

SOAP 기반의 메시지 Broker 시스템 설계 및 구현

김용수[†] · 주경수^{**}

요 약

SOAP(Simple Object Access Protocol)은 분산 환경에서의 정보 교환에 사용되는 분산 컴퓨팅 프로토콜로 분산 시스템간에 메시지를 전달하는 방법, 원격 프로시저 호출/응답을 처리하는 방법을 정의하고 있다. 이러한 SOAP는 텍스트 기반 XML을 프로토콜로 사용하고 있기 때문에 하드웨어 플랫폼, 운영체제, 프로그래밍 언어에 독립적으로 사용할 수 있다. 메시지 Broker 시스템은 SOAP을 이용하여 기업과 기업간의 메시지 전송을 담당하게 된다. 본 논문에서 SOAP을 기반으로 한 메시지 Broker 시스템을 설계 및 구현하였다. Broker 시스템을 통한 메시지 전송으로 많은 비즈니스 파트너를 통합관리 할 수 있을 것이다.

Designing and Implementing Message Broker System based on SOAP

Kim Yong Soo[†] and Joo Kyung Soo^{**}

ABSTRACT

SOAP is a protocol used for information exchange in distributed computing environment and SOAP defines mechanisms for message transferring between distributed system and remote procedure call/response process. This SOAP is able to be used independently on operating systems and program languages because it uses only a text-based XML for protocol. Therefore ebXML that is a standard for E-Commerce adopts SOAP for message exchange. In this paper, we developed Message Broker System based on SOAP. By this Message Broker System we can more easily exchange messages for B2B. Accordingly we can manage and integrate a lots of business partners.

Key words: XML, SOAP, XML Broker, 전자상거래

1. 서 론

현재 인터넷은 급속히 발전하였으며 데이터 중심의 마크업 언어인 XML이 기업과 기업간의(B2B) 전자상거래에서 표준으로 자리를 잡아가고 있다. 즉 B2B에서는 수많은 문서를 XML로 작성하고 이를 기업간에 주고받으려는 노력을 하고있다. 대기업의 경우에는 자체적으로 B2B를 위한 인프라를 구축하는데 어려움이 없으며 대기업이 원하는 XML 문서의 표준을 중소기업들이 따라가는 형편이다. 그러나 중소기업들 간의 거래에 있어서는 중간에서 XML 메시

지의 전송을 담당하고, XML 문서의 표준을 정해주는 중계자가 필요하다. 따라서 이러한 역할을 메시지 Broker가 수행하게 된다.

인터넷을 통해 기업과 기업, 기업과 고객의 장벽이 없어진 상황에서 서로 다른 인프라를 바탕으로 하는 시스템을 연결하는 방법이 절실한 현실이다. 그러나 JAVA, CORBA, COM 등의 주요 분산 컴퓨팅 기술은 독자적인 인프라와 기술을 사용하므로 상호 운용 성이 크게 떨어지게 마련이다. 즉, 현재의 분산 컴퓨팅을 구현하는 JAVA, CORBA, COM 등의 목적은 비슷하지만 목적을 구현하는 방법은 매우 다르므로 호환성을 기대하기 어렵다.

SOAP(Simple Object Access Protocol)은 분산 환경에서의 정보 교환에 사용되는 분산 컴퓨팅 프로토콜로 분산 시스템간에 메시지를 전달하는 방법, 원

접수일 : 2002년 9월 26일, 완료일 : 2003년 3월 18일

[†] 정회원, (주)아이티포웍 근무

^{**} 순천향대학교 정보기술공학부 교수

격 프로시저 호출/응답을 처리하는 방법을 정의하고 있다. 이러한 SOAP은 텍스트 기반 XML을 프로토콜로 사용하고 있기 때문에 운영체제, 프로그래밍 언어에 독립적으로 사용할 수 있다. 따라서 전자상거래 표준인 ebXML에서도 메시지 전송을 위해 SOAP을 사용하고 있다[6,8].

