

ebXML TRP 2.0을 위한 MSH

(A Message Service Handler for ebXML TRP 2.0)

송은정[†] 이호송^{**} 권택근^{***}

(Eun Jung Song) (Ho Song Lee) (Taek Geun Kwon)

요약 최근 인터넷을 통한 B2B 전자상거래가 국제적으로 확대됨에 따라 단일화된 전자상거래 표준의 정립이 시급히 요구되고 있다. 이에 UN/CEFACT와 OASIS는 국제 단일 전자 시장 형성을 목적으로 ebXML 표준을 채택하였고, 새로운 표준을 따르는 전자상거래 시장에서는 메시징 서비스를 위하여 ebXML 기반 메시지 전송 시스템이 필요하게 되었다.

본 논문에서는 ebXML 메시징 서비스 제공을 위하여 ebXML 메시지를 생성하고 신뢰성있게 전송하며 관리하는 MSH(Message Service Handler)를 구현하였다. 구현한 MSH는 다양한 시스템 환경을 지원하며, 단순한 인터페이스로 로컬 환경이나 분산 환경에서 MSH를 쉽게 사용할 수 있다. 또한 장애 발생시 메시지 재전송, 중복 메시지 처리, 확인(Acknowledgement) 메시지 전송 등을 통하여 인터넷 상에서 신뢰성있는 메시지 전송을 보장한다. 개발된 시스템은 현재 ebXML 등록기/저장소 시스템에 적용되고 있다.

키워드 : MSH, ebXML, TRP, 메시지 서비스 핸들러, SOAP, 전자상거래

Abstract Recently, as electronic commerce service via the Internet has expanded over the world, the standardization for the e-commerce has been strongly required. Therefore UN/CEFACT and OASIS adopted ebXML standard for the purpose of creating single global electronic marketplace. New e-Business systems should require ebXML-based messaging system for reliable messaging delivery.

In this paper, we have implemented Message Service Handler(MSH) packaging, transporting and managing e-Business messages for supporting ebXML messaging service. The MSH supports various system environments and allows high-level applications running either local or remote hosts with Remote Method Invocation(RMI) interface.

In addition, MSH guarantees reliable message delivery on internet since it retransmits message in case of delivery failure and handles duplicated message and applies acknowledgement message for secure message delivery. Nowadays MSH has applied to ebXML registry/repository.

Key words : MSH, ebXML, TRP, Message Service Handler, SOAP, electronic commerce

1. 서론

최근 통신망의 빠른 성장과 확산으로 인터넷은 다양한 분야에 적용되고 있다. 그 중 인터넷을 이용한 전자상거래는 초현대적인 편리함과 효율성을 제공한다는 장점으로 일반적인 상거래의 한 형태로 정착되고 있다. 초

기의 전자 상거래 형태는 일반 소비자를 대상으로 거래가 이루어지는 B2C였으나 기업들이 더 큰 이익창출을 위하여 기업을 대상으로 하는 B2B 전자상거래에 관심을 보이기 시작하였다. 현재 B2B의 점차적인 성장이나 대기업 중심의 투자 가능성을 볼 때 향후 전자상거래 시장은 B2B 분야가 주도적으로 이끌 것으로 전망된다.

이런 경향으로 전세계적 전자상거래 시장에서는 B2B 거래를 뒷받침 할 국제적 표준이 요구되고 있지만 현재까지는 명확히 나와있지 않은 실정이다.

초기엔 EDI(Electronic Data Interchange)가 비즈니스 정보 교환에 있어 효율적이라는 이상을 가지고 기업 간 거래를 주도했으나 대규모의 투자 비용 때문에 대기업 위주로 형성된다는 문제점이 발생하였다. 언급한 기업 간 전자상거래 문제를 해결하고자 많은 단체와 기업

· 본 논문은 BK21 대전, 충남 정보통신인력양성 사업단의 RA연구비 지원에 의한 것이다.

† 정회원 : 충남대학교 컴퓨터공학과
ejsong@ce.cnu.ac.kr

** 학생회원 : 충남대학교 컴퓨터공학과
hslee@ce.cnu.ac.kr

*** 종신회원 : 충남대학교 컴퓨터공학과 교수
tgkwon@ce.cnu.ac.kr

논문접수 : 2002년 4월 18일

심사완료 : 2003년 1월 9일

들이 상호 운용성을 보장하는 XML(eXtensible Markup Language)[1] 기반 표준을 정립하고 있으며, 이를 지원하는 프레임워크들을 개발하고 있다. 현재까지 발표된 대표적 프레임워크로는 RosettaNet[2], 마이크로소프트사의 BizTalk[3], CommerceNet의 eCo[4][5], cXML 프레임 등이 있다. 그러나 이러한 프레임워크들은 독자적으로 진행되고 프레임워크들 사이에 상호연동이 이뤄지지 않기 때문에 전세계적으로 통용되기엔 문제가 있다. 이에 UN/CEFACT와 OASIS는 상이했던 거래 표준을 하나로 통합하고 단일화된 국제적 전자상거래를 위하여 ebXML 표준안을 확정하였다. ebXML은 UN의 주도하에 기초적인 골격을 거의 갖추면서 곧 전자상거래 시장에 적용될 것으로 예상된다.

ebXML 표준을 따르는 거래는 기업이 저장소에 저장되어 있는 프로파일을 참조하고 기업간 조정 과정을 거친 후 계약이 이뤄지면 실질적으로 시작된다. 이때 저장소의 접근이나 기업간 신뢰성있는 메시지 전송은 메시지징 서비스를 이용한다.

