
SOAP 기반의 상호작용 상품 주문 및 가격비교 웹 서비스 시스템 설계 및 구현

김철원* · 박종훈**

Design and Implementation of Web Service System based on SOAP for Interactive Product Order and Price Comparison

Chul-Won Kim* · Jong-Hoon Park**

요 약

현재 가격비교 사이트에서는 관련 쇼핑몰 회사들이 가격비교 사이트에 가입하여 수동으로 상품정보와 가격을 입력하여야 하며, 또한 상품가격을 자동으로 실시간 가격변동을 지원할 수 없는 문제점이 있다. 따라서 본 논문에서는 웹 서비스 기술을 이용하여 실시간의 능동적인 상품 가격비교 및 직접 상품 주문에 대한 SOAP기반의 동적 상호작용 웹 서비스 시스템을 설계 및 구현하였다. 본 연구의 웹 서비스 시스템은 크게 상품검색, 상품정렬과 상품주문을 지원하는 웹 서비스 클라이언트 모듈과 상품검색과 상품주문 원격 프로시저 호출을 포함하는 서버모듈로 나누어진다. 본 웹 서비스 시스템은 SOAP 메시지 기반으로 상호 정보를 교환하도록 구현하였으며, 사실표준 환경인 SOAP, WSDL, UDDI 표준을 따르는 모든 환경에서 이식가능하고, 플랫폼에 독립적으로 지원할 수 있다.

ABSTRACT

Currently, Price comparison sites receive inputs of product information and price manually after being joined by related shopping mall companies, and can't support automatically price fluctuation of products in real time. Therefore, this paper designs and implements web service system based on SOAP for interactive product order and price comparison dynamically using web service technologies in real time. This web service system composes web service client module including functions of product search, sort and order, and server module including functions of remote procedure call for product search and order. This web service system implements interchanging product information based on SOAP messages and can support independence of platform and flexible portability in environment conforming to SOAP, WSDL and UDDI standards.

키워드

web service, SOAP, WSDL, UDDI, standard

1. 서 론

컴포넌트 기반의 분산 컴퓨팅에 사용되는 대표적인 프로토콜로서는 OMG(object Management

*호남대학교

접수일자 : 2004. 7. 16

**중부대학교

Group)의 CORBA(Common Object Request Broker Architecture) 표준 프로토콜인 IIOP와 마이크로소프트의 DCOM(Distributed Component Object Model), 그리고 Sun사의 RMI(Remote Method Invocation)프로토콜이 있다. 이러한 연결 지향적인 모델의 문제점은 유연성과 확장성에 극히 제한적이다. 이들 프로토콜을 통합시키는 방법에 대해서는 많은 시도가 있었지만, 널리 활용되지 못하였다. 초창기 처음부터 같은 프로토콜 환경에서 시스템을 설계할 경우는 문제되지 않지만, 비즈니스 사업규모가 커지고 기업간의 전자상거래가 활발해짐에 따라 서로 다른 시스템간의 연동이 중요하게 되었다. 즉 적은 비용으로 다른 시스템을 통합시키는 분산 컴포넌트 기반의 컴퓨팅 기술인 웹 서비스 기술이 필요하다. 사용하는 프로토콜은 XML 기반의 SOAP(Simple Object Access Protocol)이고, 웹 서비스와 클라이언트간의 프로토콜인 SOAP는 플랫폼 중립적인 프로토콜을 말한다. XML 문서를 인식하고 처리할 수 있는 모든 플랫폼에서 SOAP를 이용할 수 있기 때문에 웹 서비스와 클라이언트의 개별적인 개발이 가능하다[1,2,3,4]. 따라서 웹 서비스는 다양한 확장성과 잠재성을 갖고 있는 기술이기 때문에 현재 다양한 분산환경 분야에서 웹 서비스 기술을 연구하고 있다. 현재 가격비교 사이트에서는 관련 쇼핑몰 회사들이 가격비교 사이트에 가입하여 수동으로 상품정보와 가격을 입력하여야 하며, 또한 상품가격을 자동으로 실시간 가격변동을 지원할 수 없는 문제점이 있다. 그러나 이러한 문제점을 해결하기 위해 지금까지 가격비교 사이트에 일일이 자사의 상품정보를 등록했던 수동적인 작업 대신에 반대로 가격비교 사이트에서 자사의 웹 서비스 시스템을 이용해서 상품정보를 실시간으로 받아가도록 설계하는 것이 필요하다. 또한 가격비교 사이트에서 상품을 선택하여 직접 주문할 수 있도록 기능 추가와 UDDI 레지스트리에서 웹 서비스 쇼핑몰 정보를 찾아서 원하는 정보를 언제든지 가져와 가격비교 사이트의 내용과 바인딩시키는 것이 필요하다. 이와 같은 환경을 구축하기 위해, 본 논문에서는 웹 서비스 기술을 이용하여 실시간의 능동적인 상품 가격비교 및 직접 상품 주문에 대한 SOAP기반의 동적 상호작용 웹 서비스 시스템을 설계 및 구현하였다. 본 연구의 웹 서비스 시스템은 크게 상품검색, 상품정렬과 상품주문을 지원하는 웹 서비스 클라이언트 모듈과 상품검색과 상품주문 원격 프로시저 호출을 포함하는 서버모듈로 나누어지며, SOAP 표준 환경기반으로 상호 정보를 교환하도록 구현하였다. 따라서 사실표준 환경인 SOAP, WSD

L, UDDI 표준을 따르는 모든 환경에서 이식가능하며, 플랫폼에 독립적으로 지원할 수 있다.

