

碩士學位論文

웹 서비스 기반의 효율성을 고려한
분산 환경 데이터베이스 접근 시스템



컴퓨터工學科

朴 宣 禹

2003年 12月

웹 서비스 기반의 효율성을 고려한 분산 환경 데이터베이스 접근 시스템

指導教授 郭 鎬 榮

朴 宣 禹

이 論文을 工學 碩士學位 論文으로 提出함



朴宣禹의 工學碩士學位 論文을 認准함

審査委員長 이 상 준 印

委 員 곽 호 영 印

委 員 변 영 철 印

濟州大學校 大學院

2003年 12月

**Distributed Environment Database Access Model
based on Web Services for Efficiency**

Sun-Woo Park

(Supervised by professor Ho-Young Kwak)



**A thesis submitted in partial fulfillment of the
requirement for the degree of Master of Engineering**

**Department of Computer Engineering
GRADUATE SCHOOL
CHEJU NATIONAL UNIVERSITY**

목 차

Summary	1
I. 서 론	2
II. 관련 연구	4
1. 웹 서비스와 표준들	4
1) SOAP	6
2) WSDL	7
3) UDDI	8
2. 분산 데이터베이스 시스템	9
3. 웹 서비스의 향후 발전 방향	14
III. 제안된 시스템	16
1. 제안된 시스템 모델	16
2. 제안된 시스템 구조	17
1) Management	17
2) Client	18
3) Web Service	18

IV. 시스템 구현 결과	19
1. 개발환경 및 서비스 환경	19
2. Management	20
3. Client	21
4. 웹 서비스 구성 요소	22
1) SOAP	22
2) WSDL	24
3) UDDI	26
5. 상호운영 트랜잭션 결과	27
1) Management Server	27
2) Client	29
V. 결론 및 향후 연구	31
참고문헌	32



Summary

In this paper, we propose and design an interoperable system which can be used in distributed environment and heterogeneous database based on the web services for management and maintenance of the business related data in intranet or between the enterprises.

Using the WSDL, UDDI, SOAP standard technology of web services, the proposed system can be a basis of a central integrated management foundation of a local database, and a solution of managing and processing the distributed environment database under the different operating system and transactions in intranet or between the enterprises.

As a result, the proposed system shows the possibility of interoperability of distributed environment database, and being a sub-system of a distributed environment database Management.

1. 서 론

오늘날 비즈니스의 요소들은 무수히 많은 디바이스들과 약 10억 인구들로 복잡하게 연결되어 있다. 또한 전자적 시장은 수많은 처리해야할 데이터들과 다양한 자원들이 인터넷을 통해 접근되고 서로 상호 작용하게 되었다. 하지만 지금까지는 이들 비즈니스 요소들과 비즈니스 서비스들의 연결을 위해 인터넷을 사용하여 단순히 자료를 검색하고 다운로드만이 가능하였다. 그러나 이들 전자적 시장에서는 비즈니스 서비스를 기업 내 직원들과 또는 다른 파트너들과도 보다 효율적인 활동을 위해 협력을 하고 다양한 서비스들을 보다 빠르게 배포, 공급할 수 있기를 원하는 요구사항이 계속해서 증가하게 되었다[1]. 이 흐름에 따라 웹을 사용한 새로운 모델로써 XML(eXtensible Markup Language)[2]을 기반으로 한 웹 서비스가 등장하였다. 웹 서비스는 이미 표준화된 기술들 즉, SOAP(Simple Object Access Protocol)[3], WSDL(Web Service Description Language)[4], UDDI(Universal Description, Discovery and Integration)[5]들과 함께 발전하였으며 각각의 표준들과 상호 작용하는 응용들이라고 할 수 있다. 웹 서비스는 브라우저를 사용하지 않고도 프로그램에 의해 자동적으로 거래가 가능하게 할 수 있으며 분산된 컴퓨팅 환경에서 기술(description), 출판, 검색, 호출을 할 수 있게 되었다. 따라서 비즈니스 활동에 좀더 나은 서비스 제공을 위해 웹 서비스를 이용함으로써 서로 다른 비즈니스들의 응용 간에도 좀 더 빠르고 쉽게 상호작용하게 되고 효율적인 협력을 할 수 있게 되었다 [6].

이상적인 분산 데이터베이스 시스템에서, 사이트들은 공통의 전역 스키마를 공유하며(비록 몇몇 릴레이션들은 일부 사이트에만 저장되어 있겠지만) 모든 사이트들이 동일한 분산 데이터베이스 관리 소프트웨어를 구동하고, 사이트들은 다른 사이트들의 존재를 인식하고 있게 된다. 실제로 분산 데이터베이스는 각자 개별적인 스키마를 가지고 서로 다른 데이터베이스 관리 소프트웨어를 동작한다. 기존에 존재하는 다중의 데이터베이스 시스템들과 함께 연결해서 구축되어야 하며, 이러한 시스템을 다중 데이터베이스 시스템(Multidatabase System) 혹은 이질

(Heterogeneous) 분산 데이터베이스 시스템으로 불린다. 그리고 많은 새로운 데이터베이스 응용들은 이질적인 하드웨어와 소프트웨어 환경으로 구성된 곳에 위치한다. 이미 존재하는 다양한 데이터베이스로부터의 데이터를 요구한다. 이질적 분산 데이터베이스에 위치한 정보를 조작하기 위해서는 이미 존재하는 데이터베이스 시스템에 추가적인 소프트웨어 계층을 요구하고 있으며, 또한 이질적인 시스템을 동질적 분산 데이터베이스로 완전히 통합하는 것은 기술적 또는 조직적 어려움을 가지고 있다[7].

