

효율적인 ebXML 메시지 서비스 인터페이스

정문영*, 오수영, 조현성, 조현규
한국전자통신연구원

e-mail : {mchung, runtou, hscho, hkcho}@etri.re.kr

Efficient Message Service Interface for ebXML Framework

Moonyoung Chung*, Sooyoung Oh, Hyeonsung Cho, Hyun-kyu Cho
Electronics and Telecommunications Research Institute

요 약

ebXML 은 기업간의 전자상거래를 위한 글로벌 표준으로 비즈니스 프로세스 및 문서, 메시지 서비스, 기업의 공용 등록소 및 저장소, 거래 약정 등에 관련된 세부 표준들로 구성되어 있다. 기업간에 협의된 비즈니스 프로세스에 따라 상대 기업과 문서를 주고 받고자 할 때에는, 비즈니스 응용 프로그램은 전송하고자 하는 문서와 메시지를 생성하고 전송하는데 필요한 파라미터들을 메시지 전송 모듈에 전달해야 한다.

본 논문에서는, 비즈니스 응용 프로그램과 메시지 전송 모듈 사이에 메시지 서비스 인터페이스를 제공할 때의 문제점을 정의하고, 이에 대한 해결책으로 효율적인 메시지 서비스 인터페이스인 메시지 채널을 제안한다. 더불어, 메시지 채널을 생성하는 방법을 제안한다. 메시지 채널을 사용함으로써 비즈니스 응용 프로그램은 메시지를 전송하는데 필요한 파라미터들을 프로그램 내에 하드 코딩하지 않고 효율적으로 전달할 수 있다.

1. 서론

ebXML 은 UN/CEFACT 와 OASIS 가 전자상거래 시장의 단일화를 목표로 제안한 XML 기반의 기술적 프레임워크이다. ebXML 은 기업간의 전자상거래를 위한 표준으로 등장하였으며, 이를 뒷받침하기 위해서 연구 및 개발이 계속되고 있다. ebXML 표준은 비즈니스 프로세스 및 문서, 메시지, 레지스트리, 거래 약정 등에 관련된 기업간 전자상거래를 하기위한 세부 표준들로 구성되어 있다.

ebXML 기반의 전자상거래는 비즈니스 프로세스에 따른 비즈니스 문서의 교환으로 이루어진다. 따라서, ebXML 표준을 지원하기 위해서는 표준에 따른 메시지 교환이 가장 기본적인 요소이며, 비즈니스 프로세스를 구현한 비즈니스 응용 프로그램에서 비즈니스 문서를 메시지로 전송하기 위해서는 메시지 전송 모듈과의 인터페이스가 필요하다.

일반적으로 비즈니스 응용 프로그램이 ebXML 메시지를 전송할 때에는, 비즈니스 프로세스의 각 비

즈니스 트랜잭션에서 주고 받는 비즈니스 문서 이외에도 메시지의 헤더 부분에 들어갈 정보나 메시지 전송 방법 등과 같은 CPA 에 기술된 정보들을 메시지 전송 모듈에 전달해야 한다. 그러나, CPA 의 정보는 관련된 비즈니스 프로세스 안에 있는 모든 비즈니스 트랜잭션에 대한 정보를 가지고 있으므로, 비즈니스 응용 프로그램을 작성하는 개발자는 실행하고자 하는 비즈니스 트랜잭션에 해당하는 정보만을 선택하여 메시지를 생성해야 한다. 다시 말하면, CPA 에 메시지를 전송하는 데 필요한 정보가 모두 들어있지만, 이 중에서 필요한 정보만을 선택하는 과정이 필요하고, 이 과정은 자동화될 수 없으며 응용 프로그램 개발자들이 프로그램 내에서 처리해야 한다. 또한, 이 정보를 비즈니스 응용 프로그램 내에 하드 코딩해야 하므로 CPA 의 내용이 변경되었을 경우 응용 프로그램도 수정해야 하는 문제점이 있다.

따라서, 본 논문에서는 비즈니스 프로세스 내의 비즈니스 트랜잭션과 CPA 내의 메시지를 생성하는데

필요한 정보를 매핑하는 방법을 제시함으로써, 비즈니스 응용 프로그램이나 메시지 전송 모듈, 비즈니스 서버 모듈에게 효율적인 인터페이스를 제공한다.