II. 관련기술

2.1 웹 서비스 개념

웹 서비스에 대한 정의를 표준 기관(W3C)과 업체(IBM, SUN, Microsoft)들의 정의를 종합하여 설명하면 웹 서비스란 XML을 기반으로 하는 공개 표준들을 이용해서 인터넷 기반의 분산 컴퓨팅 환경에서 원격 메소드를 호출하여 사용할 수 있는 재사용 가능한 소프트웨어 컴포넌트를 일컫는다. 웹 서비스를 이용하면 웹을 통한 원격 프로시저 호출(RPC : Remote Procedure Call)이 가능해진다. 웹 서비스는 자기 기술적이며, 퍼블리싱이 가능하고, 웹이나 개방 인터넷 표준을 기반으로 하는 네트워크라면 어떤 곳에서도 설치할 수 있으며, 호출도 가능한 모듈 응용 프로그램이다. 웹 서비스는 컴포넌트 기반 프로그래밍과 웹 프로그래밍의 장점만을 취해서 만들어졌으며, 모듈형태로 패키징되어 재사용할 수 있고, 이때 웹 서비스의 사용자측과 제공자측 양쪽 각각은 구현원리나 웹 서비스 구현하는데 필요한 언어와 운영체제 또는 컴포넌트 모델에는 상호간에 독립적이다. 웹 서비스는 HTTP나 SMTP와 같이 표준 인터넷 프로토콜을 통해서 접근할 수 있으며, XML 기반으로 되어있다 [1,2,3,4].

한개의 웹 서비스를 구성하는데 세가지 역할이 있다. 이들의 역할과 각각의 관련 표준 스펙을 그림으로 도식화하면 다음과 같다.

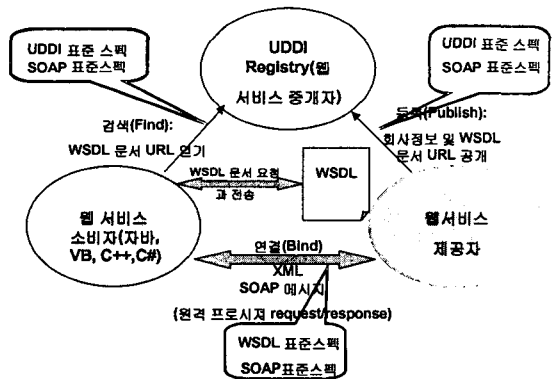


그림 1. 웹 서비스 역할과 관련 표준 스펙
Fig. 1 Role of web service and related standard spec.

위 그림1의 동작들은 세가지 역할의 환경과는 무관하게 플랫폼 독립적으로 통신하는 것을 원칙으로 하고 있기 때문에 통신규약과 같은 약속된 방법으로 진행되어야 한다. 즉 웹 서비스 제공자가 어떠한 플랫폼, 어떠한 프로그램 언어로 웹 서비스를 구현했다 하더라도, 웹 서비스 소비자는 일관된 방법으로 바인딩하여 웹 서비스가 제공하는 기능을 이용할 수 있어야 한다. 또한 웹 서비스 중개자가 어떠한 플랫폼, 어떠한 프로그램 언어로 저장소를 구현했다 하더라도 웹 서비스 제공자와 웹 서비스 소비자는 일관된 방법으로 공개 및 검색을 할 수 있어야 한다. 따라서 세가지 역할에 따라 플랫폼 독립적으로 통신할 수 있도록 하기 위해 각각의 규약이 필요하며 그에 대한 각각의 설명은 다음절에 있다.

2.2 SOAP(Simple Object Access Protocol)

SOAP은 원격 응용프로그램 간의 정보를 교환하는데 필요한 구조를 표준화한 것이다. SOAP 1.1 스펙은 2000년 5월 8일 W3C에 노트(Note)로 공식화되었으며, 2003년 현재 SOAP1.2가 Recommendation 상태이다. SOAP의 구조는 XML 문서형태를 가지고 있고, 전송 프로토콜로서 운반되기 위한 방법에 대해서 언급하고 있다[8].

SOAP은 XML 언어를 이용한 분산환경에서의 정보교환을 위한 프로토콜이다. SOAP은 XML로 구성되어 있기 때문에 XML을 이해할 수 있는 모든 시스템은 SOAP을 통해서 통신할 수 있다. 즉 이종의 플랫폼 응용프로그램 간에도 정보를 교환할 수 있다. SOAP은 단순한 정보 전달용 XML 메시지로 구성되기도 하고, 다른 분산 컴퓨팅 프로토콜(CORBA, RMI, DCOM)처럼 원격 프로시저 호출(RPC)을 하는 XML 메시지로 구성될 수도 있다. 이러한 SOAP의 장점은 SOAP의 구성요소들이 다른 분산 컴퓨팅 프로토콜의 내용보다 적기 때문에 프로토콜 자체가 복잡하지 않은 경량의 프로토콜이다. 즉 SOAP은 어떠한 중개계층도 없이 직접통신 한다. 또한 SOAP에서는 텍스트 기반 XML 포맷을 사용한다. 즉 SOAP은 하드웨어 플랫폼, 운영체제, 프로그래밍언어 전영역에 걸쳐서 독립적인 프로토콜이기 때문에 서로간에 SOAP 메시지로 정보를 교환할 수 있다. 다음으로 SOAP 메시지는 인터넷 표준 프로토콜로 전송될 수 있기 때문에 인터넷에 연결되어 있는 어떤 시스템과도 통신이 가능하다. 또한 SOAP은 분산 컴퓨팅 프로토콜을 사용하는 컴포넌트간에 중개자 역할을 함으로서, 일반적으로 서로에게 접근할 수 없었던 것을 가능하게 하는 중요한 기술을 제공한다[14,14]. SOAP 메시