본 연구는 ebXML 표준을 따른 B2B 전자상거래 기술 개발의 한 연구 분야로써 인터넷 상에서 보안을 유지하며 신뢰성 있는 전자상거래 문서 전송을 지원하는 MSH 시스템을 개발하였다. MSH는 신뢰성있는 메시지 전송 제공뿐만 아니라 SOAP(Simple Object Access Protocol)과 RMI(Remote Method Invocation) 기법 사용으로 플랫폼과 지리적 제약에 독립적이며, 확장성 및 재사용성을 지원한다.

논문의 구성은 다음과 같다.

2장에서는 메시지징 시스템의 특징을 살펴보고, 3장에서는 ebXML 표준과 TRP의 개요를 설명한다. 4장과 5장에서는 MSH 시스템의 설계 및 구현 내용을 기술하고 6장에서는 MSH 시스템의 적용 사례를 제시하겠다.

2. 메시지징 시스템

전통적으로 기업간에는 EDI(Electronic Data Interchange) X.400 전송 프로토콜이 높은 신뢰성과 다양한 기능을 제공하고 비즈니스 정보 교환에 있어 효율적이라는 판단에 기업간 메시지 전송을 지원했으나, 인터넷이 도입되고 전자상거래가 점차 확대됨에 따라 골한계를 드러냈다. 독자적인 부가가치 통신망을 통해 융통성 없이 데이터를 고정된 형태로 교환해야 했고, 불필요한 기능이 많아 시스템 속도가 느렸으며, EDI의 초기 투자 비용과 시스템 유지비용을 감수해야 하기 때문에 소수 기업에서만 사용하였다.

이런 문제점을 해결하고자 메시지징 시스템들이 제시되었고 현재 B2B 전자 상거래에서는 XML 메시지징 시스

템이 일반적이다.

XML은 HTML로 구성된 인터넷상 문서의 여러 가지 단점과 한계를 극복하기 위해 등장한 표준으로 프로그래밍 언어 및 플랫폼에 독립적이며, 사용자 정의형 태그를 사용하여 그 자체만으로 데이터와 메시지를 전달한다. 또한 DTD를 통하여 문서 자체에 문서의 구조를 기술할 수 있어 다양한 형식의 문서들을 일관성 있게 표현할 수 있다. 이런 장점으로 많은 기업과 단체들은 XML 기반 메시지징 시스템을 개발하고 있다.

각 비즈니스 시스템간 메시지징 시스템의 사용은 비즈니스 시스템을 서로 밀접하게 묶지 않고도 조직끼리 협력할 수 있게 하고 작은 단위의 비즈니스 시스템이 B2B에 참여하는 것을 가능하게 했다는 점에서 현 B2B 솔루션의 핵심 요소라 할 수 있다. 현재 메시지징 시스템에 적용되고 있는 기술들을 보면 XML 메시지징 서비스를 제공하는 JAXM(Java API for XML Messaging)[6], 편사가 자바를 기업형 환경에 적용할 수 있도록 만든 기업형 메시지징 표준인 JMS(Java Message Service)[7], 간단한 XML 기반 프로토콜인 SOAP 등이 있다.

3. ebXML

본 장에서는 ebXML의 개요와 전자상거래에서 메시지 전달에 관한 표준인 ebXML TRP를 간단히 소개한다.

3.1 ebXML의 개요

ebXML은 XML 및 인터넷 기반 개방형 전자상거래 표준으로 모든 거래 당사자들에게 상호 운용적이며 일관성있는 방법으로 전자상거래 정보 사용을 가능하도록 해준다. 전세계적 단일 전자 상거래 시장 구축이라는 목표 아래 지난 2001년 5월 UN/CEFACT(The United Nations for Trade Facilitation and Electronic Business)와 OASIS(Organization for the Advancement of Structured Information Standards)에 의해 표준으로 채택되었다[8].

ebXML은 현재 국제적 규격인 XML을 기반으로 구성된 문서 구조이고 국제 전자 문서 표준의 제정 기구인 UN/CEFACT가 직접 프로젝트에 참여, 지원하여 전세계적 전자상거래 기본 조직의 제공을 목적으로 추진되어 왔다. 이런 면으로 볼 때 ebXML은 다른 표준화 작업과는 달리 국제적인 표준 프레임워크로써 적용될 수 있는 표준이다.

ebXML 기본 구조는 6가지로 나누어 볼 수 있다.

전체 시스템 구조를 정립하는 기술적 구조[9], 거래 상대방들에게 특정 비즈니스 시나리오에 대한 상세한 기술을 제공하는 비즈니스 프로세스[10], 실제세계의 비즈

니스 개념에 대한 정보와 다른 비즈니스 정보와의 관계성 및 구문을 정의하는 핵심 컴포넌트, 두 거래 당사자들간의 상호 운용성 보장을 위하여 거래 상대자들에 대한 프로파일과 거래 상대자들간의 계약에 관련된 부분을 담당하는 거래 파트너[11], 거래 당사자들의 정보, 비즈니스 프로세스, 핵심 컴포넌트를 저장하고 그에 따른 질의 및 등록 서비스를 지원하는 등록기/저장소, 거래 상대자들간의 비즈니스 메시지들을 교환하기 위한 표준을 제공하는 전송, 라우팅 & 패키징[12]이다.