본 논문의 구성은 다음과 같다. 2 장에서는 ebXML의 구성 요소인 비즈니스 프로세스, CPP 와 CPA, 메시지 서비스에 대해서 간략히 살펴보고, 3 장에서는 ebXML 메시지 서비스 인터페이스에 대한 개념과 메시지 서비스 인터페이스를 구현하는데 어려운 점을 소개한다. 그리고 4 장에서는 본 논문에서 제안하는 효율적인 메시지 서비스 인터페이스인 메시지 채널에 대해서 자세히 살펴보고, 마지막으로 5 장에서 결론을 맺는다.

2. 관련 연구

2.1 ebXML 비즈니스 프로세스

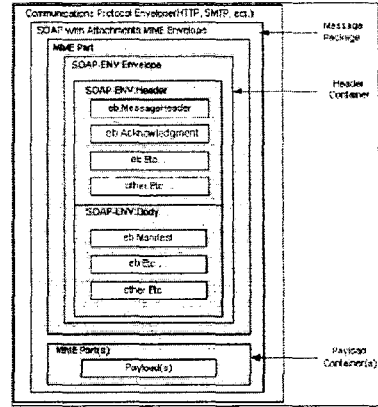
ebXML 비즈니스 프로세스 스펙은 기업간 거래에서 실제 비즈니스 시나리오와, 비즈니스 시나리오 내에서 각 기업이 어떤 역할을 수행하고 다른 기업과 어떤 관계를 맺는지 등에 관한 정보를 기술한다. 이러한 기업간의 비즈니스 프로세스는 비즈니스 트랜잭션 (business transaction) 들의 흐름으로 구성되며, 다시 각 비즈니스 트랜잭션은 비즈니스 문서의 교환으로 구성된다. 즉, ebXML 프레임웍에서 비즈니스 프로세스는 기업간의 비즈니스 문서의 교환으로 이루어진다.

2.2 ebXML Collaboration Protocol Profile/Agreements (CPP/CPA)

ebXML CPP(Collaboration Protocol Profile)는 기업의 비즈니스 프로파일 및 기업이 제공하는 기술적인 협력 방법에 관한 표준으로 그 기업이 지원하는 메시지 교환 방법, 비즈니스 프로세스, 보안 관련 사항, 다른 기업이 이 기업과 협력하는 방법 등을 포함하는 XML 문서이다. CPA (Collaboration Protocol Agreement)는 기업간의 비즈니스 협약에 관한 표준으로 기업간의 메시지를 교환하는 방법, 기업간의 비즈니스 프로세스 등을 포함하는 XML 문서이다. CPA 는 비즈니스에 참여하는 두 기업의 CPP 로부터 생성되며 거래에 참여하는 두 기업의 협상을 통하여 최종적으로 결정된다.

ebXML 비즈니스 프로세스에 따라서 기업간의 거래를 하기 위해서는 비즈니스 문서의 실질적인 교환이 이루어져야 하는데, 표준 프레임웍 내에서 어느 기업간에도 이러한 교환이 이루어질 수 있도록 하기 위해서는 메시지의 교환 포맷이 같아야 한다. 그러나, 모든 기업의 메시지 포맷이 같더라도, 메시지를 인코딩 하는 방법이나, 디지털 서명을 붙이는 방법 등 기술적인 구현까지 모두 동일해야 하는 것은 아니다. 따라서, 각 기업의 기술적인 구현 요소들을 CPP 에 기술하고, 두 기업간의 협의된 요소들은 CPA 에 기술하여, 모든 기업이 일관된 구현을 이용하지 않더라도 메시지를 주고 받는 것이 가능하다.

2.3 ebXML 메시지 서비스

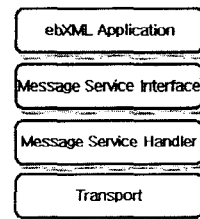


<그림 1> ebXML 메시지

앞에서도 언급했듯이, 어느 기업간에도 메시지 교환을 가능하게 하기 위해서는 메시지 포맷을 통일해야 한다. 따라서 ebXML 메시지 서비스 스펙에서는 SOAP (Simple Object Access Protocol)을 기반으로 하여 ebXML 프레임웍에 필요한 엘리먼트들을 확장한 ebXML 메시지 서비스에 관해 기술한다. 즉, ebXML 메시지 서비스는 ebXML 프레임웍에서 기업간에 서로 비즈니스 문서를 교환할 때, 전송되는 메시지의 패키징, 트랜스포트 등에 대한 표준을 정의한 것으로 아래 <그림 1>와 같이, MessageHeader, Acknowledgment, SyncReply, StatusRequest, StatusResponse, Manifest, AckRequested, Signature 등의 엘리먼트들로 이루어진다.

