

ebXML 메시징 서비스 기술 동향 분석

An Analysis of ebXML Messaging Service Technology

김영일(Y.I. Kim)

u-Logistics연구팀 연구원

황재각(J.G. Hwang)

u-Logistics연구팀 책임연구원

이용준(Y.J. Lee)

u-Logistics연구팀 책임연구원, 팀장

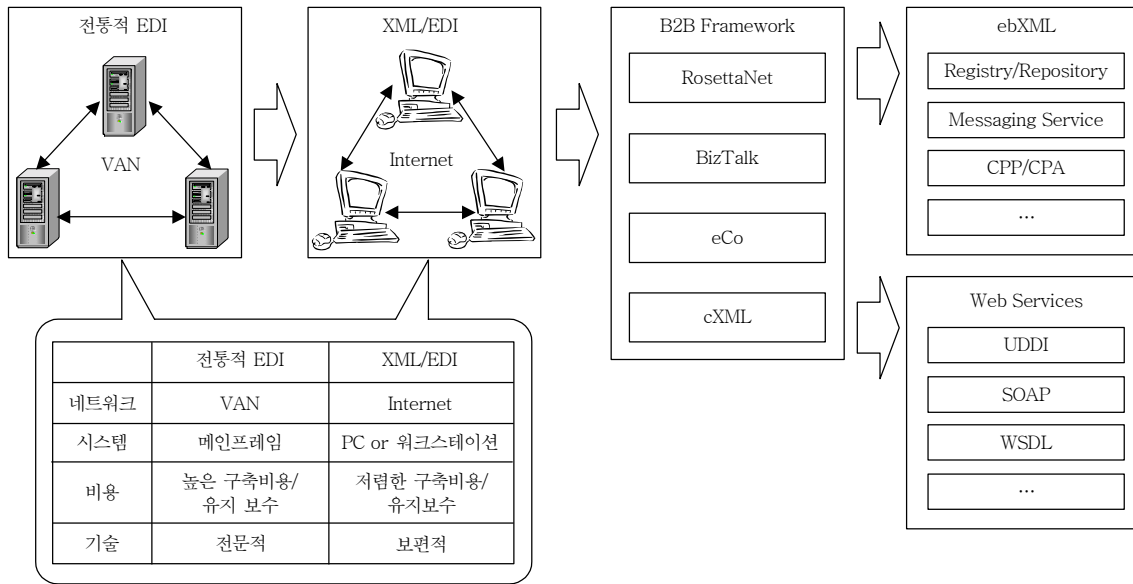
최근 인터넷을 통한 B2B 전자상거래가 확산됨에 따라 다양한 형태의 인터넷 비즈니스를 지원하기 위한 기술들이 나타나고 있다. 이에 따라 국제적인 전자 시장 단일화를 위한 노력으로 UN/CEFACT와 OASIS를 주체로 하는 ebXML이 등장했다. ebXML은 기업간의 전자상거래 표준화를 위해 정의된 프레임워크로 기업 비즈니스 프로파일 및 약정, 기업의 공용 등록저장소, 비즈니스 프로세스 정의, 메시징 서비스 등에 대한 스펙을 정의하고 있다. 본 고에서는 ebXML 프레임워크 중에서 중요한 역할을 담당하고 있는 메시징 서비스에 대하여 소개하고, 현재 국내외에서 메시징 서비스에 대한 어떠한 연구 활동들이 일어나고 있으며, 전자상거래상의 비즈니스들의 확장성을 고려하는 글로벌 상호운용성에 대하여 설명한다. 또한 메시징 서비스의 핵심 역할을 제공하는 MSH의 구조와 기능에 대하여 자세히 설명하며, 향후 메시징 서비스의 개발 방향에 대하여 논의하고자 한다.

I. 서론

인터넷의 발달로 인하여 최근의 비즈니스 환경에서는 전자상거래라는 분야에 대한 많은 논의와 개발이 진행되고 있다. 초기의 전자상거래는 기업과 소비자를 직접적으로 연결해주는 B2C 분야의 사업이 많이 진행되었으나, 점차적으로 기업과 기업이 직접 연결되어 보다 높은 부가가치를 창출할 수 있는 B2B 시장이 활성화 되었다. 이러한 상황에서 거래 당사자간의 비즈니스에 대한 연동은 가장 중요한 이슈가 되어 활발한 연구가 진행되고 있다[1].