지의 구조는 루트 엘리먼트로 봉투(envelope)가 있으며 이것은 메시지의 컨테이너역할을 하며 필수 요소이다. 그 하위 엘리먼트로는 헤더(header)와 바디(body)가 있다. 헤더는 선택적이며, 클라이언트의 인증, 트랜잭션 관리와 같은 추가 기능을 확장하기 위해 사용한다. 여기에 사용하는 속성인 액터(actor)는 중개자 중 누가 헤더 엔트리를 처리할 것인가를 지정하며, mustUnderstand 속성은 헤더 엔트리를 반드시 처리해야 하는지를 지정한다. 바디는 필수 요소이며, 최종 수신자가 수신하는 정보를 포함한다. 여기에는 RPC와 관련된 메소드와 파라미터를 기술하고 또한, 처리 도중에 발생한 오류정보를 발신자로 보내는 정보도 포함하고 있다.

2.3 WSDL(Web Service Description Language)

WSDL은 IBM, Microsoft 등의 회사들이 웹 서비스 시스템 기능을 명세화시키는 방법을 표준화하기 위해 만들어졌으며, W3C에서 2001년 3월15일에 WSDL 1.1 Note로 처음 공식화되었으며, 2003년 이후 현재는 버전 2.0(Working Draft)로 되어 있다[14,9,14].

웹 서비스 제공자는 웹 서비스의 기능, 즉 프로시저의 이름, 인자의 종류, 반환형의 종류, 전송 프로토콜의 종류, 웹 서비스의 종점(URL) 등을 WSDL 문서로 작성해서 웹 서비스를 소비자가 언제든지 다운로드 받을 수 있도록 웹서버에 저장한다. 그리고 WSDL 문서의 다운로드 URL을 UDDI 레지스트리에 공개해야 한다. 웹 서비스 제공자는 WSDL 문서를 직접 작성할 수도 있지만, 웹 서비스 코드로부터 WSDL 문서를 자동적으로 생성하는 도구를 이용한다. 웹 서비스 소비자인 클라이언트 응용프로그램은 UDDI 레지스트리에서 웹 서비스 명세서인 WSDL 문서를 다운로드 받을 수 있는 URL을 얻은 후 실제로 해당 URL에서 WSDL 문서를 다운로드 받는다. WSDL 문서를 해석한 후 클라이언트 응용프로그램은 웹 서비스 시스템의 사용법을 알게되고, SOAP 메시지를 생성하여, 전송할 수 있게된다.

WSDL은 웹 서비스 시스템에서 제공하는 기능들을 외부에서 이용할 수 있도록 사용법을 알려주는 인터페이스 언어이다. WSDL은 다른 분산 컴퓨팅 기술인 CORBA의 IDL(Interface Definition Language)에 해당한다. IDL은 원격 클라이언트에서 호출 가능한 원격서버 객체의 메소드에 대한 부분을 정의하는 인터페이스 언어로 프로그래밍언어에 독립적이다. 또한 WSDL도 XML로 작성되었기 때문에 프로그래밍언어에 독립적인 인터페이스 언어이다. WSDL 문서는 특정 프로그램언어로 변환

되어 클라이언트 응용프로그램에서 사용된다. WSDL 문서에서 생성된 코드는 웹 서비스 클라이언트 시스템에 바인딩하여 SOAP 메시지를 생성하고, 전송시키는 스텝(stub)역할을 맡게 된다.

WSDL 문서의 구조로 다음 그림2와 같은 요소로 구성되어 있다.

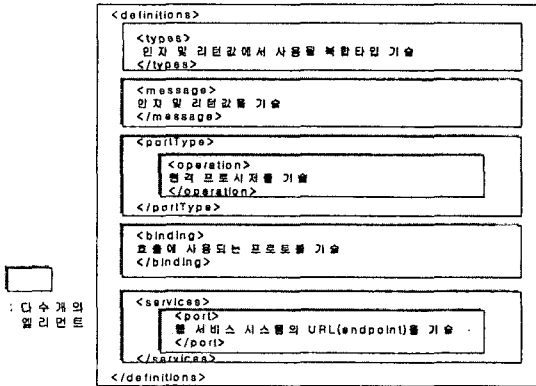


그림 2. WSDL의 문서 구조
Fig. 2 Document structure of WSDL

2.4 UDDI(Universal Description, Discovery and Integration)

UDDI 레지스트리는 각종정보들을 생성, 저장, 검색할 수 있는 XML 기반의 자료저장장치(S/W+H/W)를 말한다. UDDI 레지스트리의 클라이언트가 UDDI 레지스트리에 접근해서 정보를 저장하고, 찾기 위해서는 SOAP 메시지를 이용한다. UDDI 레지스트리는 지역별, 업종별로 회사정보를 분류하여 특정회사를 쉽게 검색할 수 있도록 구성되며, 마치 전화번호부와 비슷한 방식으로 웹 서비스에 대한 정보들을 다음과 같이 분류화시킨다. 화이트페이지에 속하는 정보는 회사의 이름, 주소, 전화번호 그리고 회사에 관한 설명이 포함되며, 옐로우 페이지에 속하는 정보는 산업계의 분류체계(NAICS)별, 생산물과 웹 서비스의 분류체계(UNSPEC)별, 그리고 지역별 회사 목록이 여기에 속한다. 그린페이지에 속하는 정보는 각 회사에서 제공하는 웹 서비스에 대한 기술적 정보이며, 즉 웹 서비스 시스템의 종점 URL, WSDL 문서의 URL 등이 여기에 포함한다. UDDI 레지스트리가 XML 기반의 자료 저장장치이기 때문에 UDDI 레지스트리 개발언어 및 실행 플랫폼과는 상관없이 UDDI 레지스트리 간의 데이터 교환이 자유롭다. 데이터 교환 포맷으로 XML 문서를 사용하기 때문이다. 따라서 한개의 UDDI 레지스트리에 정보를

저장하더라도 복제 관계에 있는 모든 레지스트리에 복제가 되어 저장되기 때문에 웹 서비스 정보가 보다 많은 클라이언트들에게 홍보될 수 있다 [10,11,14].