3.2 ebXML TRP(Transport, Routing & Packaging)

ebXML 기반 전자상거래에서 인터넷을 통한 모든 비즈니스 문서의 전송은 신뢰성있는 전송을 위하여 ebXML TRP(Transport, Routing & Packaging) 규격을 따르는 메시징 서비스를 이용해야 한다.

3.2.1 ebXML 메시지 구조

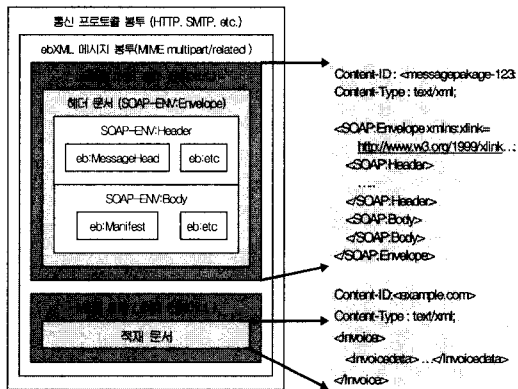


그림 1 ebXML 메시지 구조

ebXML 메시지 구조는 외부의 통신 프로토콜 봉투와 내부의 ebXML 메시지 봉투로 구성되며 서로 독립적이다. 메시지 패키지에는 ebXML 헤더 문서를 담고 있는 헤더 컨테이너와 실제 전송되는 데이터를 담는 적재 컨테이너, 두 MIME으로 구성되고 헤더 컨테이너 내부는 메시지 전송 정보를 담은 SOAP 헤더와 SOAP 바디로 구성된다.

ebXML 메시지 헤더는 메시지 전송에 필요한 정보를 담은 "MessageHeader", 오류를 표현한 "ErrorList", 전자서명을 포함한 "Signature", 동기적 응답을 요구하는 "SyncReply", 응답 메시지를 요구하는 "AckRequested", 메시지의 재정렬을 요구하는 "MessageOrder" 요소들로 구성된다.