이러한 연구의 시초는 1960년대 후반에 등장하여 1980년대에 활발하게 진행되었던 전통적인 EDI(Electronic Data Interchange) 시스템들이다. 그러나 이 시대의 전통적인 EDI 시스템들은 메인 프레임과 같은 고 비용의 장비를 요구하며, VAN과 같은 고 비용의 망을 사용하게 되어 일부 대규모의 기업에서만 구축이 가능하였다. 또한 교환에 사용된 전자 문서의 형식이 복잡하여 구축과 유지 보수를

위해서는 전문적인 개발 기술과 인력이 필요하게 되었으며, 전자 문서에 대한 표준화 작업이 부족하여 타 시스템과의 확장성에 있어서 많은 어려움을 갖게 되었다. 이러한 상황에서 인터넷의 발달은 기존에 EDI 시스템에 대한 재정적인 부담을 갖고 있던 소규모 기업들에게 인터넷을 통한 전자상거래라는 새로운 기회를 제공하게 되었다. 우선적으로 고가의 메인 프레임을 대신하여 그동안 꾸준한 개발을 통해 성능이 높아진 PC와 워크스테이션급의 장비를 활용하였으며, 고 비용의 VAN을 대신하여 상대적으로 저렴한 인터넷을 사용하고 VAN이 제공해 주던 보안성은 인터넷을 위해 개발된 다양한 보안 프로그램들을 통해 해결하였으며, 기존에 분석과 이해가 어려웠던 전자 문서의 복잡도를 낮추어 개발자들이 문서에 대한 전문적인 지식이 없더라도 쉽게 문서의 내용을 이해할 수 있도록 하기 위해 XML로 문서의 내용을 표현하였다. 이 과정에서 XML로 표현된 전자 문서를 인터넷을 이용하여 전송할 수 있도록 SOAP(Simple Object Access Protocol)을 사용하



(그림 1) 전자상거래의 발전 단계

였다. 이러한 시도를 일반적으로 XML/EDI의 시작이라고 말할 수 있으며, 이러한 흐름을 타고 등장하였던 프레임워크로는 RosettaNet, BizTalk, eCo, cXML 등이 있다. 그러나 이러한 움직임들은 각 업체들이 나름대로 독자적인 연구 개발을 통해 구축된 업체 표준으로 전 세계적으로 운영하기엔 어려움이 많았다. 이러한 상황에서 UN/CEFACT와 OASIS는 전자상거래 시장에서의 단일화를 목표로 하여 ebXML 프레임워크를 정의하게 된다. ebXML은 기업간의 전자상거래 표준화를 위해 정의된 프레임워크로 기업 비즈니스 프로파일 및 약정, 기업의 공용 등록저장소, 비즈니스 프로세스 정의, 메시징 서비스 등에 대한 스펙을 정의하고 있다. 이와는 대조적으로 기존에 XML/EDI 형태의 프레임워크를 개발하던 업체들은 UDDI(Universal Description, Discovery and Integration)와 WSDL(Web Services Definition Language) 등으로 이루어지는 웹 서비스를 발전시키고 있다(그림 1) 참조).

UDDI는 웹 서비스를 등록하고 이를 실시간으로 검색할 수 있는 공용 디렉토리 또는 프로토콜의 집합체로 분산된 XML 기반 정보 레지스트리용으로 MS와 IBM, Ariba 등의 업체가 개발을 추진하고 있

다. 초기에는 웹 서비스는 서로 다른 시스템에서 제공하는 서비스들을 활용하여 새로운 서비스를 생성시키는 방식으로 인터넷을 통한 서비스 프로세스의 통합을 접근방식으로 하였으며, ebXML은 거래 당사자간에 비즈니스 프로세스의 연동을 목적으로 개발되었다. 현재는 웹 서비스와 ebXML이 적용 분야를 확장해 나아가면서 서로의 장점들을 수용하고 발전해 나가고 있는 상황이다.

본 고에서는 ebXML 프레임워크의 일반적인 특징과 이를 구성하는 요소 중에서 중요한 역할을 담당하고 있는 메시징 서비스를 중점적으로 살펴보고, 현재 국내외적으로 진행중인 메시징 서비스에 대한 연구동향을 살펴보고자 하겠다. 또한 메시징 서비스의 핵심 역할을 제공하는 MSH(Message Service Handler)의 내부 구조와 기능, 활용 분야를 정리하여 살펴보고, 향후 어떠한 연구가 필요한 것인가를 살펴보고자 하겠다.

II. ebXML 프레임워크 개요

ebXML은 다양한 전자상거래 프레임워크가 개발됨에 따라 발생하는 글로벌 비즈니스 통합에 대한

확장성의 문제 해결을 위해 단일화된 글로벌 전자상거래 시장 구성의 필요성이 증대하면서 등장하였다. ebXML은 XML 기반의 메시지 교환을 통하여 인터넷 상에서 기업의 규모나 위치에 상관없이 글로벌 e-marketplace에서 비즈니스를 수행할 수 있도록 제안된 개방형 전자상거래 프레임워크이다. 거래 상대방간의 정보 교환을 위해서는 서로간에 지원되는 비즈니스 협력 방식, 당사자의 역할, 메시지를 주고받는 방식에 대한 정보가 필요하게 된다. 본 장에서는 ebXML 프레임워크에서 정의한 구성 요소들 중에서 중요한 요소들에 대해 기술하고자 한다[2].