처음에는 UDDI 레지스트리를 제안한 Microsoft, IBM, Ariba 3개 회사가 UDDI 초안 스펙을 작성하였으나, UDDI 레지스트리에서 정보를 공유하고, 상호검색 할 수 있는 산업표준을 정의하기 위해 UDDI.ORG가 이후에 발족되었다. 이들은 2002년 6월 UDDI 버전 3.0을 발표하면서, 표준화에 대한 작업 일체를 ebXML로 잘알려진 OASIS에 일임하였으며, 현재 운영되고 있는 스펙은 버전2 또는 버전3 스펙을 가지고 구현하고 있다[12,13].

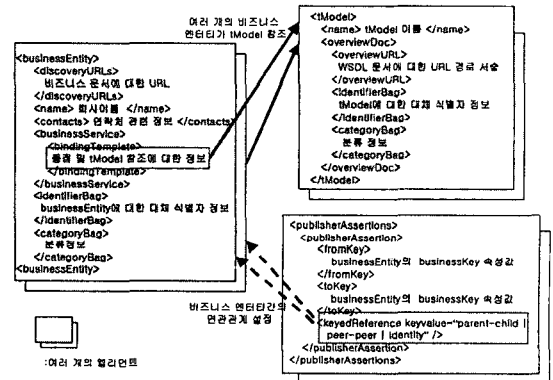


그림 3. UDDI의 문서구조와 관계
Fig. 3 Relation and document structure of UDDI

UDDI 레지스트리 데이터 구조는 XML 스키마로 정의되어 있으며, <businessEntity>, <publisherAssertion>, <identifierBag>, <categoryBag>, <businessService>, <bindingTemplate>, <tModel>의 대표적인 7가지 요소로 구성되어 있으며 그 관계는 그림3과 같다.

III. SOAP 기반의 상호작용 웹 서비스 시스템 설계 및 구현

3.1 시스템 개발도구 및 개발 환경

본 시스템에서의 개발환경으로 운영체제는 MS 윈도우 2000, 자바환경은 j2sdk1.4.1_03이며, EJB환경을 위해 BEA SYSTEM사의 Weblogic 서버를 사용하고, 데이터베이스 도구로는 MySQL을 이용하였으며, MySQL 사용자 인터페이스 도구로는 MySQL-Front를 사용하였다[15,16,17]. 웹 서비스

개발 도구로는 Sun사에서 배포하는 JWSDP(Java Web Services Developer Pack)을 이용하여 웹 서비스를 개발한다. 이 툴은 자바 개발자들이 XML 응용 프로그램과 웹 서비스 시스템 및 클라이언트를 쉽게 개발할 수 있도록 도와주는 여러 가지 도구 및 API를 제공하는 통합 개발 툴킷이다. 이 툴킷에서 본 시스템 설계에 직접 이용하고 있는 구성 요소는 다음과 같다[6]. JAX-RPC(Java API for XML-based RPC)는 SOAP 메시지에 근거한 RPC 방식의 웹 서비스 시스템 및 웹 서비스 클라이언트를 개발하기 위한 API이다[5,7]. Tomcat 서버는 Apache Group에서 제작한 서블릿, JSP를 실행시킬 수 있는 컨테이너이다. 서블릿 엔진이라고도 하며, 웹 클라이언트에서 서블릿 또는 JSP 실행 요청을 하면 웹 서버는 서블릿 엔진을 통하여 실행하고, 결과를 다시 웹 클라이언트에게 보낸다.

3.2 시스템 설계 및 구현

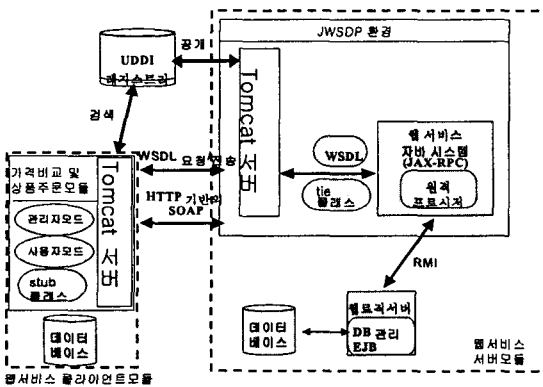


그림 4. 웹 서비스 시스템 설계 구조
Fig. 4 Design structure of web service system

본 논문에서는 그림4에서 바탕색이 있는 모듈들을 설계 및 구현하였다. 본 과제의 웹 서비스 시스템은 크게 웹 서비스 클라이언트 모듈과 서버모듈로 나누어지며, 클라이언트 모듈, 서버모듈, UDDI 레지스트리와의 SOAP 표준 환경기반으로 상호 정보를 교환하도록 설계 및 구현하였다. 따라서 표준인 SOAP환경을 따르는 모든 환경에서 이식가능하며, 플랫폼에 독립적으로 지원할 수 있다. 클라이언트 모듈은 사용자로부터 제품 검색과 가격비교를 위한 다양한 소팅기능을 포함하고 있고, 주문을 입력 받는 요소로 구성하였다. 서버모듈은 웹 서비스 자바시스템, DB관리 EJB로 구성하였다. 서버모듈은 여러개의 온라인 상점이 존재할 경우 여러개의 모듈이 동시에 존재할 수 있다. 웹 서비스 클라

이언트 모듈에서는 각각의 상점에 대한 모듈을 인식하기 위해 IBM의 UDDI 레지스트리에서 각 상점의 WSDL의 위치를 인식하여 각각의 WSDL 문서를 다운받아 자동으로 웹 서비스 클라이언트와 스텝 클래스를 설정하도록 함으로서 시스템 확장이 용이하도록 한다.