예제 1 ebXML 메시지

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
<SOAP:Header>
<eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0">
<eb:From>
<eb:PartyId>urn:duns:123456789</eb:PartyId>
</eb:From>
<eb:To>
<eb:PartyId>urn:duns:912345678</eb:PartyId>
</eb:To>
<eb:CPIAId>20001209-133003-28572</eb:CPIAId>
<eb:ConversationId>20001209-133003-28572</eb:ConversationId>
<eb:Service>urn:services:SupplierOrderProcessing</eb:Service>
<eb:Action>NewOrder</eb:Action>
</eb:MessageData>
```

3.2.2 메시지 처리

인터넷을 통한 전자상거래에서 중요한 것은 신뢰성있는 메시지 전송으로 다음과 같이 처리한다. 상위 응용 시스템의 요구 발생시 메시지를 생성하고 저장 장치에 저장한 후 라우팅 테이블을 참조하여 다음 MSH로 전송한다.

수신측은 전달 받은 메시지가 중복되었는지 확인, 저장하고 메시지를 수신하였음을 알리는 확인 메시지를 송신측에 보낸다. 만약 중복 메시지라면 그 전에 전송했던 확인 메시지를 재전송하고, 현재 메시지 수신측이 최종 도착지가 아니라면 위와 같은 방식으로 메시지를 목적지로 전송한다. 이때 MSH 간 전송 여부는 확인 메시지로 파악한다. 그림 2의 (1),(3)과 같이 메시지를 전송하면 송신측은 (2),(4)와 같이 전송된 메시지의 확인 메시지를 송신측에 보낸다. 확인 메시지를 요구한 메시지의 응답이 오지 않았을 경우 분실로 간주하여 저장된 메시지를 일정한 간격으로 재전송 횟수만큼 전송하고 메시지 전송 도중 오류가 발생하면 오류 메시지를 생성하여 메시지를 보낸 MSH에게 오류 발생을 알린다. 또한 메시지 암호화[13][14]기법으로 메시지 변조 여부를 확인한다.

ebXML에서는 안전한 메시지 전송을 위해 XML 전자서명기법을 사용하여 권한 부여, 메시지 보존, 변조

여부 테스트 등을 하지만 보안이 우선이라 할 수 있는 전자상거래에서 ebXML 표준의 보안 기술 부분은 아직 미흡한 실정이다.

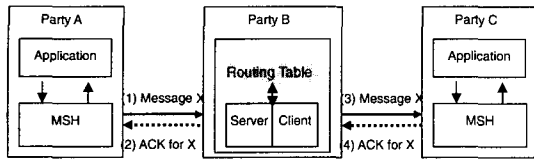


그림 2 메시지 전송

4. MSH 시스템 설계

4.1 시스템 설계 명세 및 적용한 방법론

- ebXML TRP 규격서 지원
- 객체지향 기반 : 소프트웨어 모듈을 객체개발방법론 기반으로 개발하여 재사용이나 버전 확장을 용이하게 하였다.
- HTTP 기반 전송 : HTTP를 기반으로 구현하여 방화벽 문제를 해결하고 기존의 보안 기술이 쉽게 적용 가능하도록 하였다.
- 플랫폼 독립 운용성 : 자바를 사용하여 윈도우 9X, NT, 2000과 리눅스, 솔라리스 등 JVM이 설치 가능한 모든 플랫폼에서 동작 가능하도록 하였다.
- 다양한 응용 지원을 위한 인터페이스 설계 : 객체간 인터페이스를 최소화 함으로써 새로운 버전의 ebXML TRP 규격에 대처 가능하도록 하였고, RMI를 사용하여 로컬 및 원격 호스트에서의 수행 및 다양한 전자상거래 응용과의 연동을 용이하게 하였다.
- 분산 시스템의 연동 : XML과 HTTP를 기반으로 하는 SOAP을 이용하여 이 가중 시스템과 객체를 공유하여 사용할 수 있게 하였고, 독자적인 기술로 개발되어 상호 운용성 문제를 용이하게 해결하였다.
- 기술 발전에 따른 유연성 : SOAP 모듈, XML 파서, 웹 서버 등 공개 소프트웨어를 채택하여 이들 소프트웨어의 기능 확장시 쉽게 적용 가능하도록 하였다.

4.2 MSH 구성

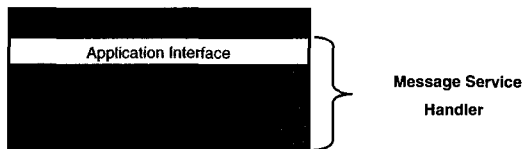


그림 3 ebXML 메시지 서비스 핸들러 계층도

MSH는 B2B 전자상거래 시장에서 ebXML 표준을 기반으로 ebXML 메시지 전송 서비스를 제공하는 시스템이다.

이는 비즈니스 관련 문서의 신뢰성있는 전송 뿐만 아니라 다른 MSH의 상태나 전송된 메시지의 상태 확인 서비스도 제공한다.

MSH 시스템은 HTTP 요청/응답 형태를 기반으로 XML의 확장성을 접목시킨 SOAP을 전송 프로토콜로 사용하고, ebXML TRP 규격에 따라 메시지 생성, 패키징, 전송한다. B2B 응용 시스템들은 MSH를 통하여 메시지를 전송하기 위해 RMI 인터페이스를 사용한다.

4.3 클래스 다이어그램

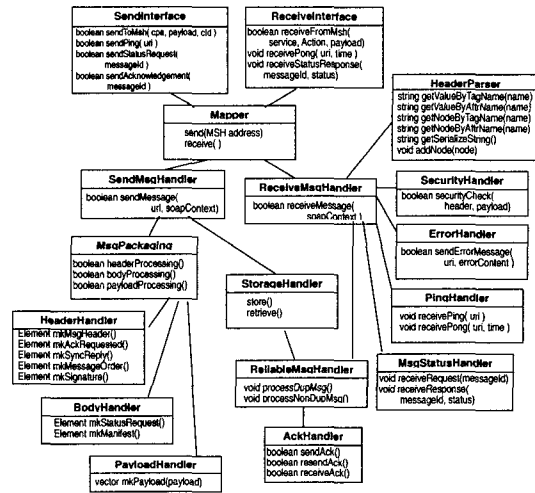


그림 4 MSH 클래스 다이어그램