3. ebXML 메시지 서비스 인터페이스

아래 <그림 2>와 같이 ebXML 비즈니스 응용 프로그램과 메시지를 전송/수신하는 메시지 전송 모듈 사이에는 메시지 서비스 인터페이스가 필요하다.



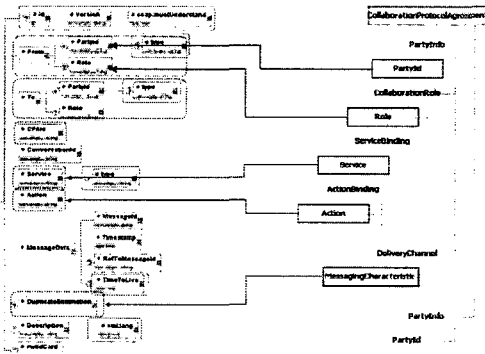
<그림 2> ebXML 메시징 구조

메시지 서비스 인터페이스는 ebXML 비즈니스 응용 프로그램이 문서를 상대 기업에 전송하고자 할 때, 실제 메시지 전송하는 부분인 메시지 전송 모듈에 파라미터를 전달하는 방법을 제공해야 한다. 비즈니스 문서 자체를 전달할 뿐 아니라, 비즈니스 문서를 패키징 하는 방법, 인코딩 하는 방법, 암호화하는 방법 등과 같은 기술적인 부분을 전달해야 한다. 이러한 내용은

두 기업간의 CPA 에 기술되어 있으므로, CPA 에 기술된 내용을 참조하여 전달하게 된다.

정리하면, 기업간의 비즈니스가 이루어질 때, 그 사이에는 비즈니스 프로세스와 그 비즈니스 프로세스에 맞는 CPA 문서가 존재하게 된다. 비즈니스 프로세스 내의 비즈니스 트랜잭션은 비즈니스 문서의 교환으로 이루어지는데, 각 문서의 교환마다 필요한 기술적인 정보는 CPA 문서에 기술되어 있다.

따라서, 비즈니스 응용 프로그램에서 비즈니스 문서를 전송하고자 할 때, 문서와 함께 CPA 를 파라미터로 주면 메시지 전송 모듈가 비즈니스 문서와 CPA 를 참조하여 ebXML 메시지를 구성하여 전송하는 역할을 수행할 수 있다. 그러나, 비즈니스 프로세스 내의 모든 트랜잭션에 관한 기술적인 정보가 하나의 CPA 에 들어있고, 하나의 항목은 여러 개의 선택사항들로 이루어져 있으므로 그 중에서 한가지를 선택하여 사용해야 하는데, 메시지 전송 모듈에서 이를 구분하여 메시지를 생성할 방법이 없다. 따라서, 비즈니스 응용 프로그램 내에서 수행하려는 비즈니스 트랜잭션에 해당하는 정보를 선택하여 ebXML 메시지를 생성해 주어야만 한다. 즉, 비즈니스 응용 프로그램 내에서 직접 CPA 에서 적당한 정보를 선택하여 메시지를 생성하여야 한다.



<그림 3> MessageHeader 와 CPA 사이의 매핑

예를 들면, ebXML 메시지의 MessageHeader 엘리먼트와 CPA 정보 사이의 매핑은 위의 <그림 3>와 같다. 이때 MessageHeader 내의 Action 엘리먼트와 CPA 내의 Action 엘리먼트를 매핑시켜야 하는데, CPA 안에는 여러 개의 Action context 가 존재하므로 이 중에서 한 메시지에 해당하는 Action 을 선택하여야 한다. 이러한 정보의 매핑은 MessageHeader 뿐 아니라, ebXML 메시지의 다른 구성 엘리먼트들에 대해서도 모두 이루어져야 한다. 이러한 매핑 과정은 비즈니스 응용 프로그램을 작성하는 데에 너무 많은 부담을 줄 뿐만 아니라, 이를 프로그램 내에서 하드 코딩해야 하는 단점이 있다.