본 논문에서 SOAP을 기반으로 한 메시지 Broker 시스템을 구현하였다. Broker 시스템을 통한 메시지 전송으로 많은 비즈니스 파트너를 통합관리 할 수 있을 것이다. 2장에서는 관련연구와 관련기술에 대해 알아보고, 3장에서는 Broker 시스템에 대해 설명하고, 4장에서는 Broker 시스템을 분석하고 설계한다. 5장에는 구현된 테스트 결과를 보여 주며, 6장에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련 연구 및 기술

2.1 관련 연구

현재까지 XML을 이용한 메시지 교환 시스템은 기존의 전통적인 EDI(Electronic Data Interchange) 시스템을 어떻게 XML로 메시지를 교환할 것인가에 초점이 맞추어져 있다. EDI/XML 시스템의 성공적인 구축은 데이터 교환 모델을 위하여 XML을 사용하고, 모습을 표현하기 위하여 XSL(XML Style Language)을 이용하며, 전통적인 EDI와 쉬운 통합 방안을 지니기 위하여 DTD를 사용한다. 또한 문서 중심의 조회와 처리가 가능케 하고, 타 정보 시스템과의 연동이 가능하도록 개발하는 것이다. 이러한 문제를 해결하기 위하여 전자상거래 표준 기술로서 XML이 등장하게 되었으며 광범위한 영역에서 입지

를 확고히 해 나가고 있다[9,10].

표 1은 각각 다른 방향으로 e비즈니스를 위해 표준화 작업을 하고 있는 전자상거래(EC) 프레임워크 비교표이다. 표와 같이 모든 프레임워크들은 XML 기반으로 움직이고 있다는 것을 알 수 있다. 또한 통신 수단으로 웹서비스, Biztalk, 그리고 최근에는 ebXML에서도 HTTP 위에서 SOAP 기반으로 메시지를 전송 하고있다[1,2,13].

2.1.2 XML 메시지 Broker

고객과 공급자간, 비즈니스 파트너간의 인터넷을 통한 XML 문서의 교환은 네트워크를 통한 경제활동의 중요한 요소이다. Tamino X-Bridge는 엔터프라이즈급의 XML 기반 B2B 정보교환을 위한 중앙의 연결 Broker 이다. XML 문서는 사용자가 미리 정의한 규칙에 따라 그 내용과 구조가 분석되어, 사람의 개입 없이도 해당 문서를 처리할 어플리케이션에 자동으로 전달된다. 또한, 필요한 경우 XML 문서를 수신 어플리케이션이 이해할 수 있는 포맷으로 변환할 수 있다. 다음은 Tamino X-Bridge의 기능이다.

① 라우팅 테이블을 통한 XML 데이터의 중개 : 비즈니스 파트너로부터 수신된 XML 문서는 Tamino X-Bridge에 지정된 콘텐츠/구조 기반의 라우팅 규칙에 따라 처리할 어플리케이션이 결정된 뒤 전달 된다.

② 문서 변환 : Tamino X-Bridge는 송신할 문서나 수신된 XML 문서를 처리할 어플리케이션이 요구하는 포맷으로 변환해 줄 수 있다. 문서 변환은 XSLT(eXtensible Stylesheet Language Transformations) 표준에 기반하여 이루어진다.

표 1. 전자상거래(EC) 프레임워크 비교표

	ebXML	eCo	UDDI	Rosettanet	Biztalk
Industry Target	Unspecified	Unspecified	Unspecified	IT & Electronic	Unspecified
Communication protocol	HTTP, SMTP...	HTTP	HTTP(SOAP)	HTTP/CGI	HTTP(SOAP)/MSMQ
Repository	Distributed	Locally maintained	Distributed	Not addressed	Centralized repository
Message format	XML Document	XML Document	XML Document	XML Document	Biztalk document based on Biztag
Security	S/MIME, Digital Signature	Optional	Authentication in each UDDI Registry	SSL with HTTP; digital certificate and signature	Leverages existing standards
Ontology	Common Business Objects	Common Business Library	NAICS, UN/SPSC	Technical and Business Dictionaries	Collection of Biztag

③ 인터넷과 XML을 위해 고안 : Tamino X-Bridge는 인터넷 표준 프로토콜인 HTTP를 지원하므로 HTTP를 지원하는 어떤 어플리케이션에서도 어드레싱이 가능하다. 새로운 비즈니스 파트너나 어플리케이션은 Tamino X-Bridge의 인터넷 어드레스(URL)만 제공함으로써 즉시 추가할 수 있다.