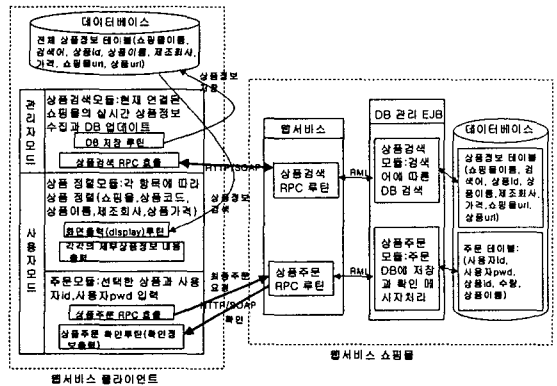


그림 5. 클라이언트와 쇼핑몰과의 웹 서비스 처리 흐름도

Fig. 5 Process flow diagram of web service between client and shopping mall

위의 그림5는 웹 서비스 클라이언트, 웹 서비스와 DB관리 EJB와의 처리 흐름과 작업 모듈들과의 관계를 나타내고 있으며, 웹 서비스 쇼핑몰은 여러개가 존재할 수 있다. 웹 서비스 클라이언트와 일대다 관계로 쉽게 확장 가능하도록 웹 서비스를 구성할 수 있다. 각각의 상세 설명은 다음과 같다.

1) 웹 서비스 클라이언트 모듈

클라이언트 모듈을 실행하는 환경은 Tomcat 서버를 사용하였다. 웹 서비스 클라이언트는 제품별 가격비교를 위한 다양한 소팅기능을 포함하는 사용자 인터페이스와 해당 제품 구매를 위한 처리시스템으로 구성되어 있다. 이 클라이언트는 원격프로서저를 호출하기 위해 JAX-RPC의 API를 이용하여 SOAP 형식으로 데이터를 송/수신한다. 각 제품별로 분류하여 여러 상점에서의 물건에 대한 정보와 가격을 호출하여 클라이언트에게 출력시킨다. 또한 해당 물건을 주문하기 위해 구매자의 아이디와 패스워드를 입력하여 인증 확인한 후 물건 주문이 성공되었음을 구매자에게 통지하여 준다. 또한 처리 속도를 빠르게 하기 위해 웹 서비스로 연결된 상품들의 업데이트 정보와 상품 검색어를

관리자 모드에서 수집하여 가격비교 데이터베이스에 저장하며, 사용자 모드에서는 최근의 업데이트된 상품 검색어를 이용하여 동적으로 검색할 수 있도록 지원하였다. UDDI 검색 브라우저는 UDDI에서 해당 웹 서비스의 WSDL 위치를 확인한 후 해당 웹 서비스의 WSDL 파일을 다운받도록 지원한다. 이 WSDL문서는 원격 프로시저의 호출을 위해 클라이언트의 스텝클래스를 생성할 때 사용한다. 원격 프로시저를 호출하기 위해 스텝/타이 호출 인터페이스 기법을 사용하였는데, 이것은 클라이언트 측에서 UDDI를 통해서 거래 쇼핑몰의 WSDL 문서를 다운받은 후 스텝 클래스를 생성한다. 프로그램 개발시 스텝 클래스를 참조하여 원격 프로시저를 호출할 수 있도록 설계하였다.

2) 웹 서비스 자바시스템

본 시스템은 원격 프로시저를 내장하고 있으며, 웹 서비스의 전체 입/출력 사양을 정의하는 WSDL 문서와 클라이언트와의 통신을 위한 타이 클래스를 포함한다. 본 시스템은 웹 서비스 클라이언트로부터 요청받은 원격 프로시저를 실행하며, 이에 대한 처리 결과를 다시 웹 서비스 클라이언트에게 통보해준다. 여기서도 JAX-RPC의 API를 이용하여 SOAP 형식으로 데이터를 송/수신한다. 원격 프로시저는 클라이언트의 요청에 따라 제품검색을 실행하는 기능과 주문에 대해 검증하고 주문 접수에 대한 정보를 클라이언트에게 전달하는 기능이 포함된다. 또한 원격 프로시저를 구성하기 위해 타이클래스와 WSDL 문서를 생성한다. 여기서 제품 정보관리를 위해 데이터베이스를 사용하며, 이 데이터베이스는 위치와 장소 독립적으로 구축할 수 있다.

3) DB 관리 EJB

DB 관리를 위한 DB 관리 EJB를 설계하였고, 웹 로직 서버환경을 이용하였다. 이러한 환경은 웹 서비스와 독립적으로 분산환경에서도 지원할 수 있도록 분산 RMI 방식을 채택하였다.