1. CPP/CPA/NCPA/NDD

ebXML에서는 CPP(Collaboration Protocol Profile)을 통해 거래 당사자의 협업을 위한 정보를 표현한다. CPP에는 거래 기업 정보, 전송 프로토콜, 전송 보안 프로토콜, 메시징 프로토콜, 비즈니스 프로세스 명세서 등의 정보를 담게 된다. CPA(Collaboration Protocol Agreement)는 CPP에 정의된 내용을 기반으로 거래 상대방과의 합의를 통해 도출된 협약 정보를 표현한다. CPA에는 거래 당사자들의 CPP의 공통적인 정보들과 메시징 서비스, 협업에 대한 요구 사항 등의 정보를 담게 된다. 실제 비즈니스가 수행될 때에는 CPA를 이용하여 정해진 메시지 교환 방식에 따라 거래를 수행하게 된다. 또한 최근 들어 CPA의 내용을 자동으로 협상하기 위해 NCPA(Negotiation CPA)와 NDD(Negotiation Descriptor Document)를 사용하는 방안도 나오고 있다. NCPA는 Negotiation BPSS(Business Process Specification Scheme) 문서를 기반으로 만들어지며, 이 NCPA에 상대방의 partyid, endpoint 등을 입력하고 NCPA의 협상 프로토콜을 따라서 협상하게 된다. NDD는 CPP나 CPA에 있는 내용 중에서 특정 항목의 협상 가능 여부와 협상의 범위 등을 표현하게 된다[3],[4].

2. 비즈니스 프로세스

거래 당사자간의 비즈니스 프로세스는 BPSS에

맞추어 작성된 BP 문서에 나타난다. BPSS에서는 실행 시간에 비즈니스 트랜잭션의 집합체가 수행되기 위해 시스템을 구성하는 데 필요한 요소들에 대한 명세를 제공하며, 의미와 요소, 비즈니스 협력을 정의하기 위해 필요한 특성 등을 제공한다. BP에는 실제적으로 전자 거래 수행을 위한 비즈니스 문서와 비즈니스 트랜잭션의 상태를 나타내는 신호들을 주고 받는 과정을 기술하게 된다.

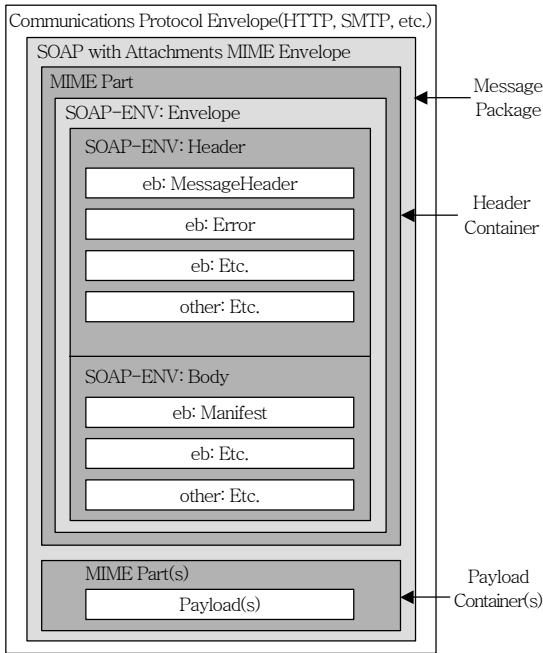
3. 등록 저장소

ebXML 프레임워크에서 주요한 구성 요소 중의 하나인 등록저장소는 거래에 필요한 기업의 정보를 등록하고, 검색하는 등의 거래를 위한 일련의 정보를 유지, 관리하는 기능을 제공한다. ebXML의 등록저장소에 대한 스펙은 크게 데이터 저장을 위한 RIM(Registry Information Model)과 저장된 데이터를 접근하기 위한 표준 API를 정의한 RS(Registry Service)로 분류된다. 등록저장소에는 기업의 거래 흐름을 정의하는 BPS, 거래를 위해 사용될 문서를 정의하기 위한 기본 정보를 제공하는 Core Component, 기업의 협업 정보들인 CPP/CPA 등의 정보들이 저장되게 된다. 이러한 정보들은 ebXML 레지스트리 서비스를 통해 ebXML 등록저장소에 BIOs(Business Information Objects)의 형태로 관리된다[5],[6].

4. 메시징 서비스

ebXML 메시징 서비스(ebMS) 스펙에서는 HTTP나 SMTP와 같은 통신 프로토콜을 이용하여 ebXML 메시지를 전송하기 위한 메시지 envelope와 header에 대한 문서 스키마를 정의하고 있다. 또한 ebXML 메시지를 송수신하기 위한 소프트웨어의 기능도 정의하고 있다[7].

ebMS는 SOAP을 통한 통신과 SOAP에 첨부물 갖고 있는 SWA(SOAP With Attachment)를 기반으로 확장된 레이어들을 정의하고 있다. 또한 국제적 문서 교환을 위해 필요한 보안성과 신뢰성과 관



(그림 2) SOAP과 SWA를 이용한 ebXML 메시지

련된 부분들도 정의하고 있다. 이 외에도 메시지에 대한 패키징과 라우팅, 전송 등과 같은 기능을 제공하게 된다.