그림 1. Tamino X-Bridge 구조도

Tamino X-Bridge는 위와 같은 기능을 가지고 있으며 그림 1과 같은 구조를 가지고 있다.

본 논문에서 구현한 메시지 Broker 시스템 또한, XML 메시지의 전송을 주된 기능으로 하고 있다. Tamino X-Bridge와 본 논문에서 구현한 시스템과의 차이는 SOAP을 기반으로 하였다는 것이다. Tamino X-Bridge는 메시지를 단순히 HTTP를 이용해 전송하고 있다. 그러나 본 논문에서 구현한 메시지 Broker는 SOAP을 기반으로 하기 때문에 많은 거래 파트너의 다양한 어플리케이션에 접근이 가능하다.

2.2 관련기술

SOAP은 XML과 HTTP를 사용해 플랫폼에 독립적으로 서비스 혹은 분산 객체를 액세스하는 방법을 정의한다. HTTP는 인터넷 표준이며 누구나 어떠한 플랫폼에서도 사용할 수 있는 프로토콜이다. 그 어떤 개인이나 기업도 HTTP를 인정한다. XML은 HTTP보다 상대적으로 늦게 나온 기술이지만 최근 들어 널리 사용되며 업계 표준으로 자리잡고 있다. 또한 HTTP와 XML은 공통적으로 텍스트에 기반하고 있으므로 이들을 처리하는 소요 비용(프로세싱 시간, 메모리 등)은 상대적으로 매우 적다. 텍스트 문자열을 제어하고 TCP/IP에 접근할 수 있는 어떠한 응용 프로그램도 HTTP를 통해 XML 데이터를 전송하거나 수신할 수 있다는 것이다. 이러한 손쉬운 사용성은 이들 두 기술이 빠르게 정보기술 환경에 적응하는 발판이 되었다[2-5].

SOAP은 이런 XML과 HTTP를 사용함으로써 이들이 갖는 장점을 모두 포함하면서 시스템의 상호운용성을 높일 수 있는 것이다

2.2.1 SOAP의 독립성

SOAP은 XML과 HTTP를 이용해 어떤 인터페이스의 메소드를 호출할 것이며, 이 메소드에 대한 매개변수를 알리는 역할만 담당한다. 결론적으로 말해, SOAP은 JAVA, CORBA, COM 등의 분산 객체 기술에 구애받지 않는 프로토콜이며 심지어 이들 분산 객체 기술을 전혀 사용하지 않는 애플리케이션도 SOAP을 통해 원격 프로시저 호출이나 데이터 전송을 수행할 수 있다. SOAP이 원격 객체의 구현에 대해 언급하고 있지 않으므로, 원격 객체는 어떤 프로그래밍 언어로도 구현할 수 있다[4].

2.2.2 인터넷 적용성

SOAP은 transport로서 HTTP를 사용한다. HTTP를 사용함으로써 얻을 수 있는 장점은 인터넷에서 널리 사용할 수 있다는 점이다. 실제로 SOAP은 인터넷에서 원격 객체를 액세스하기 위해 고안된 프로토콜이다. 그렇다면 기존 객체 지향 기술은 인터넷에서 문제가 있었던 것인가 살펴보도록 하자. JAVA, CORBA, COM 등이 사용하는 transport는 고유의 transport로서 RMI, HOP, DCOM을 사용하지만, 이들 프로토콜은 TCP/IP 포트를 동적 할당하는 메커니즘을 사용하고 있다. 이것 때문에 이들 프로토콜이 방화벽을 통과하는데 문제점이 드러난다[4].