4. 메시지 채널 생성

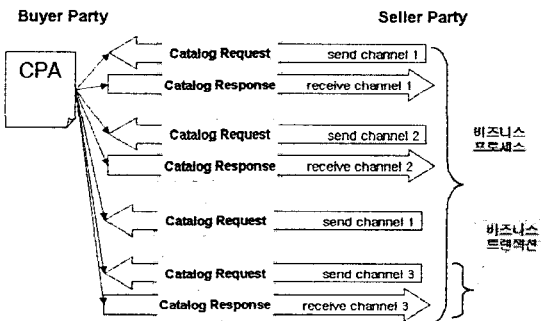
이 장에서는 본 논문에서 제안하는 비즈니스 응용 프

로그램과 ebXML 메시지 사이에 인터페이스를 효율적으로 하는 방법을 살펴본다.

4.1 메시지 채널 개념

앞에서 살펴본 바와 같이, 두 기업간의 거래에는 비즈니스 프로세스에 따른 CPA 문서가 존재하며, 이 CPA 문서 안에는 비즈니스 프로세스 내의 모든 비즈니스 트랜잭션에서 문서를 주고 받을 때 필요한 기술적인 정보들이 들어있다. 따라서, CPA 의 정보에서 각 트랜잭션에서 필요로 하는 정보, 즉 하나의 메시지를 주고 받을 때에 필요한 정보만을 선택하여 각 트랜잭션에 매핑시켜 이를 프로파일로 저장해 두면, 비즈니스 응용 프로그램에서는 메시지를 만들 때 필요한 파라미터들을 프로그램 내에 하드 코딩하지 않고 매핑시켜 둔 정보를 읽어서 메시지를 보낼 수 있다.

개념은 <그림 4>와 같다. 각 트랜잭션은 요구(request)와 응답(response)으로 이루어진다. 요구와 응답은 각각 메시지를 전송하고 수신하는 것이다. 따라서, 각 요구와 응답에 해당하는 정보들을 CPA 에서 선택하여 하나의 프로파일로 만들어 둘 수 있는데, 본문에서는 이를 **메시지 채널**이라 지칭한다. 즉, 각 트랜잭션마다, 메시지를 보내는 데 필요한 보내는 기업의 비즈니스 프로파일과 받는 기업의 프로파일, 문서 인코딩 방법, 암호화 방법, 트랜스포트 프로토콜 등과 같은 정보를 CPA 로부터 선택하여 보내는 메시지 채널과 받는 메시지 채널을 생성한다. 이 메시지 채널은 CPA 내부에 있는 엘리먼트인 DeliveryChannel 과는 구별되며, 메시지 채널은 DeliveryChannel 과 Service, Action 을 포함하는 보다 포괄적인 개념이다. 비즈니스 응용 프로그램에서는 보내고자 하는 비즈니스 문서와 이 메시지 채널을 메시지 전송 모듈에 전달함으로써 문서 전송을 가능하게 한다. 이 경우에는, 메시지 채널 안의 내용이 변경되더라도 비즈니스 응용 프로그램을 다시 작성하거나 변경할 필요가 없으므로 효율적인 메시지 서비스 인터페이스가 된다.



<그림 4> 메시지 채널을 이용한 메시지 전송

이렇게 생성된 메시지 채널은 비즈니스 프로세스의 각 트랜잭션에 매핑되어 저장되므로, 비즈니스 프로세스를 처리하는 서버나 메시지 전송 모듈에서도 사용

될 수 있다.

4.2 메시지 채널 생성 방법

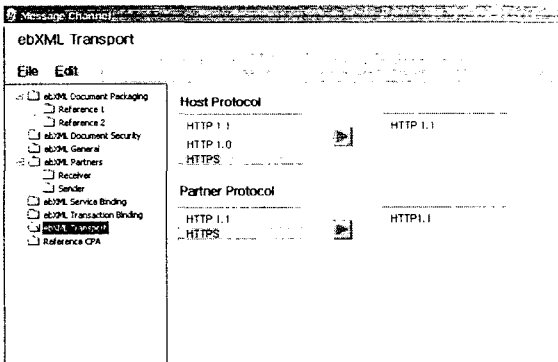
메시지 채널은 CPA 문서 안의 내용을 기반으로 하여 각 트랜잭션에서 필요한 정보들로 구성된다. 이 때, 어떤 정보를 어떻게 구성하는가는 메시지 채널을 효과적으로 이용하는 데 중요하다. 메시지 채널에서 필요한 정보는 다음과 같다.