(그림 2)는 SOAP과 SWA를 이용한 ebXML 메시지를 보여준다.

ebMS의 스펙은 EDI와 XML 등과 같은 다양한 형태의 메시지를 전송할 수 있도록 디자인 되었다. 이 스펙에서는 메시지에 사용되는 비즈니스 페이로드에 대한 언급이나, 메시지 전송중에 MSH가 페이로드 내의 문서에 대한 유효성이나 제약사항에 대한 처리를 어떻게 할 것인지에 대한 언급은 없으며, 하나의 메시지 방식으로 비즈니스 파트너들과 다양한 형태의 문서를 전달하기 위한 기능만을 명시하고 있다.

현재의 ebMS 스펙에서는 HTTP와 SMTP에 대한 바인딩 방식을 명시하고 있으며 점차적으로 다른 방식들에 대한 언급이 있을 것으로 예상된다. 비즈니스 문서의 신뢰성 있는 전송을 위해 Reliable Message 라는 모듈을 갖고 있다. 이것은 두 메시징 서비스 노드들 사이에서 재전송(retry)과 응답(acknowledgement)을 이용하여 메시지의 확실한 전송을 보장

해 준다. ebMS 스펙에서는 새로운 보안 모듈을 소개 하진 않고 있으며, 단지 인터넷상에서 e-비즈니스를 수행하는 데 사용되는 기존의 스탠다드를 사용하도록 제안하고 있다. ebMS는 구현을 위한 요구사항에 따라 크게 두 가지 형태로 구별하고 있다.

가. Core Modules

1) ebMS Packaging Based on SOAP & SWA

메시지 패키지는 크게 두 부분으로 이루어져 있다. 우선 패키징에 반드시 필요한 부분으로서 Header Container라 불리우는 마임 파트로서 SOAP 1.1 형태의 메시지를 갖는 부분이다. 또 다른 하나는 Payload Container라 불리우는 마임파트로서 애플리케이션 레벨의 비즈니스 문서를 페이로드의 형태로 담겨지는 부분이다. Payload Container는 패키지 내에 다수가 있을 수도 있고 없을 수도 있게 된다. ebXML 메시지의 구조는 (그림 3)에 표현되어 있다.

2) Security

이 모듈은 B2B에서의 안전한 상호작용을 제공하며 Non-repudiation, Integrity, Privacy, Confidentiality, Authorization 등의 특성을 제공한다.

3) SyncReply

몇몇 응용 프로그램들은 HTTP 프로토콜을 이용하여 하나의 커넥션을 통해 요청한 즉시 응답 메시지를 요구할 필요가 있다. 이러한 방식은 응답을 위해 별도의 커넥션을 생성하지 않아도 되는 장점이 있다. 동기식 통신을 사용할 경우에는 반대편에 존재하는 응용 프로그램이 특정한 요청에 대하여 동일한 커넥션을 이용하여 응답을 하게 된다.

4) Error Handling

이 모듈은 하나의 MSH가 다른 MSH에게 ebXML 메시지를 이용하여 에러를 전달하는 방법을 설명하고 있다. MSH의 에러 레포팅과 핸들링 모듈은 SOAP에서 제공하는 프로세스 레이어 위에 존재한다. 즉, SOAP의 관점에서 보면 MSH의 동작은 응용 레벨에서 메시지를 핸들링하는 것으로 보게 된다. 따라서 SOAP 프로세서는 메시지를 처리할 수

없는 경우에는 SOAP Fault 메시지를 발생시켜서 호출한 MSH에게 전달할 것이고, MSH는 이를 받아서 처리하게 된다.

나. Optional Modules

1) Reliable Message

이 모듈은 ebXML 메시지의 신뢰할 수 있는 전송에 대하여 기술하고 있다. 이 기능은 재전송과 MSH 레벨에서의 응답을 이용하여 메시지가 전송되는 것을 보장한다. 이 신뢰성을 보장하기 위해 반복시도 횟수와 반복시도 주기와 같은 파라미터가 CPA와 같은 문서에 명시되게 된다. 이 모듈은 MSH의 송수신을 위한 프로토콜을 액션 집합으로 명시하고 있다. 또한 이 정보들은 SOAP Header에 명시되게 된다.

2) Message Status Service

이 모듈에서는 MSH에 특정 메시지의 처리 상태를 질의하는 기능을 설명하고 있다. 모든 메시지는 자신만의 MessageId를 갖고 있어 해당 MessageId로 특정 메시지에 대한 상태를 요청하게 되며 응답으로는 NotRecognized, Received, Processed 등과 같은 상태 정보를 응답하게 된다.