예를 들어 DCOM을 인터넷에서 사용하려면 방화벽에 일정 영역의 TCP/IP 포트를 열어 놓아야 하는데, 이것은 방화벽을 사용하지 않는 것과 다름없게 만들어 버린다. 반면 대부분의 기업 네트워크는 자사의 웹 사이트를 위해 HTTP가 사용하는 포트를 열어 놓고 있다. SOAP은 HTTP를 사용하므로 아무런 문제 없이 방화벽을 통과할 수 있다. 또한 대부분의 방화벽 제품은 HTTP 헤더의 내용을 읽어 필터링을 수행할 수 있으므로 특정 인터페이스의 특정 메소드를 호출하는 SOAP 메시지를 통과시키도록 설정할 수도 있다. SOAP은 HTTP를 통해 인터넷에 분산된 객체들에 접근할 수 있으며 방화벽이나 HTTPS 등의 인터넷 보안 기술을 그대로 적용받을 수 있는 것이다.

3. SOAP 기반의 Broker 시스템

3.1 SOAP 기반 XML 메시지 구조

SOAP 메시지는 두 개의 데이터 구조인 SOAP 머리글(header)과 SOAP 본문(body)을 둘러싸는 SOAP 봉투(envelope)로 구성된다. 봉투는 머리글 및 본문을 포함하는 것 외에도 이러한 데이터 구조의 정의에 사용되는 이름 공간을 정의하는 정보를 포함한다. 머리글은 생략할 수 있지만 SOAP 본문에 정의된 요청에 대한 정보를 전달하려는 경우에 사용된다. 예를 들어, 머리글은 트랜잭션, 보안, 컨텍스트 또는 사용자 프로필 정보를 포함할 수 있다. 본문에는 웹 서비스의 요청이나 요청에 대한 응답 XML 형식으로 포함된다. 그림 2는 SOAP 메시지의 상위 레벨 구조를 나타낸 것이다.

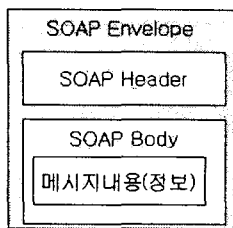


그림 2. SOAP 메시지 구조

3.2 Broker 시스템 구조

메시지 Broker 시스템은 Broker 시스템을 통해 다른 클라이언트에 메시지를 전송할 수 있도록 한다. 그림 3은 메시지 Broker 시스템의 구조를 보여준다. 클라이언트 A가 클라이언트 B에 XML 문서를 전송하기 위해 Broker 시스템에 전송하면 Broker 시스템은 이를 저장하고 클라이언트 B에 XML 문서를 재전송한다. 전송의 응답이 다시 Broker 시스템을 통해

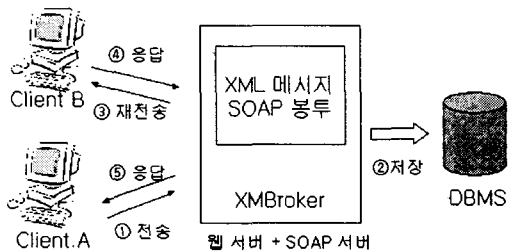


그림 3. Broker 시스템 구조

클라이언트 A에 전달된다. 이와 같은 방법은 많은 거래 파트너를 클라이언트가 모두 관리해야 하는 일을 줄일 수 있다. 또한 서로 다른 XML 문서의 형태를 가지고 있을 경우에 Broker 시스템을 통해 적절한 형태로 변형되어 전송할 수 있다.

3.3 Broker 시스템의 기능

메시지 Broker 시스템에서 서버 기능을 수행하며, 받은 메시지에 대하여 헤더처리, 암호화/복호화 및 보안, 예외처리, 라우팅, 호출 5가지 기능을 수행한다[8,11].

3.3.1 헤더처리

메시지를 수신했을 때 메시지 브로커에서 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.

3.3.2 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.

3.3.3 오류와 예외처리

오류와 예외처리는 메시지 브로커가 수행하는 중요한 기능 중 하나다. 클라이언트가 수신한 메시지가 유효하지 않거나, 요청을 수행할 수 없는 경우에는 예외 메시지를 클라이언트에 보내야만 한다.