그림 4는 메시지 송신 부분과 메시지 수신 부분 클래스들을 몇 개의 핸들러로 통합하여 표현하고 있다. 대표적인 클래스로는 상위 응용 시스템으로부터 정보를 받는 클래스, ebXML 표준에 맞게 메시지를 생성하는 클래스, 메시지를 저장 및 관리하는 클래스들이 있다. 또한 수신한 메시지의 오류 검사, 보안 검사하는 클래스, 신뢰성있는 메시지 전송 및 처리를 담당하는 클래스, 상위 응용 시스템으로 메시지를 알리는 클래스 등이 있다. 이외에도 다중 MSH에서의 메시지 전송을 처리하는 클래스, 다른 MSH나 전송된 메시지의 상태를 제공하는 클래스들이 있다.

5. MSH 시스템 구현

본 시스템은 신뢰성 있는 메시지 전송을 위하여 ebXML TRP 1.0, 2.0 규격을 따른 MSH(Message Service Handler)를 구현하였다. TRP 1.0으로 설계된 MSH에서 2.0으로 확장한 구조로 버전 2.0을 기반으로 버전 1.0 메시지 처리도 지원한다. 버전 1.0과 2.0의 메

시지 구조는 약간의 차이가 있기 때문에 메시지 처리 및 전송은 수신 메시지의 버전을 따른다.

시스템 구현은 윈도우 환경에서 자바언어를 사용하였고 XML 문서를 파싱하기 위한 파서로는 xerces를 이용하였으며 웹 서버는 탐켓을 활용하였다. MSH의 서버는 탐켓뿐만 아니라 웹프록 서버도 대체 가능하며 실행 환경은 윈도우 계열, 유닉스, 리눅스, 솔라리스 모두에서 실행 가능함을 테스트하였다.

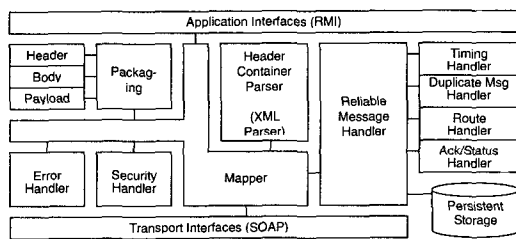


그림 5 MSH 구조

그림 5는 전체적인 MSH 구조를 나타낸 것으로 상위 응용 시스템과의 인터페이스로 RMI를 사용하고 전송 인터페이스로 SOAP을 적용한다. 이것은 지리적으로 분산된 응용 시스템을 통합하여 메시지 서비스를 지원하고, 하드웨어 및 OS와 독립적으로 운용되도록 하였다. ebXML의 TRP 규격은 복잡하여 원격 응용 시스템에 적용하기 어렵기 때문에 RMI 인터페이스로 상위 응용 시스템과의 연동을 단순화하였다. 대부분의 비즈니스 프로세스를 위하여 단지 몇 개의 함수로 메시지 서비스를 제공할 수 있다.

5.1 전송 인터페이스

전송 인터페이스에 사용되는 SOAP(Simple Object Access Protocol)은 분산된 컴퓨팅 환경에서 데이터를 공유하기 위해 고안된 XML 기반 프로토콜로 언어, 운영체제, 객체 모델과 관계없이 다른 컴퓨터의 함수나 객체를 사용할 수 있게 한다[15]. 1998년 IETF와 W3C에 표준 안으로 제출되었으며 상업적으로 다른 XML 프로토콜에 비해 많은 성과를 보이고 있다. SOAP은 메시징 시스템부터 RPC에 이르기까지 광범위한 영역의 상위 응용 시스템을 지원할 수 있으며 HTTP, SMRTP, MIME 등과 같은 기존의 인터넷 프로토콜 형식과 복합적으로 사용될 수 있다.

시스템에서는 인터넷 표준으로 가장 널리 사용되는 전송 메커니즘인 HTTP를 이용하여 메시지를 전송한다. HTTP POST 함수의 요청/응답 모델을 따르고 HttpServletRequest, HttpServletResponse 객체를 매

개변수로 하는 서블릿 기법을 이용한다.

5.2 MSH(Message Service Handler)의 내부적 구조 및 처리

MSH는 크게 송신 MSH와 수신 MSH로 구분되지만 신뢰성있는 메시지 전송을 위한 부분과 보안을 위한 전자서명, 메시지 저장 및 관리 등 많은 부분이 두 MSH에 재사용되기 때문에 이 절에서는 송·수신으로 구분하지 않고 각각의 기능별로 기술하겠다.

패키징 핸들러는 응용 시스템에서 받은 정보와 적재 문서를 표준에 맞는 메시지로 생성하고 목적지로 전송한다. 이때 경로 설정을 위하여 라우팅 핸들러를 참조한다.

파서는 수신한 XML 문서에서 필요한 정보들을 추출하는 기능을 하고, 오류 핸들러는 수신한 메시지에 오류가 발생했을 때 오류 메시지를 생성하여 송신측에 알리는 기능을 한다.

타이밍 핸들러는 타임 스탬프를 생성하거나 재전송 시간 계산 등의 시간 관련 일을 수행한다.

송·수신 메시지는 저장 장치에 저장하여 신뢰성있는 메시징 핸들러로 중복 메시지, 분실 메시지를 처리하고 라우팅 핸들러로 다중 MSH 메시지 전송을 수행한다. 다중 MSH 전송을 위해서 각 MSH는 최종 도착지에 대한 다음 MSH 주소를 갖는 라우팅 테이블을 관리한다.

다른 MSH의 상태는 핑, 톱 메시지의 타임 스탬프 값으로 왕복 시간을 측정하여 파악한다. 응답으로 톱 메시지를 받지 못하면 시간 초과로 처리하고 상대방 MSH가 작동하지 않음으로 인식한다.

보낸 메시지의 상태는 상태를 알고자 하는 메시지의 아이디어를 포함한 메시지를 보내면 수신 MSH는 저장된 메시지들을 검색하여 메시지의 상태를 응답으로 알려준다.

보안 핸들러는 전자서명 생성 및 메시지 수신 후 메시지의 정확성을 검증한다.