3) Message Service Ping

이 모듈은 특정 MSH가 정상적으로 동작하고 있는지를 살펴보기 위한 기능을 제공한다. 이 기능은 두 개의 메시지로 이루어져 있다. 하나의 MSH가 ping 메시지를 다른 MSH에 전송하면 이 메시지를 받은 MSH가 pong 메시지로 응답하는 형태이다. 이 방식은 SOAP Body를 이용하여 ping/pong 메시지를 추출하게 된다.

4) Message Order

이 모듈은 메시지의 전송에 있어서 순차적인 진행을 제공한다. 이 모듈은 Reliable Messaging 모듈과 관련이 있으며 SOAP Header 부분을 이용하여 연동하고 있는 MSH의 정보와 상태 등을 추출하여 처리하게 된다.

5) Multi-hop

이 모듈은 메시지 전송 프로세스에서 발생하는 중계 역할을 하는 MSH에 대한 동작 방식을 기술하

고 있다. 이러한 역할을 하는 MSH의 임무는 송수신을 제공하는 MSH와는 다르게 동작된다. 또한 중계자가 포함된 메시지에 대한 Message Ordering과 Reliable Delivery를 만족시키기 위한 MSH의 역할도 기술하고 있으나 현재 나와 있는 스펙만으로는 정확한 동작 방식을 설명하기가 어렵다.

III. 메시징 서비스 관련 기술 동향

ebXML이 등장하면서 전 세계의 많은 기업들이 ebXML을 지원하는 플랫폼을 개발하고자 노력하고 있다. 특히 ebXML 프레임워크를 구성하는 많은 요소들 중에서 가장 기본이 되는 기능인 ebXML 등록 저장소와 메시징 서비스 부분을 가장 먼저 연구하기 시작했다. 이것은 이 두 가지 요소가 가장 중요한 요소이면서 스펙이 가장 상세하게 정의되어 구현이 용이했기 때문이다. 이러한 과정에서 많은 업체들이 자신들이 개발한 시스템에 대한 완성도를 테스트하거나 서로 다른 시스템간에 상호 운용성 테스트에 대한 인식이 높아지게 되었다.

한국의 전자거래진흥원(Korea Institute for Electronic Commerce)과 일본의 ECOM(Electronic Commerce Promotion Council), 대만의 TCA(Taipei Computer Association) 등에 의해 결성된 ebXML 아시아 위원회(ebXML Asia Committee)에서는 ebXML 메시징 서비스에 대한 상호 운용성 테스트를 위하여 ITG(Interoperability Test Group)을 결성하여 아시아 지역의 20여 개의 회사와 단체에서 개발중인 ebXML 메시징 시스템에 대한 상호 운용성 테스트를 진행하고 있다. 상호 운용성 테스트는 기존의 OASIS에서 제공하는 메시징 서비스 상호 운용성 테스트 스펙을 참조하여 일본의 ECOM에서 정의한 테스트 항목을 이용하고 있다.

테스트를 위한 평가 항목으로는 크게 T1부터 T5까지 구분하고 있으며 T1의 경우에는 best effort 기능에 대한 테스트를 목적으로 하여 ebXML 메시징의 기본이 되는 기능에 대한 평가를 주 목적으로 하고 있다. T2는 SyncReply 기능에 대한 평가로 현

재는 BPSS에서 처리해야 되는 비즈니스 메시지에 대한 부분을 제외한 순수 MSH 레벨의 시그널에 대한 처리 방식만을 평가하고 있다. T3의 경우에는 전송 프로토콜에 대한 보안 항목에 대한 평가를 목적으로 SSL 방식의 서버 인증, 클라이언트 인증의 평가를 목적으로 한다. T4는 XML signature에 대한 평가 방식으로 SOAP 메시지의 Header 부분에 대한 디지털 서명, 메시지의 페이로드에 대한 서명, NonRedudiation에 대한 처리를 평가 항목으로 한다. T5의 경우에는 메시징 서비스에서 필수적인 요소인 Reliable Messaging에 대한 평가 부분으로 요청 메시지에 대한 응답을 통한 확인 절차, 메시지 전송 실패에 대한 재전송 방식에 대한 테스트, 중복 수신된 메시지에 대한 처리, 메시지의 시퀀스 넘버에 따른 처리 방식 등에 대한 평가를 목적으로 한다[8].