3.3.4 라우팅

메시지 라우팅은 두 단계로 이루어진다. 하나는 헤더 라우팅으로, 수신된 메시지가 어느 응용 프로그램에서 처리돼야 할지를 결정한다. 다른 하나는 페이

로드 라우팅으로 해당 응용프로그램에서 어떤 프로세스나 메소드가 사용돼야 할 지를 결정한다.

3.3.5 호출

호출 단계에서는 실제 수신 메시지에 있는 페이로드의 자료를 가지고 메소드를 호출하게 된다. 여기서는 메소드 호출을 통해 브로커에서 클라이언트로 반환할 수행 결과가 만들어질 수도 있다.

4. Broker 시스템 분석 및 설계

4.1 Use Case 다이어그램

그림 4는 유스케이스 다이어그램을 나타냈다. 사용자는 XML 문서를 작성하고, 작성된 문서를 토대로 SOAP 봉투 처리를 하고 이렇게 생성된 XML 문서를 Broker 시스템에 전송하여 어떠한 처리를 요구한다. XML 문서 처리를 하기 위해서는 XML 문서가 있어야만 가능하기 때문에, XML 처리 유스케이스는 생성 유스케이스를 반드시 필요로 하는 관계가 <<include>>로 표현된다[7,12].

그림 5는 클라이언트가 XML 문서를 생성하여 Broker에 전송하고, Broker는 다시 클라이언트에 XML 문서를 전송하여 처리 결과를 응답하는 순차 다이어그램을 나타낸 것이다. 클라이언트는 사용자 인터페이스를 통하여 요청 서비스를 포함하는 XML 문서를 작성하고 XMLMessage를 통하여 SOAP 봉투를 작성한다. 이렇게 작성된 문서를 사용자가 HTTP를 통해 문서를 전송한다. 그리고 Broker에서는 다시 XML 문서를 다른 클라이언트에 재전송하고 클라이언트로부터 관련된 처리를 요청한다. 최종 전송된 클라이언트에서 HTTP 요청을 받아서, 이를 처리할 MessageBroker에 전달한다. MessageBroker는 먼저 헤더를 분석해서 클라이언트가 요청한 서비

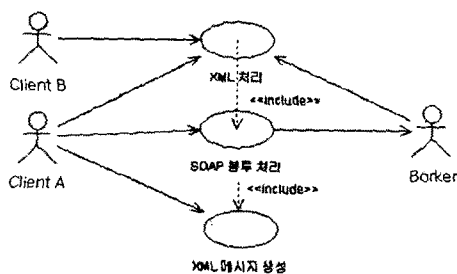


그림 4. 유스케이스 다이어그램

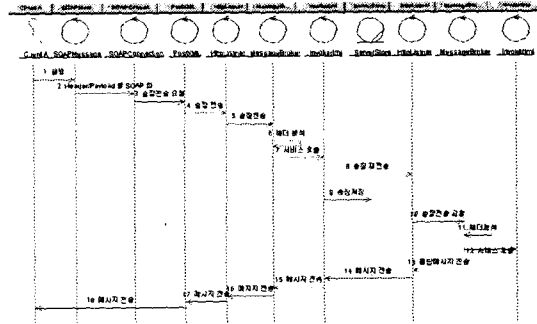


그림 5. 순차 다이어그램

스를 확인한다. 그 다음에 해당되는 서비스를 호출하는데, 현재는 송장의 저장만을 대상으로 하므로 InvokerImpl가 호출되고, 이 클래스는 저장소에 저장한 후, 처리가 성공했음을 알리는 XML 메시지를 만든다. 작성된 메시지는 Broker 시스템에 성공/실패 메시지를 전달하고 Broker 시스템은 다시 최초 클라이언트에 성공/실패 메시지를 전달한다.