ebXML 전자상거래 표준에서는 보안을 위하여 XML 전자서명 기술을 따르고 있다. 메시지는 전자서명 생성 과정을 거쳐야 하고, 송수신 때마다 검증 단계를 수행하여야 한다.

DSA-SHA1 알고리즘으로 암호화한 값을 전송 메시지에 추가하고 수신측은 메시지를 같은 알고리즘에 적용, 수신한 암호화 값과 비교하여 메시지의 변조여부를 파악한다. 다중 MSH 전송에서는 메시지 헤더의 일부 데이터가 변경되므로 암호화 값도 변경하여 전송한다.

그림 6은 메시지 전송에서 동기적 응답을 받는 과정을 다이어그램으로 표현한 것이다.

상위 응용 시스템이 동기적 응답을 요청하는 메시지를 생성, 저장하고 HttpServletResponse 객체를 이용

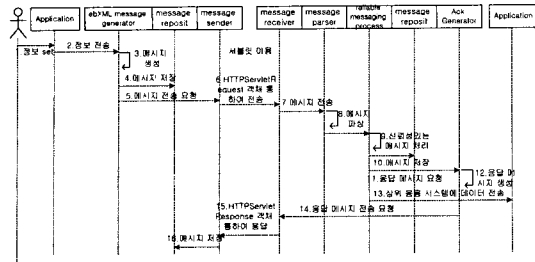


그림 6 동기적 응답 메시지 전송에 대한 순차 다이어그램하여 전송한다. 수신측은 객체를 통해 메시지를 받으면 XML문서에서 필요한 정보를 파싱하여 조건에 맞게 처리 및 저장하고 동기적 응답 메시지를 생성하여 HTTP ServletResponse 객체로 응답한다. MSH는 받은 메시지를 상위 응용 시스템에 함수 호출로 알려주고 상위 응용 시스템은 메시지의 응답을 위해 다시 그림 6의 과정을 따른다.

비동기적 응답을 요청하는 MSH는 동기적 메시지 처리와 달리 응답을 기다리지 않고 다음 작업을 진행한다. 그러면 수신측은 응답 메시지가 준비되었을 때 전송한다.

5.3 신뢰성있는 메시징 전송

본 시스템의 특징 중의 하나는 인터넷을 통한 전자상거래에서 신뢰성있는 메시지 전송을 제공한다는 것이다.

송신 MSH가 메시지를 전송하면 수신 MSH는 성공적으로 메시지를 수신했음을 알리기 위해 응답 메시지를 생성하여 발송한다.

송신 MSH는 메시지 전송에 실패했거나 응답 메시지를 분실하여 응답을 받지 못했을 경우는 메시지 분실로 판단하여 응답메시지를 받을 때까지 재시도 횟수만큼, 일정한 간격으로 저장했던 원래 메시지를 재전송한다. 또한 시스템의 정지로 메시지 전송에 실패했을 경우는 시스템 회복과 함께 메시지의 재전송을 시도한다.

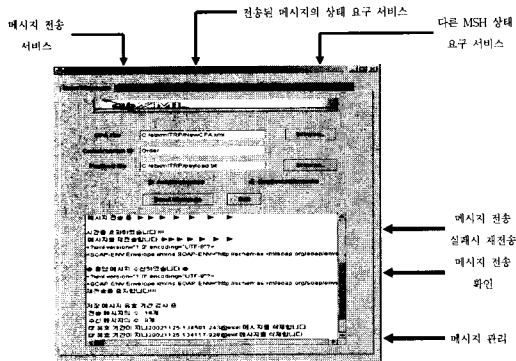


그림 7 신뢰성있는 메시지 전송 테스트 화면

전송 MSH가 전송 횟수만큼 전송 후에도 응답 메시지를 수신하지 못하면 상위 응용 시스템에 실패를 알린다.

그림 7은 신뢰성있는 메시징 전송 테스트 프로그램으로 메시지 전송, 메시지 분실 및 전송 오류에 따른 메시지 재전송, 확인 메시지 수신, 메시지 관리 등을 보여준다.

이외에도 수신측은 전달 받은 메시지가 중복되었는지 확인하여 중복 메시지라면 앞서 전송했던 응답 메시지를 재전송한다.

5.4 전송 메시지의 저장 및 관리

ebXML TRP에서는 신뢰성 있는 메시지 전송을 보장하기 위해서 송신 및 수신 메시지를 저장하고, 메시지들은 고유한 메시지 아이디[16]로 다른 메시지와 구분한다. 메시지 아이디는 유일한 값을 주기 위해 UTCTime Stamp클래스에서 타임스탬프 값과 호스트 이름을 조합하여 생성한다. 메시지는 전송되는 시점에 전송의 성공 여부와 관계없이 메시지의 모든 내용을 오브젝트 형태로 변형, 벡터에 추가하여 파일에 저장하여 관리한다.

즉, 하나의 벡터는 여러 개의 오브젝트 타입 XML Object로 구성되고, 새로운 메시지는 오브젝트로 변형하여 벡터에 추가된다. 이때 메시지 아이디로 메시지 중복 여부를 검사하여 중복된 메시지는 저장하지 않고 응답 메시지만을 다시 전송한다.

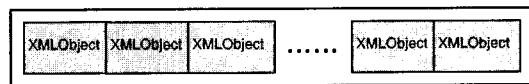


그림 8 파일에 저장되는 Vector 클래스의 구조

메시지의 삭제는 MSH가 일정한 시간마다 검사하여 유효기간이 만료된 메시지에 한해 이루어지며, 삭제 없이 종료되면 재실행되는 시점에서 메시지를 삭제한다.

5.5 상위 응용 시스템과의 인터페이스

MSH 간 메시징 서비스를 위해서는 우선 상위 응용 시스템과 MSH 간의 통신이 이루어져야 하는데 본 시스템에서는 상위 응용 시스템과의 통신을 위하여 RMI 기법을 사용한다.

RMI는 분산 환경에서 메시지 전송이 아닌 다른 시스템의 함수 호출로 통신하는 기법으로 응용 시스템과 MSH 시스템의 연동을 단순하게 해준다.

상위 인터페이스로 RMI를 사용한 것은 응용 시스템이 로컬 환경뿐만 아니라 분산환경에서도 MSH 서비스를 제공받고, SOAP을 사용하고자 할 경우 상위 응용 시스템도 MSH의 환경을 모두 갖춰야 하는 시스템적 낭비가 발생하게 되므로 MSH와 상위 응용 시스템과의 독립성을 위해 SOAP이 아닌 RMI를 사용하였다. 또한

RMI는 원격 호스트의 응용 시스템이 자바환경을 갖추고 있다면 완전히 다른 시스템일지라도 독립성을 유지한 채 MSH를 쉽게 사용할 수 있고, 직렬화 속도가 SOAP보다 빠르다는 장점이 있다.