본 EJB 모듈은 웹 서비스 자바 시스템의 서버와 위치에 독립적이며, 실제로 데이터베이스를 관리하며, 처리하는 모듈이다. 해당 쇼핑몰에서 소유하고 있는 데이터베이스를 관리하며, 제품별 검색 기능과 주문에 대한 관리를 처리하는 기능으로 구성되어 있다. 데이터베이스내에 상품정보 테이블이 존재하고, 구매자의 확인을 위해 아이디와 패스워드를 관리하며, 구매자들에 대한 주문을 기록하는

주문 테이블이 존재한다. 본 시스템에서 사용하는 데이터베이스로는 MySQL을 사용하였다. 웹 서비스 자바시스템과 비즈니스서버 EJB사이에서 원격 프로시저 호출을 위해 자바의 RMI방식을 이용하여 설계하였다.

3.3 웹 서비스 클라이언트 실행 결과

본 시스템의 전체적인 실행 결과 시나리오는 먼저 웹 서비스 쇼핑몰 회사에서 UDDI 레지스트리에 웹 서비스 정보를 등록하고, 웹 서비스 가격비교 클라이언트에서 UDDI 레지스트리를 검색하여 웹 서비스 정보를 얻는다. 이 정보를 이용하여 웹 서비스 쇼핑몰과 바인딩하여 상품정보 검색을 실행하며, 이과정에서 HTTP/SOAP 메시지를 확인한다. 또한 상품정렬과 상품주문에 대한 순서적인 실행 결과를 보여준다.

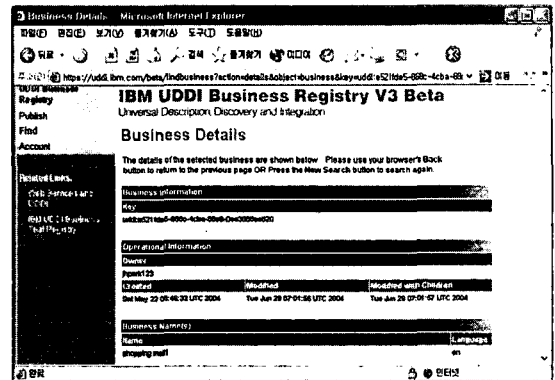


그림 6. UDDI 레지스트리에 등록된 화면
Fig. 6 Screen registered to UDDI registry

(1) 웹 서비스 쇼핑몰회사에서 IBM의 UDDI 레지스트리에 정보를 공개한 후 웹 서비스 가격비교 클라이언트에서는 이러한 정보를 검색하여 웹 서비스 쇼핑몰회사의 WSDL 파일을 얻는다. 이 WSDL 파일을 이용하여 클라이언트의 스텝 클래스를 생성하여 쇼핑몰회사와 바인딩한다. 위 화면은 쇼핑몰 회사가 IBM UDDI 레지스트리에 비즈니스 정보를 등록한 화면이다. 비즈니스 엔터티 이름을 "shopping mall1"으로 등록된 것을 보여주고 있고, 등록할 때 쇼핑몰 웹 서비스의 URL(endpoint)과 WSDL의 URL을 등록한다 (그림6 참조).

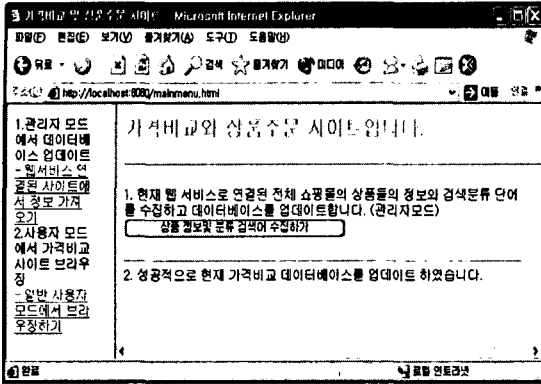


그림 7. 관리자모드에서 웹 서비스로 연결된 상품정보 수집하기

Fig. 7 Searching product information linked to web service in management mode

(2) 위 그림7은 가격비교 사이트에서 관리자가 최근의 정보를 수집하기 위해 현재의 데이터베이스를 업데이트 한다. 이에 따라 사용자의 상품 검색어를 동적으로 수시로 변경시킬 수 있다. 또한 사용자의 매번 웹 서비스 검색으로 각각의 쇼핑몰 데이터베이스를 검색하는 것보다 한번의 데이터베이스 전체검색으로 현재의 가격비교 서버의 데이터베이스에 업데이트 함으로서 속도 향상에 영향을 준다. 이 경우 가격비교 사이트와 쇼핑몰 서버들과는 SOAP 프로토콜을 이용하여 주고받는다. 사용자모드에서는 가격비교 서버의 데이터베이스를 검색함으로써 처리 속도 향상을 준다.

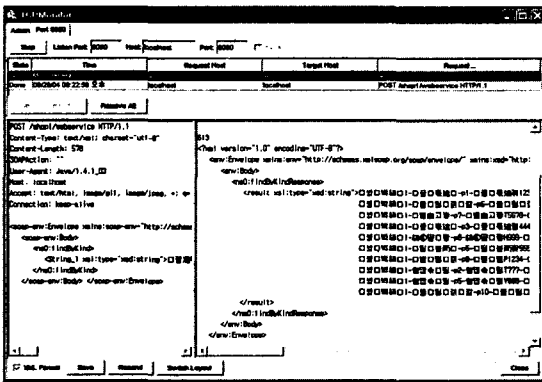


그림 8. TCPMonitor를 이용한 상품검색 요청과 응답 결과

Fig. 8 Result of product search request and response using TCPMonitor

(3) 위 그림8은 실제로 요청 SOAP 메시지와 응답 SOAP 메시지를 확인하기 위해 아파치 그룹에서 지원하는 TCPMonitor를 이용하여 SOAP 메시지를 확인하였다[18]. 왼쪽 영역은 클라이언트에서 요청 SOAP 메시지를 보내는 화면이고, 오른쪽 영역은 웹 서비스 쇼핑몰 서버에서 결과를 응답하는 SOAP 메시지이다. 즉 웹 서비스 클라이언트에서 find ByKind() RPC 메소드와 매개변수로 “전체 검색”이라는 문자열을 전송한 후, 그에 대응하는 쇼핑몰 서버에서 DB의 상품 정보 테이블을 검색하여 그 응답 결과를 SOAP 메시지로 보여주고 있다. TCPMonitor에서 한글이 제대로 보이지 않고 있으나, 이것은 문자 인코딩의 차이이며, 웹 서비스 실행에는 어떠한 영향을 주지 않는다.