4.2 메시지의 구조

그림 6은 본 논문에서 사용될 XML 메시지의 구조를 나타낸다. 이 XML 문서는 헤더 부분과 바디 부분으로 나뉘어져 있다. 헤더 부분은 전달할 목적지와 발신자의 정보를 저장하고 있으며, 바디 부분에는 송장에 관한 여러 가지 정보를 입력하였다. XML 메시지에서 정보를 저장하는 부분이 바디 부분이므로 이곳에 주문 내역에 대한 정보를 입력하였다.

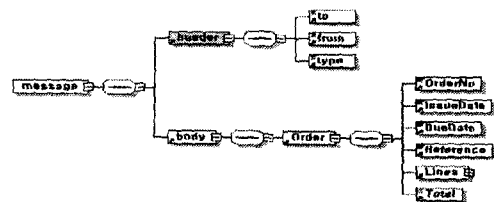


그림 6. 메시지의 구조

4.3 Broker 시스템

Broker 시스템의 처리과정은 그림 7, 그림 8, 그림 9와 같다. 클라이언트 측에서는 사용자가 작성한 XML 문서를 SOAPHeader와 SOAPPayload를 이용하여 헤더와 페이로드를 작성한다. 또한 이렇게 Header와 Payload를 작성한 후 SOAPMessage를 이용하여 Header와 Payload를 SOAP 봉투화 한다. 이

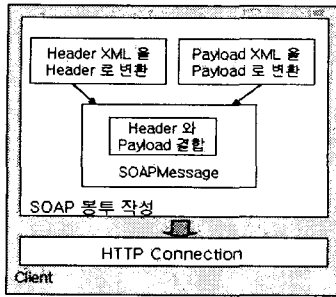


그림 7. 클라이언트 측 메시지 처리 과정

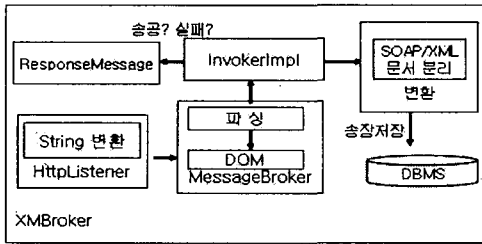


그림 8. Broker에서 XML 메시지 처리과정

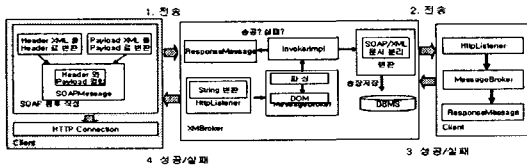


그림 9. 메시지 전달 과정

렇게 작성된 메시지를 HTTP를 이용하여 전송한다. 이렇게 클라이언트가 메시지를 만들어서 Broker에 보내게 되면, Broker는 다시 최종 수신하는 클라이언트에 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 ResponseMessage가 있다.

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 HTTP를 이용

하여 전송하게 된다. 전체 메시지를 구성하는 각각의 메시지의 특성들을 Content-Type과 Type, Boudner, Version 등을 통하여 설정할 수 있도록 구성한다.

4.3.3 HttpListener

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

4.3.4 MessageBroker

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

4.3.5 InvokerImpl

Invoker 인터페이스를 구현한 것인 InvokerImpl 클래스이다. 여기에서는 파일 시스템에 송장을 저장하고, ResponseMessageMaker에서 응답 메시지를 만든다. ResponseMessage Manager는 요청이 성공적으로 이루어 졌을 때는 클라이언트에 보낼 '성공' 메시지나, 실패했을 경우의 '에러' 메시지를 XML 문서로 만든다.

5. Broker 시스템 구현

Broker 시스템의 환경으로 운영체제는 Windows 2000 Server, 웹 서버로는 Jakarta Tomcat V3.1.3을 사용하였으며, 아파치 SOAP2.2, XML Parser V2를 이용하여 구축하였다. 다음 그림 10은 XML 문를 이용하여 작성된 송장을 나타낸다. 송장에서 헤더 부분과 페이로드 부분으로 나뉘는 것을 볼 수 있을 것이다. OrderNo를 이용하여 각 송장에 일련번호를 부여하였으며, 송장은 Broker 시스템에 전달되어 저장 시스템에 저장되고 최종 클라이언트에 다시 전송된다.

