를 담고 있기 때문에 방화벽에도 친근하다. 또한 특정 기술 혹은 프로그래밍 언어에 의존하지 않기 때문에 ebXML 명세만 따른다면 메시지 교환이 가능하다. 이러한 장점을 바탕으로 실제 B2B환경 구축시 더 안전하고 저렴하게 시스템 구축이 가능하다[1][2].

ebXML명세를 따르고 이를 바탕으로 B2B환경을 구축했을 경우 서로간의 메시지를 명확하게 주고 받게 할 수 있는 메시징 서비스 엔진이 필요하게 된다. 이 메시징 서비스 엔진이 바로 ebXML 메시징 엔진이다. ebXML 메시징 서비스는 ebXML을 바탕으로한 기업 간 메시지 교환의 표준 프레임워크를 제공한다.

그러나 아직까지 ebXML 기반의 B2B환경에 적용 가능

한 ebXML 메시징 서비스 엔진의 개발은 미흡한 단계이다. 또한 SOAP메시지를 담고있는 ebXML 메시징 서비스 개발 역시 필요한게 사실이다.

본 논문은 B2B를 위한 ebXML에 적용할 수 있는 메시징 서비스 엔진을 설계, 개발하는 것이며 메시지 서비스 V2.0에 맞는 서비스 엔진을 개발할 것이다. 또한 메시지 수신, 발신시에 발생하는 오류를 해결하기 위하여 각각의 오류를 따로 기입하고 수신, 발신과정에 따른 오류처리를 다르게 처리하도록 구성하였다.

#### 2. SOAP의 정의 및 구조

SOAP은 인터넷을 통해서 XML 기반의 정보를 교환하는 메소드이다. SOAP프로토콜은 최소한의 기능만을 삽입한 간결한 프로토콜이다. SOAP 프로토콜이 정의하는 프레임워크는 확장성이 있고 모듈로 나뉘질 수 있으며, 메시지 전송 시스템을 강력하고 유연하게 만들어 준다. SOAP 프로토콜 명세가 HTTP에 묶여있지 않고 한 지점에서 다른 지점으로 SOAP Envelope들을 전송할 수 있는 모든 전송 프로토콜, 예를 들어 SMTP, FTP등을 사용할 수 있다[3].

SOAP 메시지는 SOAP Header블록과 SOAP Body블록으로 구성되어 있고 Header와 Body는 다시 SOAP Envelope안에 포함되는 구조를 지니게 된다. (그림 1 참고) 각각의 요소는 다음과 같은 기능을 수행한다.

Envelope 요소는 SOAP의 Envelope 네임스페이스 <http://schemas.xmlsoap.org/soap/envelope/>에 속하는 것이다. 이 요소는 XML문서의 루트요소로 TM이며 보통 네임스페이스 선언들을 포함하는데 쓰인다.

Header 요소는 SOAP 메시지에서 호출하고 있는 프로시저에 의해 정의되는 내용 이외의 부가적인 정보를 넣어 줄 필요가 있을 때 사용된다. SOAP은 Header 안의

본 연구는 산업자원부 지정 인천대학교 동북아전자물류연구센터의 지원에 의한 것임.

구성 요소에 포함되는 mustUnderstand 속성을 허용한다. SOAP 메시지를 받는 웹서비스는 mustUnderstand 속성으로 표시된 구성 요소를 처리하거나 표시되지 않은 SOAP 요청은 처리하면 안된다[3].

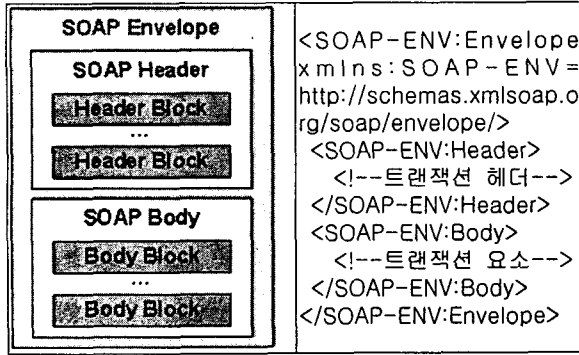


그림 1. SOAP 메시지 구조와 예제

Body 요소는 SOAP 메시지 안에서 표현되고, 메시지의 핵심 내용을 담고 있다. 보내지는 메시지의 형태는 Body 요소 안에 무엇이 있는지 정의한다. SOAP 메소드를 호출하기 위해서 Body는 XML 포맷으로 된 각각의 파라미터와 메소드의 이름을 가진 직렬화된 메소드 호출 요청을 포함하고 있다.

### 3. 시스템 설계

본 절에서는 일반적인 ebXML 메시지 서비스의 내부구조를 설명하고 실제 구현할 시스템 내부구조와 메시지 수발신 구조를 설명한다.