- 1) 수신자, 송신자
- 2) 서비스 바인딩
- 3) 트랜스포트 프로토콜
- 4) 첨부할 비즈니스 문서의 패키징 방법 및 보안 방법
- 5) 메시지 전체의 보안 방법
- 6) CPA 문서에 대한 참조

이러한 정보를 갖는 메시지 채널도 CPA 문서와 같이 XML 파일로 구성되며, 이 채널을 관리하는 모듈을 필요로 한다.

CPA로부터 효과적으로 메시지 채널을 생성하려면 메시지 채널 생성 도구 프로그램이 필요하다. 메시지 채널 생성 도구 프로그램의 사용자 인터페이스는 CPA 문서를 읽어서 각 비즈니스 트랜잭션에 필요한 파라미터를 선택함으로써 쉽게 생성될 수 있다.

메시지 채널 생성 도구의 사용자 인터페이스는 아래 <그림 5>와 같다. 프로그램에서 CPA 를 읽어서 파라미터들을 선택하여 메시지 채널을 생성할 수 있다.



<그림 5> 메시지 채널 생성 툴

이렇게 생성된 채널은 비즈니스 트랜잭션에 매핑되어 있어서 비즈니스 응용 프로그램에서 비즈니스 문서를 전송할 때에 채널만을 선택하여 메시지 전송 모듈에 전달하여 주면 메시지 전송 모듈은 비즈니스 문서와 메시지 채널을 이용하여 ebXML 메시지를 생성하여 전송하게 된다.

5. 결론

기업간 전자상거래에서 ebXML 메시지 서비스 표준에 따른 메시지를 교환할 때, 메시지 전송 모듈은 협

약 된 CPA 에 따라 메시지를 생성하고 전송하게 된다. 그러나, CPA 문서는 참조하는 비즈니스 프로세스 내의 모든 트랜잭션에 관련된 정보를 포함하므로 어떤 트랜잭션과 관련된 정보를 비즈니스 응용 프로그램에서 선택해 주어야 한다. 이를 비즈니스 응용 프로그램 내에서 하려면, 개발자에게 부담이 주어질 뿐 아니라 CPA 가 변경될 경우 프로그램을 수정해야 하는 단점이 있다.

따라서, 본 논문에서는, 비즈니스 프로세스 내의 비즈니스 트랜잭션과 메시지 생성에 필요한 정보를 매핑하는 메시지 채널 개념을 도입하고 메시지 채널을 이용하여 메시지 서비스 인터페이스를 효율적으로 설계하였다. 메시지 채널을 이용함으로써, 비즈니스 응용 프로그램에서는 메시지 채널만을 참조함으로써 쉽게 메시지 서비스를 이용할 수 있으며, 메시지 채널 안의 내용이 변경되더라도 비즈니스 응용 프로그램을 다시 작성하거나 변경할 필요가 없으므로 효율적인 메시지 서비스 인터페이스가 된다.

참고문헌

- [1] ebXML Technical Architecture Project Team, *ebXML Technical Architecture Specification*, v1.0.4, published 16 Feb., 2001
- [2] ebXML Trading Partners Team, *Collaboration-Protocol Profile and Agreement Specification*, Version 2.0, published May 5, 2002
- [3] ebXML Transport, Routing, and Packaging Project Team, *Message Service Specification ebXML Transport, Routing & Packing*, Version 2.0 rev C, published February 21, 2001
- [4] *ebXML Proof-Of-Concept*, Technical Planning Document for End-to-End Demonstration in Vienna version, Working Draft, April 26, 2001
- [5] ebXML Requirements Team, *ebXML Requirements Specification*, Version 1.06, May 8, 2001
- [6] ebXML Business Project Team, *ebXML Business Process Specification Schema*, Version 1.05 (Candidate for version 2.0), July 15, 2002
- [7] Simon S.Y. Shim, Vishnu S. Pendyala, Meera Sundaram, and Jerry Z. Gao, *Business-to-Business E-Commerce Frameworks*, IEEE Computer, 2000
- [8] ebXML Registry Project Team, *ebXML Registry Services v1.0*, May 10, 2001