그림 11은 최종 클라이언트가 메시지를 받을 때 나타나는 화면이다. 메시지 전송이 시작되면 Servlet 이 시작되고, 전송이 완료되면 Servlet이 끝나며, 전송 결과는 Broker 시스템에 전달되어 그림 12와 같은 전송 내용과 결과를 웹서버의 실행 창에 보여 주게 된다.

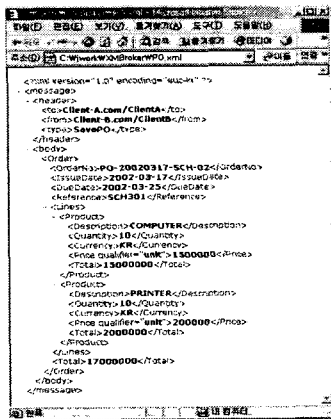


그림 10. sample XML 문서

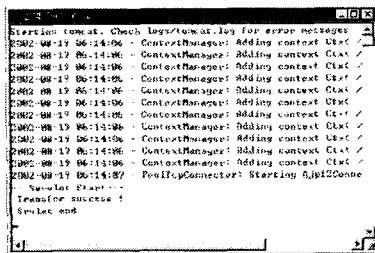


그림 11. 최종 Client 실행화면

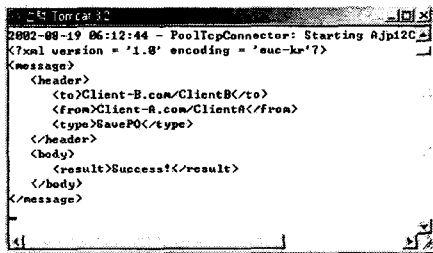


그림 12. Broker 시스템 실행화면

5.1 비교 및 고찰

Broker 시스템은 전자상거래 표준인 XML 문서를 전송 중계기의 역할을 수행하게 된다. 물론 현재 이와 같은 서비스를 지원하는 시스템이 많이 있다. 그러나 본 논문에서는 각 기업들이 가지고 있는 다양한 어플리케이션에 확장하여 사용할 수 있도록 SOAP을 기반으로 하였다.

표 2에서 본문에서 개발한 Broker 시스템과 Tamino X-Bridge를 비교하였다. 표에서 보는 바와 같이 Tamino X-Bridge 또한 많은 기능을 가지고 있으며 본 논문에서 구현한 Broker 시스템보다도 더 많은 기능을 가지고 있는 것이 사실이다. 그러나 본

표 2. 기능 비교 표

기능	Broker 시스템	Tamino X-Bridge
구현 언어	JAVA	JAVA
전송 대상	XML 문서	XML 문서
전송 방법	HTTP	HTTP
라우팅 기능	구현	구현
호출 기능	SOAP을 이용해 모든 어플리케이션 가능	자바 어플리케이션만 가능
문서변환 기능	기능 없음	가능

논문에서는 Tamino에서 구현되지 않은 SOAP 방식을 이용한 XML 메시지 전송을 이용하고 있기 때문에 기존 어플리케이션의 환경에 상관없이 서비스 확장이 가능하다는 장점을 가지고 있다.

5.2 Broker 시스템의 활용 및 효과

전자상거래의 초기에는 B2C가 시장의 대부분을 형성하였다. 그러나 B2C의 시장보다는 B2B의 시장이 크고 앞으로 모든 기업은 B2B를 실현하게 된다. 그렇게 되면 대기업들은 수많은 하청 기업들을 상대로 B2B를 하게 된다. 그러나 중소기업들간의 거래는 이들 스스로 B2B를 실현하기에는 너무 많은 노력과 비용이 소비된다. 그렇기 때문에 이 기업들을 중계해 주는 시스템이 필요 한 것이다. 중소기업들이 전자상거래를 통한 B2B를 실현하게 될 경우 오프라인에 존재하던 중간 Broker들은 모두 사라지고 이것은 결국 원가 절감과 실시간으로 이루어지는 거래로 생산기간 단축을 이룰 수 있게 된다. 이것은 결국 기업의 경쟁력을 키우는 효과를 가져올 것이다. 본 논문에서는 B2B를 위한 XML 메시지 Broker를 SOAP 기반으로 함으로써 기업들의 기존 어플리케이션에 독립적으로 Broker 시스템에 접근 할 수 있는 장점이 있다. 관련 연구에서 설명한 Tamino X-Bridge의 경우에도 HTTP를 통해 메시지를 전송하고 있지만 SOAP 형태의 메시지를 보내는 것은 아니다.