#### 3.1 일반적인 메시지 서비스의 구성

ebXML 메시지 서비스는 개념적으로 (1) 추상적인 서비스 인터페이스, (2) 메시지 서비스 처리기에 의해 제공되는 기능, (3) 기본적인 전송 서비스와 연계하기 위한 맵핑으로 나뉜다.

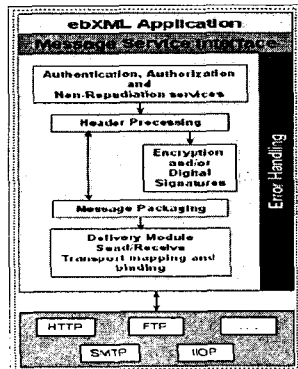


그림 2. ebXML 메시징 서비스 모델

(그림2) 는 하나의 구현 가능한 ebXML 메시지 서비스 구조의 내부를 기능적 모듈들을 기반으로 하여 논리적으로 정리해 놓은 것이다[4][5].

#### 3.2 메시지 서비스 엔진 설계

메시징 엔진은 크게 두 부분으로 나누어 볼 수 있다. 메시지를 라우팅 처리하는 서버 부분과 메시지를 송/수신하는 클라이언트 부분이다. ebXML 기반 애플리케이션 아키텍처는 구성하기에 따라 여러 가지 구성도를 가질 수 있지만 활용성이 높은 C/S 타입의 메시징 엔진을 구성하였다. 이를 바탕으로 다른 애플리케이션 구조로의 확장도 가능할 것이다.

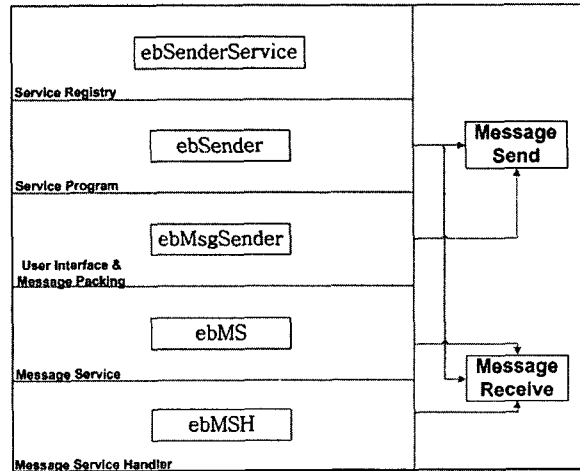


그림 3. 프로그램 구조

설계된 메시지 서비스 엔진은 총 5개의 부분으로 구분될 수 있다. 최상단부의 서비스 등록 부분은 메시지를 보내는 역할을 하는 Service를 Windows Service에 등록하는 Service Installer의 역할을 수행한다. 서비스 프로그램은 Windows 서비스에 등록되는 서비스 프로그램으로 DBMS에 발송용으로 저장한 메시지를 추출하여 목적지 MSH에 전송하는 역할을 한다. 중간의 ebMsgSender는 UI 및 ebXML 메시지와 첨부파일 처리등을 담당한다. 즉 사용자가 각각의 메시지와 첨부파일을 선택할 경우 HTTP를 통해서 보낼 수 있는 메시지로 만들어 DBMS에 저장한다. ebMS는 HTTP를 통해 들어오는 ebXML 메시지를 수신하는 역할을 한다. 최 하단부는 MSH(Message Service Handler)로서 메시지 서비스의 실제 구현부 역할을 수행하게 된다.

#### 3.3 메시지 수신, 발신 과정

전체 시스템 진행 과정은 전체적으로 총 2가지의 경우로 나누어서 볼 수 있다. 첫번째 메시지 수신의 과정은 두 가지의 경우의 수가 있다. 하나는 ebMS를 통한 HTTP Request이며 다른 하나는 ebSender를 통하여 메시지가 발송된 후 응답을 받은 경우이다. 즉 각각의

경우에 다음과 같은 형식으로 진행되게 된다. ebMS를 통하여 수신되는 메시지의 형식은 Status Request, Ping, 그리고 일반적인 메시지이다. 이에 응답하는 형식은 차례대로 StatusResponse, Pong, Ack이다.

ebSender를 통하여 수신되는 메시지의 형식은 Status Response, Pong, Ack이다. 즉, ebSender는 메시지를 발송하는 역할을 하기 때문에 그에 대한 응답을 받아 Database에 저장하는 것이다.

두번째 메시지 송신의 과정은 DBMS에 WillSend Messages 테이블에 메시지가 저장되면 Service로 등록된 ebSender 애플리케이션이 메시지 전송하는 과정을 수행한다. 메시지 발신 구조가 DBMS의 테이블에 저장되면, ebSender라는 Service에 의해서 전송되는 이유는 본 MSH를 이용하여 ebXML 솔루션을 구축할 때 기존의 시스템과 연동하기 수월하게 하기 위해서이다. 즉 메시지를 전송하는 애플리케이션을 작성할 때 본 MSH의 구조나 언어에 상관없이 DBMS의 테이블 명세만 숙지하여 작성하면 되기 때문이다. 또한 ebSender는 테이블에 쌓인 메시지를 순서대로 전송한 후 적절한 응답이 오지 않으면 최대 3회 재시도한다. 실패한 경우는 Sent Messages Table에 RetryCount가 0, isDone 값이 f인 저장한다.