ebXML 문서에는 전송을 위한 메타 정보를 포함하고 있는데 그 정보는 거래 당사자의 계약 문서인 CPA (Collaboration Profile Agreement)를 따른다. 상위 응용 시스템이 이 계약 문서와 전송 데이터를 지정하고 전송을 요구하면 송신 MSH에서 메시지 생성 및 전송 작업을 수행한다. 메시지 수신 MSH는 상위 응용 시스템의 함수를 호출하여 메시지를 전송한다.

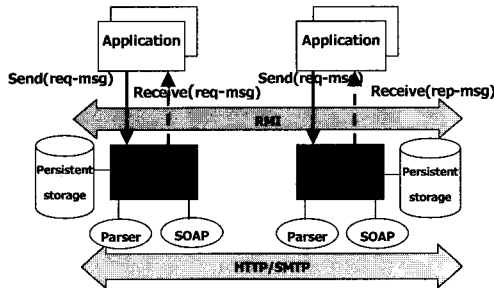


그림 9 메시지 송신 및 수신

표 1은 상위 응용 시스템이 MSH를 사용할 수 있도록 제공된 인터페이스이다. 응용 시스템은 이 함수들을 사용하여 메시지 송신 및 수신을 할 수 있다.

표 1 상위 응용 시스템과 MSH 간의 인터페이스

함수	설명
SendMessage()	메시지 송신을 위한 생성자
void setConversationId(String conversationId)	메시지 그룹 아이디 지정
void setCPAFile(String path)	CPA 파일 지정
void setPayload(Vector payloads)	전송할 문서 첨부
void setAckRequested(String value)	다음 MSH에 수신되었을 때 확인 메시지 요청
void setRefToMessageId(String messageId)	현 메시지와 관련된 메시지의 아이디 지정
boolean send(String MSHAddress)	MSH로 전송
boolean sendPing(String MSHAddress, String uri)	특정 MSH의 상태 요구
boolean sendStatusRequest(String MSHAddress, String messageId)	메시지의 상태 요구
APPServerImpl()	메시지 수신을 위한 생성자
receiveFromMsh(String service, String action, Vector payloads, String syncReplyMode, String messageId)	메시지 수신
receiveStatusResponse(String messageId, String messageStatus)	해당 메시지의 상태 수신
receivePing(String uri, String ping)	해당 MSH의 상태 수신

6. 구현 적용 사례 및 활용 분야

자체적인 테스트 프로그램으로 신뢰성 있는 메시지

전송, 오류 탐지, 핑 서비스, 전송된 메시지의 상태 확인 서비스 등이 정상적으로 이루어짐을 확인했다.

다른 시스템으로의 적용은 ebXML 등록기/저장소 시스템[17]에서 정보, 비즈니스 프로세스, 핵심 컴포넌트를 저장하고, XML 형태로 질의, 등록, 수정, 삭제 할 때 사용하였다.

그림 10은 객체를 등록하는 화면으로 클라이언트가 등록을 요청하면 생성된 객체가 메시지 서비스 핸들러를 통하여 서버에 저장된다. 그림 11은 Browse & Drilldown 질의 서비스 화면으로 왼쪽의 분류 트리에서 원하는 노드를 선택하면 검색 결과가 오른쪽 창에 나타난다. 이때 질의 요청이나 실행 결과를 반환할 때에도 메시지 서비스 핸들러를 사용한다. 등록기/저장소 시스템에서는 이외에도 객체 관리 서비스인 객체 등록, 객체 승인, 객체 경시, 객체 제거 서비스와 각종 질의 서비스에도 적용된다.

이 뿐만 아니라 ebXML 메시지 시스템은 ebXML 표준을 따르는 웹 기반이나 이동 통신 전자상거래 시스템에 적용될 수 있다.

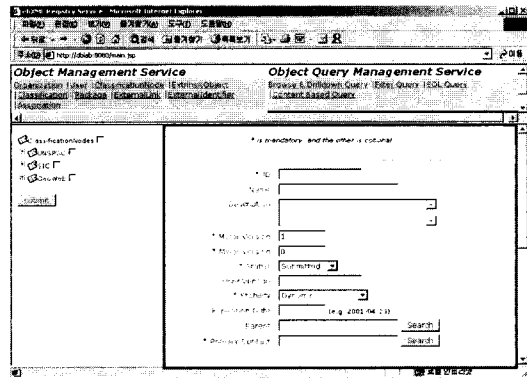


그림 10 Object 등록 인터페이스

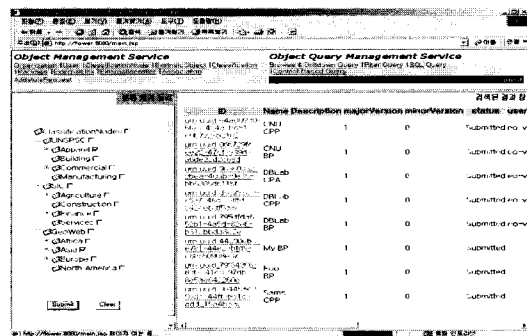


그림 11 Browse & Drilldown Query 인터페이스

7. 결 론

본 논문에서는 차세대 전자상거래에서 ebXML 표준을 기반으로 비즈니스 문서의 신뢰성 있는 전송을 지원하는 메시지 서비스 핸들러를 설계 및 구현하였다.

완벽한 ebXML TRP v1.0, v2.0 규격을 지원하고 RMI와 SOAP을 이용하여 지리적 제약 및 플랫폼에 독립적이며, 다중 MSH 환경에서의 메시지 전송에서도 신뢰성 보장이 된다.

또한 단순한 인터페이스로 쉽게 활용될 수 있으며, 유연성을 제공하기 위해 공개 소프트웨어인 XML 파서, 웹 서버와 soap을 수정, 활용하여 소프트웨어 확장이 쉽게 이루어진다.

현재 TRP 기술 명세서에는 보안 내용과 다중 MSH 간에서의 메시지 전송 및 처리 내용, 상위 응용 시스템과의 인터페이스에 관한 자세한 기술은 없다. 메시지 변조여부를 판단할 수 있는 수준의 보안 기술이 실려있는데 인터넷과 같은 개방형 네트워크에서 이 수준의 보안은 거래의 모든 과정이 무방비한 상태로 노출되어 있는 것과 같다. 또한 상위 응용 시스템과의 인터페이스나 다중 MSH 간 메시지 전송 및 처리는 개발 중인 기업들 처리를 생각하거나 본 연구처럼 나름대로의 기술을 적용하고 있는 실정으로 빠른 시일 내에 구체적이고 명확한 표준이 나와야 하겠다.