6. 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 XML이다. 데이터를 XML로 표현하고 전송 수단을 HTTP 환경의 SOAP을 이용한다면 지금 까지 가지고 있던 여러 가지 문제를 한번에 해결할 수 있는 것이다.

본 논문에서는 현재 다양한 방향으로 연구가 진행 중인 XML 문서 교환용 Broker 시스템을 설계 및 구현하였다. 이 Broker 시스템에서 전송되는 메시지의 형태는 SOAP에서 정의한 메시지 봉투를 적용하였다. 이를 이용하면 기업간의 문서를 전달하는데 좀더 효율적인 방법을 제공한다. 향후 연구방향으로는 Broker 시스템을 좀더 발전시켜 HUB 시스템을 구현하고자 한다. XML HUB 시스템은 XML 메시지 전달, 메시지 변환, 저장, 비동기전송 등을 지원하며, 기업간의 메시지 전송에서 가장 핵심이 되는 부분으로 발전할 것이다.

참 고 문 헌

[1] Microsoft, "Biztalk Framework 1.0 Independent Document Specification" <http://www.biztalk.org>

[2] Extensible Markup Language(XML)1.0, <http://www.w3.org/TR/1998/REC-xml-19980210>

[3] Liam Quin, Open Source XML Database Toolkit, WILEY, 2000.

[4] Simple Object Access Protocol(SOAP)1.1, <http://www.w3.org/TR/SOAP>

[5] Simple Object Access Protocol(SOAP), http://www.microsoft.com/korea/msdn/workshop/xml/general/SOAP_White_Paper.asp

[6] Apache SOAP Documentation: User's Guide <http://xml.apache.org/soap/docs/index.html>

[7] DAVID CARLSON, MODELING XML APPLICATIONS WITH UML, Wesley, 2001

[8] Bequet, Henry. Professional Java SOAP, Wrox, 2001.

[9] 신동규, "XML/EDI 시스템의 설계 및 구현", 한국정보처리학회 논문지 8-D권 제 2호, 2001.

[10] 이태웅, "전자상거래 표준화 기술로서의 XML", 추계국제학술대회 논문집, Vol.0, No.0, 1999.

[11] 김채미, 전문가와 함께가는 XML Camp, 아이트Press, 2001.

[12] 조완수, UML 객체지향 분석·설계, 홍릉과학출판사, 2000.

[13] 김채미, 최학열, 글로벌 e비즈니스 리더를 위한 ebXML, 대청미디어, 2001.



김 용 수

2001년 순천향대학교 전산학과 졸업(학사)
 2003년 순천향대학교 일반대학원 전산학과 졸업(공학석사)
 2003년~현재 (주)아이티포웹 근무
 관심분야 : Database Systems, EJB, 전자상거래, B2B

E-mail : yskim@it4web.com



주 경 수

1980년 고려대학교 이과대학 수학과 졸업(학사)
 1985년 고려대학교 일반대학원 전산학과 졸업(석사)
 1993년 고려대학교 일반대학원 전산학과 졸업(박사)
 1986년~현재 순천향대학교 정

보기술공학부 교수

관심분야 : Database Systems, System Integration, Object-oriented Systems.

E-mail : gsoojoo@sch.ac.kr

교 신 저 자

주 경 수 336-745 충남 아산시 신창면 읍내리 646 순천향대학교 멀티미디어관 M601