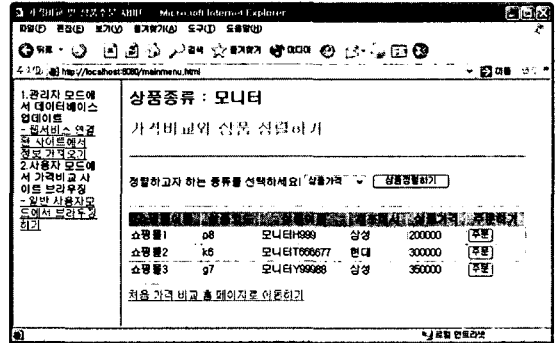


그림 9. 사용자모드에서 상품가격으로 정렬한 화면
Fig. 9 Screen sorted by price in user mode

(4) 위 그림9는 검색어를 “모니터”로 하여 데이터베이스를 검색한 후 상품정렬하기에서는 쇼핑몰이름, 상품코드, 상품명, 제조회사, 상품가격 정렬하기 중에서 상품가격을 기준으로 정렬하였을 경우의 화면을 보여준다.

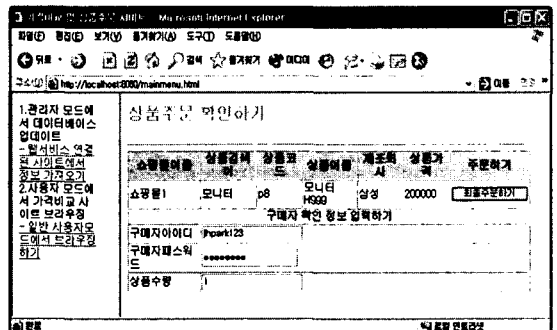


그림 10. 쇼핑몰의 p8상품을 구매할 경우의 화면
Fig. 10 Screen ordering p8 product of shopping mall

(5) 위 그림10은 쇼핑몰1의 p8상품을 주문한 경우의 화면이다. 이 경우 구매상품에 대한 정보와 구매자 확인을 위한 아이디와 패스워드, 상품 수량을 입력한 후 최종 주문버튼을 클릭한다. 클릭한 후 쇼핑몰1의 웹 서비스 서버 모듈에 관련 제품의 주문을 의뢰한다.

쇼핑몰1의 MySQL 데이터베이스에 주문을 신청한 화면을 보여주고 있다. shop1 데이터베이스가 쇼핑몰1이며, 그 안에는 주문테이블(order1)과 상품정보 테이블(product)이 있다. 주문 테이블에는 사용자 아이디, 패스워드, 상품코드와 수량이 테이블의 맨 마지막에 저장된다.

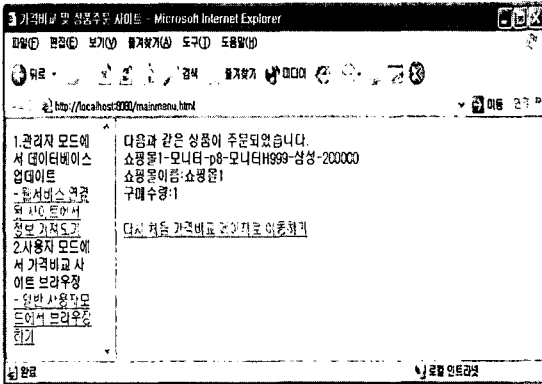


그림 11. 구매자에게 신청한 내용을 보여주는 확인 화면

Fig. 11 Confirmation screen showing order product to buyer

(6) 위 그림11은 최종 주문하기 버튼을 클릭한 후 올바르게 상품이 신청된 경우 구매자에게 신청한 내용을 보여주는 화면이다. 관련 웹 서비스 서버 모듈에서 주문 승인에 대한 정보를 클라이언트로 전송하면, 웹 서비스 클라이언트 모듈에서 확인 메시지를 출력한다.

3.4 고찰

가격비교 사이트에서 쇼핑몰의 상품정보와 가격을 실시간으로 업데이트 가능하며, 쇼핑몰에서 가격비교 사이트로 상품정보 수동 입력이 불필요하다. 또한 가격비교 사이트에서 각항목별로 정렬이 가능하도록 기능을 지원하였으며, 상품주문도 실시간으로 직접 주문가능하며, 그에 대한 확인메시지도 지원하였다. 그러나 상호호환적인 웹 서비스 환경을 지원하기 위해, 쇼핑몰의 데이터베이스내에 상품주문과 상품정보 스키마가 표준적으로 통일되어야 하며, 검색어에 대한 키워드도 일치되어야 웹 서비스 가격비교 클라이언트에서 일관된 방법으로 웹 서비스가 가능하다.