현재 급속히 성장하는 기업간 전자상거래는 모든 기업에 있어서 피할 수 없는 현실이며 그에 부응하여 각 기업과 단체에서는 각자의 표준을 정립하고 개발에 주력하고 있다. 그 중 UN이 제시한 ebXML은 국제 표준으로 자리 잡았고 향후 전자상거래에서 큰 역할을 할 것이다. 이에 전자상거래에서의 메시지 서비스 시스템에 관한 연구가 불가피할 것이며 본 연구는 차세대 전자상거래 표준인 ebXML의 메시지 서비스 시스템 연구에 기초가 될 것이다.

참 고 문 헌

- [1] W3C, Extensible Markup Language(XML)-1.0 (Second Edition), <http://www.w3.org/TR/2000/RFC-xml-200106>
- [2] RosettaNet, <http://www.rosettanet.org>
- [3] Microsoft BizTalk Server, <http://microsoft.com/biztalk>
- [4] eCo Architecture for Electronic Commerce Interoperability, <http://eco.commerce.net>
- [5] eCo Semantic Recommendations, <http://eco.commerce.net>
- [6] Java API for XML Messaging (JAXM), Specification Version 1.1, <http://java.sun.com/xml/jaxm/index.html>

- [7] Java Message Service, <http://java.sun.com/products/jms>
- [8] 차세대 전자상거래 표준화 웹사이트, <http://www.ebxml.or.kr>
- [9] UN/CEFACT and OASIS, ebXML Technical Architecture Specification Version 1.04, <http://www.ebxml.org/specs/ebTA.pdf>, 2001.
- [10] UN/CEFACT and OASIS, Business Process Specification Version 1.01, <http://www.ebxml.org/specs/ebBPSS.pdf>, 2001.
- [11] UN/CEFACT and OASIS, Collaboration- Protocol Profile and Agreement Specification Version 1.0, <http://www.ebxml.org/specs/eb CPP.pdf>, 2001.
- [12] UN/CEFACT and OASIS, Message Service Specification v2.0 ebXML Transport, Routing and Packaging, <http://www.ebxml.org/specs/ebMS.pdf>, 2002.
- [13] W3C, XML-Signature Syntax and Processing, <http://www.w3.org/TR/xmlsig-core>
- [14] W3C, SOAP Security Extensions: Digital Signature, <http://www.w3.org/TR/SOAP-dsig>
- [15] W3C, Simple Object Access Protocol(SOAP), Version 1.1, W3C Technical Reports, www.w3.org/TR/SOAP, 2000.
- [16] IETF RFC 2392, Content-ID and Message-ID Uniform Resource Locators. E.Levinson, Published August 1998.
- [17] 전희영, 김계용, 유정연, 이규철, "ebXML 등록기/저장소에서의 객체 질의 관리 시스템", 한국정보과학회학회지, Vol.28, Num. 2, p.37-39.



송 은 정

1999년 충남대학교 컴퓨터공학과(학사)
2003년 충남대학교 컴퓨터공학과 대학원(석사)
관심분야는 멀티미디어통신, 전자상거래, 시스템 소프트웨어



이 호 송

2002년 충남대학교 컴퓨터공학과(학사)
현재 충남대학교 컴퓨터공학과 대학원(석사).
관심분야는 멀티미디어통신, 전자상거래, 임베디드 시스템



권 택 근

1988년 서울대학교 컴퓨터공학과(학사)
1990년 서울대학교 컴퓨터공학과 대학원(석사).
1996년 서울대학교 컴퓨터공학과 대학원(박사). 현재 충남대학교 컴퓨터공학과 교수.
관심분야는 멀티미디어통신, 전자상거래, 초고속 통신망 ATM