#### 4. 시스템 구현

본 논문의 시스템은 Microsoft Windows 2000 Server 상에서 .NET Framework 1.0를 바탕으로 개발하였다. 구현언어는 Microsoft VisualStudio.NET 의 C#과 ASP.NET 이 사용되었고 데이터베이스는 Microsoft SQL Server 2000 Enterprise을 사용하였다. 그리고 실제 구현을 함에 있어 Microsoft사에서 제공하는 SOAP Toolkit을 사용하여 메시징 엔진을 구현해 보았다.

##### 4.1 구현 프로그램

실제 메시지입력은 ebMsgSender를 통하여 이루어 지고 변화되어 DB에 저장된 메시지는 ebSender를 통하여 보내지게 된다.(그림 4 참고)

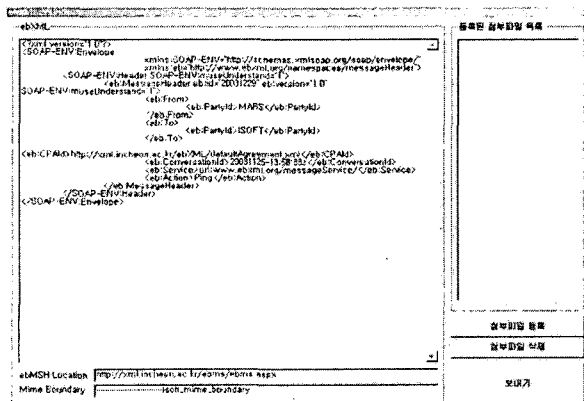


그림 4. 메시지 전송 프로그램(ebMsgSender)

#### 4.2 에러 처리

에러처리 과정은 전체적으로 ebMS(메시지수신)에서 발생하는 오류와 ebSender(메시지 송신)에서 발생하는 에러의 두가지 경우로 나누어 생각할 수 있다.

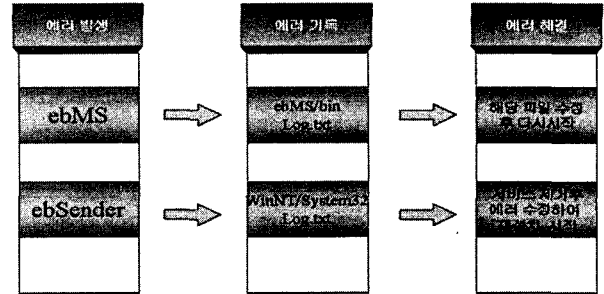


그림 5. 오류 처리 과정

#### 5. 결론 및 향후 과제

본 논문에서는 B2B에 적용할 수 있는 ebXML메시징 엔진을 작성해 보았다. ebXML 메시징 엔진을 설계, 구현해 봄으로써 메시징 엔진이 가져야 할 기능을 정리해 보고 한 메시징 엔진의 한 모델을 제시했으며 DB를 이용하여 기존 시스템과 연동할 수 있는 기능을 제안하였다. 이 ebXML 메시징 서비스는 기업 전반에 걸쳐 충분히 가치있는 것이다. 이것은 기업만이 아닌 기업 대 공공기관, 기업 대 개인 등의 어떠한 형태의 거래나 전자문서 교환 등등의 모델에도 적용해 볼 수 있을 것이다.

향후 연구로는 실제 ebXML을 도입하려는 기업이나 기관에 알맞은 메시징 엔진을 설계 제작하고 좀더 명확한 송수신 로직과 범용성이 있는 형태의 엔진으로 발전 시켜야 할 것이다. 또 기존 에러처리에 관련하여 사용자가 에러 사항을 명확히 파악하고 수정할 수 있도록 연구해야 할 것이다.

#### 6. 참고문헌

- [1] 김영일, 이용준, 황재각, "ebXML TRP 2.0을 위한 MSH," 한국정보과학회 논문지C, Vol. 9, No. 2, pp.205-0212, 2003.
- [2] S.Patil and E.Newcomer, "ebXML and Web services," *IEEE Internet Computing*, 2003.
- [3] Wrox Author Team, *Professional ebXML Foundations*, WROX Press, 2001.
- [4] "Message Service Specification v 2.0," [Online] <http://www.ebxml.org>, 2002.
- [5] "ebXML Technical Architecture Specification v1.0.4," [Online] <http://www.ebxml.org>, 2001.
- [6] Mark Berbeck, et al., *Professional XML*, 정보문화사, 2002.