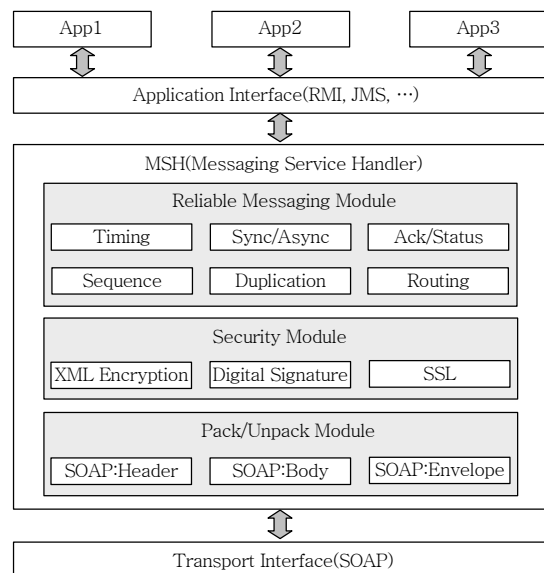
eAC의 ITG에서 진행중인 상호 운용성 테스트는 테스트 결과에 대한 검사 및 평가 방식에 대한 정확한 판단 기준이 되는 자료가 부족하고 자동화된 테스트 방식을 제공하지 않고 있어, 테스트를 참여하는 업체에서 상당히 많은 부분에 대한 수동적인 테스트 시간을 요구하게 되며, 평가 방식에 있어서도 두 업체가 동시에 테스트를 만족해야 성공으로 인정되므로 판정 방식으로 인해 한 파트너가 정상적으로 개발되었다 하더라도 테스트 파트너의 비정상적 동작으로 인한 불이익이 발생하는 경우도 있어 어려움이 있다.

이 외에도 한국기술표준원과 국내 기업들인 KTNET, INNODIGITAL, POSDATA, KWARE 등을 중심으로 결성된 KorBIT(Korea B2B/A2A Interoperability Testbed)를 살펴볼 수 있으며, 현재 OAG/NIST와의 협력을 통해 아시아 지역의 상호 운용성 테스트는 KorBIT에서 주관하고, 미주 지역의 운용성 테스트는 OAG/NIST에서 주관하여 상호간에 테스트 결과를 인증해 주는 방식을 협상중에 있다. 또한 전자상거래 표준화 통합포럼(ECIF)에서는 전자상거래의 지속적인 성장을 위해 상호 운용성에 대한 필요성이 높아짐에 따라서 ECIF를 중심으로 하는 e-비즈 상호 운용성 워킹그룹을 만들어 국

내외에서의 상호 운용성 테스트 베드에 대한 로드맵을 작성하는 등의 활동을 하고 있다[9].

IV. 메시징 서비스의 구조

메시징 서비스를 제공하기 위하여 메시지를 생성하고 신뢰성 있는 전송 기능을 제공하며, 송수신된 문서에 대한 관리 기능을 제공하는 모듈이 MSH이다. MSH는 거래 파트너로부터 인터넷을 통해 SOAP 기반의 ebXML 메시지를 전송 받아 상위 계층의 응용 프로그램에 전달하는 역할을 하게 된다. 일반적인 MSH의 구조는 (그림 3)과 같다.



(그림 3) Messaging Service Handler의 구조

1. Transport Interface

ebXML 메시징 서비스에서 제공하는 통신 프로토콜은 HTTP와 SMTP를 기본으로 한다. 일반적으로 전송(transport) 계층은 웹 서버를 이용한 서블릿의 형태로 HTTP 프로토콜을 지원하게 된다. SMTP 프로토콜을 사용할 경우에는 기존의 이메일 서버를 이용하는 경우가 대부분이며, 이때 SMTP 프로토콜의 한계로 SyncReply 방식의 통신은 사용

<표 1> CPA의 BTC에 대한 처리 방식

CPA의 BTC	Actions
isNonRepudiation-Required	· 'true'인 경우 모든 페이로드와 ebXML envelope을 부인방지를 위해 서명하고 로깅한다.
isNonRepudiation-ReceiptRequired	· 'true'인 경우 도착한 원문서에 대한 message digests를 ReceiptAcknowledgement 엘리먼트의 NonRepudiationInformation 엘리먼트에 값으로 넣어 보내야 한다. 그리고 그 메시지는 부인방지를 위해 서명하고 로깅한다.
isAuthenticated	· 'none'인 경우 전자 서명을 체크하지 않는다. · 'transient'인 경우 송신자를 SSL을 이용하여 체크한다. · 'persistent'인 경우 송신자를 전자 서명을 통해 체크한다. 페이로드와 envelope 모두 서명해야 되며, SSL을 통한 체크는 선택사항이다. · 'persistent-and-transient'인 경우 송신자는 전자 서명을 통해 체크하며, 페이로드와 envelope 모두 서명하고, SSL을 통해 체크해야 한다.
isConfidential	· 'none'인 경우 페이로드는 암호화 하지 않는다. · 'transient'인 경우 페이로드와 SOAP envelope을 SSL을 이용하여 암호화하며, 전자 서명은 사용하지 않는다. · 'persistent'인 경우 페이로드를 암호화한다. · 'transient-and-persistent'인 경우 페이로드를 암호화하며 암호화된 페이로드와 SOAP envelope은 SSL로 한 번 더 암호화 된다.

할 수 없게 된다.