IV. 결 론

현재 가격비교 사이트에서는 관련 쇼핑몰 회사들이 가격비교 사이트에 가입하여 수동으로 상품정보와 가격을 입력하여야 하며, 또한 상품가격을 자동으로 실시간 가격변동을 지원할 수 없는 문제점이 있다. 이러한 문제점을 해결하기 위해 본 논문에서는 웹 서비스 기술을 이용하여 실시간의 능동적인 상품 가격비교 및 직접 상품 주문에 대한 SOAP기반의 동적 상호작용 웹 서비스 시스템을 설계 및 구현하였다. 본 연구의 웹 서비스 시스템은 크게 상품검색, 상품정렬과 상품주문을 지원하는 웹 서비스 클라이언트 모듈과 상품검색과 상품주문 원격 프로시저 호출을 포함하는 서버모듈로 나누어지며, SOAP 표준 환경기반으로 상호 정보를 교환하도록 구현하였다. 클라이언트 모듈은 사용자로부터 제품 검색 및 가격비교 소팅과 주문을 입력 받는 요소로 구성하였고, 서버모듈은 웹 서비스 자바시스템, DB관리 EJB 모듈로 구성하였다. 서버 모듈은 여러개의 온라인 상점이 존재할 경우 여러개의 모듈이 동시에 존재할 수 있도록 클라이언트 모듈과 일대다 관계로 구성할 수 있다. 웹 서비스 클라이언트 모듈에서는 각각의 상점에 대한 모듈을 인식하기 위해 IBM의 UDDI 레지스트리를

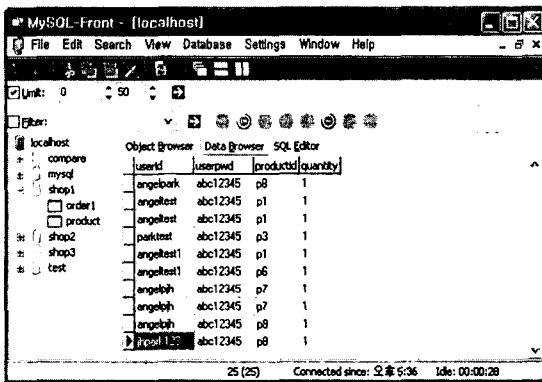


그림 12. 데이터베이스에 주문 상품이 저장된 화면

Fig. 12 Screen stored with order product in database

(7) 위 그림12는 구매자가 상품을 선택하여 해당

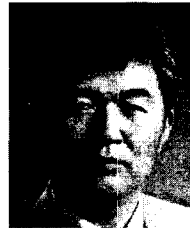
이용하여 각 상점의 WSDL의 위치를 인식하여 각각의 WSDL 문서를 다운받아 자동으로 웹 서비스 클라이언트와 스텝클래스를 설정하도록 하였다. 본 시스템은 사실표준 환경인 SOAP, WSDL, UDDI 표준을 따르는 모든 환경에서 이식가능하며, 플랫폼에 독립적으로 지원할 수 있다.

참고문헌

- [1] Patric Cauldwell and et al., Professional XML Web services, wrox, 2002.
- [2] Scott Short, Building Xml Web Services for the Microsoft .Net Platform, MicroSoft Press, 2002.
- [3] Adam Freeman and Allen Jones, MicroSoft .Net XML Web Services Step by Step, MicroSoft Press, 2003.
- [4] Simeon Simeonov and et al., Building Web Services with Java, SAMS, 2002.
- [5] Eric Armstrong and et al., The Java Web Services Tutorial, <http://java.sun.com/webservices/docs/1.3/tutorial/doc/index.html>, 2003.
- [6] Java Web Services Developer Pack, <http://java.sun.com/webservices/downloads/webse rvicespack.html>, 2003.
- [7] Janice J. Heiss, "Jax-RPC brings Potability to Web Services", http://java.sun.com/features/2003/01/jax_rpc.html
- [8] W3C, SOAP Version 1.2(part0,part1,part2) (Recommendation), <http://www.w3.org/2002/ws/>, 2003.
- [9] W3C, WSDL Version 2.0(Working Draft), <http://www.w3.org/2002/ws/>, 2003.
- [10] UDDI standard spec, <http://www.uddi.org/specification.html>
- [11] UDDI Version 3.0 UDDI Spec Technical Committee Specification, <http://uddi.org/pubs/uddi-v3.00-published-20020719.htm>, 2002.
- [12] IBM UDDI Business Registry V3 beta, <https://uddi.ibm.com/beta/registry.html>
- [13] MicroSoft UDDI Business Registry Node, <https://uddi.beta.microsoft.com/>.
- [14] W3C recommendation, Extensible Markup Language(XML)1.0(third edition), <http://www.w3.org/TR/REC-xml/>, 2004.
- [15] Bea systems, Bea weblog platform 8.1, http://www.bea.com/framework.jsp?CNT=homepage_main.jsp&FP=/content.
- [16] MySQL, "MySQL reference manual", <http://dev.mysql.com/doc/>.
- [17] Star Tools, "Welcome to MySQL-Front", <http://www.mysqlfront.de/>.
- [18] Apache Group, Using the Axis TCPMonitor tool, <http://ws.apache.org/axis/java/user-guide.html>.

저자소개

김철원(Cheol-Ueon Kim)



1997년 광운대학교 컴퓨터공학과(공학박사)
1988년~현재 호남대학교 컴퓨터 공학과 교수
※관심분야 : XML 응용, 멀티미디어 정보

박종훈(Jung-Hun Park)



광운대학교 전자계산기공학과(학사)
광운대학교 대학원(석사)
광운대학교 대학원(박사)
한국전산원 선임연구원
현재 중부대학교 컴퓨터멀티미디어공학과 조교수
※관심분야 : XML 웹서비스, XML 표준, XML 검색