2. Pack/Unpack Module

SOAP 프로토콜 방식으로 전송된 메시지는 XML 파서를 이용하여 파싱되어 SOAP:Header, SOAP:Body, SOAP:Envelope으로 구분되게 된다. SOAP:Header에는 송수신자 정보, CPAId, ConversationId, 서비스명, 액션명 등의 정보를 담고 있으며 이 값들을 이용하여 상위 모듈에서 암호호화를 할 것인지, 메시지를 어떻게 처리할 것인지를 결정하게 된다.

3. Security Module

인터넷을 통해 메시지를 전송하는 메시징 서비스는 보안 면에서 기밀성을 유지하지 못하는 취약점이 있다. 이를 극복하기 위해서 웹 서버에서 주로 제공하는 SSL(Secure Socket Layer)를 이용하게 된다. 또한 거래 상대방에 대한 확인 작업과 비즈니스의 진행 과정에서 발생한 문서에 대한 부인 방지 등의 기능을 제공하기 위해 전자 서명(digital signature)을 제공하게 되며 일반적으로 RSA-SHA1 알고리즘을 사용하고 있다.

Security 모듈은 메시지의 SOAP:Header의

CPAId에 해당하는 CPA 문서에 선언된 BTC (Business Transaction Characteristic)의 값을 이용하여 <표 1>과 같은 암호호화 과정을 수행하게 된다.

4. Reliable Messaging Module

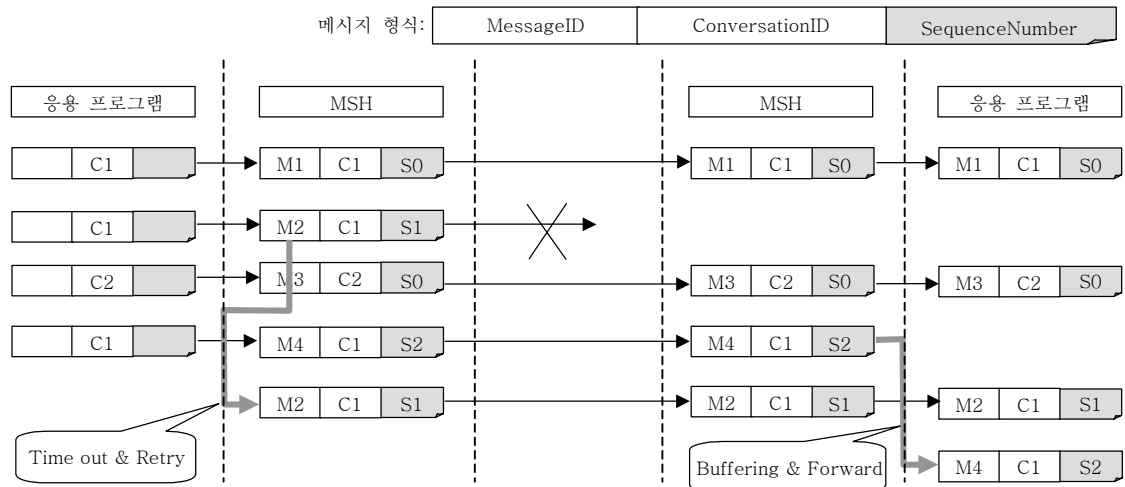
신뢰성 있는 메시지 전송을 위하여 송수신을 위한 메시지에 다양한 처리 과정이 필요하며, 이를 살펴보면 다음과 같다.

가. Timing

수신된 메시지의 eb:MessageHeader에는 eb:TimeToLive와 eb:Timestamp 엘리먼트가 있으며, 수신자 측은 이 값을 이용하여 수신된 메시지가 만료된 것인지를 판단하고 에러 메시지를 보내거나 받은 메시지를 무시하게 된다.

CPA에 정의된 timeToPerform이나 timeToAcknowledgeAcceptance, timeToAcknowledgeReceipt 같은 엘리먼트의 값은 송신자가 메시지를 전송한 후 해당 시간이 경과될 때까지 응답 메시지가 도착하지 않을 경우 재전송을 할 수 있도록 하는 역할을 한다.

이러한 처리 방식은 메시지 전송의 신뢰성을 위



(그림 4) 메시지 오더링

한 가장 기본이 되는 기능이 된다.

나. Ack/Status

수신된 메시지에 eb:AckRequested 엘리먼트가 세팅되어 있을 경우 수신측에서는 응답으로 Acknowledgement 메시지를 전송해야 한다. 응답으로 보내는 Acknowledgement 메시지의 eb:RefTo-MessageId 엘리먼트의 값은 수신했던 메시지의 eb:MessageId 값으로 세팅하게 된다. 이렇게 함으로써 송신측에서 다수의 메시지를 보낸 후 다수의 Ack를 받았을 때 이를 분별할 수 있게 된다.

다. Duplication

송신측에서 메시지를 전송하여 수신측에서 메시지를 수신한 후 동일한 eb:MessageId를 갖는 메시지가 중복되어 도착한 경우 수신측 MSH는 두번째 도착한 메시지를 무시하고 처리하지 않는다. 만약 메시지에 eb:AckRequested 엘리먼트가 세팅되어 있어도 두번째 메시지에 대해서는 Ack를 전송하지 않는다. 이러한 처리를 위해서는 이전에 전송받은 메시지에 대한 로깅을 남기거나 영속적인 (persistent) 데이터베이스에 저장하고 관리하는 기능이 필요하다.

라. Sequence

ebXML 메시징 서비스에서는 상위 계층에 존재하는 응용 프로그램이 사용 용도에 따라 원하는 대로 메시지를 분류할 수 있도록 eb:ConversationId를 제공하고, 전송하는 메시지에 순서를 부여할 수 있도록 eb:SequenceNumber를 제공하여 메시지 오더링(ordering) 기능을 지원한다. 메시지 오더링은 송신측에서 순차적으로 전달한 메시지가 수신측에 비 순차적으로 도착하였을 경우 수신측의 MSH가 상위 계층의 응용 프로그램에 순차적으로 전달할 수 있도록 하는 기능이다.

따라서 (그림 4)와 같이 수신측 MSH에 (C1,S2) 메시지가 (C1,S1) 메시지보다 먼저 도착했다 하더라도 수신측의 응용 프로그램은 (C1,S1), (C1,S2) 순으로 메시지를 전달 받게 되는 것이다.

5. Application Interface

MSH에서 정상적으로 처리된 메시지는 상위 계층에 있는 응용 프로그램에 전달되게 된다. ebXML 메시징 서비스 스펙에서는 이 부분에 대하여 특정 프로토콜로 제시하지 않고 있어 각 응용 프로그램에 맞는 프로토콜을 사용하여 전달해야 한다. 일반적으로 MSH를 자바 기반으로 개발하였을 경우에는

RMI를 통한 방식을 주로 사용하게 되는데, 이는 원격에서 접속하는 응용 프로그램이 자바 환경을 갖추고 있을 경우 두 시스템이 독립성을 유지한 채 MSH를 쉽게 사용할 수 있고, 직렬화 속도가 SOAP보다 빠르다는 장점이 있다.

하는 노력이 보다 절실히 요구되는 시점이다. 또한 각자가 개발을 통해 얻은 노하우를 바탕으로 아직까지 ebXML 메시징 서비스 스펙에서 명확하게 정의되지 못한 부분에 대한 지원 노력도 잊지 말아야 할 것이다.

V. 활용 분야 및 향후 전망

ebXML이 글로벌 비즈니스 통합을 위한 표준으로 등장하면서 많은 업체에서는 ebXML의 기술을 활용하기 위한 움직임을 보이고 있다. 현대자동차의 경우 일본의 미즈비씨 및 국내 VAN/EDI 사용자와 통합하고자 ebXML 메시징 서비스를 채택하고 있다. 또한 삼성, LG CNS, 한진택배, 데이콤과 같은 전통적인 EDI 방식을 사용하던 업체들 간에도 상호간 연동을 위해 KTNET에서 제공하는 GXMLHub를 이용하여 ebXML 메시징 서비스를 통한 상호연동 시스템을 활용하고 있다.

그러나 아직까지도 많은 부분들에 있어서 자체적으로 메시징 서비스를 개발하고 소규모 그룹을 구성하여 상호 운용성 테스트를 위한 노력을 진행하고 있으나 ebXML이 추구하는 진정한 비즈니스 프로세스의 통합을 위해서는 보다 체계적인 상호 운용성 평가 방식이 요구되며, 현재의 수동적인 테스트 방식들도 자동화 할 수 있는 환경으로 발전시키는 노력이 필요하다.

따라서 현재 산재되어 있는 다양한 테스트 베드들 간의 조율과 국제적인 협력을 통해 전세계에서 인증 가능한 글로벌 상호 운용성 체계를 구축하고자

참고 문헌

- [1] Morgan Stanley Dean Writer, *The B2B Internet Report*, Apr. 2000.
- [2] David A. Chappell 11, *Professional ebXML Foundations*, Wrox Press, 2001.
- [3] ebXML Collaboration Protocol Profile and Agreement Technical Committee, *Collaboration-Protocol Profile and Agreement Specification Version 2.0*, Sep. 2002.
- [4] ebXML Collaboration Protocol Profile and Agreement Technical Committee, *Automated Negotiation of Collaboration Protocol Agreements Specification Version 0.10*, Nov. 2003.
- [5] ebXML Registry and Repository Technical Committee, *Registry Services Specification Version 2.0*, Dec. 2001.
- [6] ebXML Registry and Repository Technical Committee, *Registry Information Model Specification Version 2.0*, Dec. 2001.
- [7] ebXML Messaging Services Technical Committee, *Messaging Service Specification Version 2.0*, Apr. 2002.
- [8] Electronic Commerce Promotion Council of Japan, *ebXML Interoperability Test Specification, Part I: ebXML Message Service Version 1.2.3*, Sep. 2003.
- [9] 한국전자거래진흥원, *ebXML Magazine*, Vol. 2, Feb. 2004.