따라서, 본 논문에서는 기업 내에서 또는 기업간 비즈니스와 관련된 데이터들을 관리하고 유지하기 위해 웹 서비스 기반의 분산되면서 이질적인 분산 환경의 데이터베이스를 상호 운영할 수 있는 시스템을 제안하고 설계하였다. 설계된 시스템은 한 조직의 지역 데이터베이스를 중앙에서 통합 관리할 수 있는 기반이 되며 또한 서로 다른 운영 환경하의 이질적 분산 환경 데이터베이스를 활용하고 관리할 수 있는 방안을 제시한다.

본 논문의 구성은 2장은 관련연구로 웹 서비스와 표준들과 분산 환경 및 분산 데이터베이스, 웹 서비스 향후 발전 방향에 대해 서술하였고, 3장에서는 제안된 시스템 모델을 기술하였으며, 4장에서는 구현 결과를 보였다. 마지막으로 5장에서 결론 및 향후 연구를 기술하였다.

II. 관련 연구

1. 웹 서비스와 표준들

웹 서비스는 XML 기반으로 웹 표준들을 서로 상호 작용하는 응용이다. 웹 서비스의 정의는 다음과 같다. W3C에서는 “웹 서비스는 공개된 인터페이스와 바인딩을 XML 을 사용하여 정의하고, 기술하고, 발견이 가능해지며, 인터넷 기반의 프로토콜을 거쳐 XML 기반의 메시지를 사용하는 다른 응용 소프트웨어 응용들과 직접적인 상호작용이 가능한 URI(Uniform Resource Indicator) 에 의해 식별된 소프트웨어 시스템이다.“이라 정의하고 있으며, 웹 서비스에 기반이 되는 표준들로는 SOAP, WSDL UDDI 가 있다. SOAP은 웹 서비스에 접근하기 위해 사용되는 XML 기반의 통신 프로토콜이다. WSDL 은 웹 서비스 기술 언어로써 오퍼레이션들과 관련된 데이터 타입들, 지원되는 전송 프로토콜을 포함한다. UDDI 는 웹 서비스들을 출판하고 발견하기 위한 SOAP기반의 API(Application Programming Interface)이다[8].

WWW 이 소개된 지는 비록 10여년 밖에 안됐지만 지속적으로 발전해 초기의 건조한 텍스트 기반의 링크가 고작이었던 웹은 오늘날 멀티미디어는 물론이고 글로벌 비즈니스를 지향하는 기업 전산 환경에 없어서는 안 될 중요한 수단으로 자리하고 있다. 가트너는 웹 서비스를 웹의 발전 과정이라는 관점에서 다음과 같이 설명하였다.

- Pre-Web services : 인터넷 기술이나 XML 을 사용해 프로세스와 데이터에 대한 액세스를 제공하는 단계로, XML/EDI 어플리케이션이나 XML 을 지원하는 데이터 소스뿐만 아니라 웹 페이지와 웹 콘텐츠를 포함하며, 주식 시세나 콘텐츠를 내려받는 것과 같은 기초적인 웹 기능을 수행한다.

- Nonstandard Web services : 서비스 지향적(Service-oriented) 구조이지만 모두가 웹 서비스 표준은 아니다. 예를 들어 SOAP, UDDI 를 사용하지 않는 EJB 나 COM 컴포넌트 서비스로 종종 XML 과 SOAP 을 사용할 수 있다.
- Well-behaved Web services : SOAP · WSDL · UDDI 와 같은 모든 주요한 웹 서비스 표준을 지원한다. 경우에 따라 이러한 서비스는 비즈니스의 상호 운영 기능을 개선하기 위해 WSFL/XLANG, XSLT 혹은 그 밖의 새로운 표준을 사용할 수 있다.

웹 서비스의 동작과정은 세 가지 역할과 그 역할 간의 세 가지 상호작용으로 요약할 수 있다. 즉 서비스 제공자는 서비스를 개발해 그 상세내역(description) 과 함께 서비스를 ①공개하고(publish) 서비스 요청자는 원하는 서비스를 ②검색(find)한 후, 자신이 웹서비스나 어플리케이션에 ③바인딩(binding)하는 과정을 거치게 된다. 그림1은 웹 서비스 기본 아키텍처를 나타낸다.

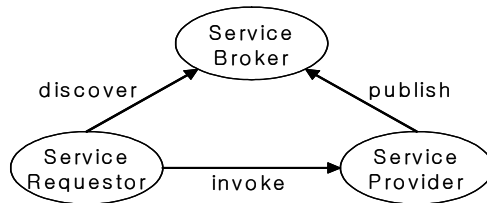


그림 1. 웹 서비스 기본 아키텍처

- Service Requester : 비즈니스 관점에서 보면, 서비스 요청자는 원하는 특정 기능을 요청하는 비즈니스가 된다. 기술적인 관점에서 보면, 서비스를 검색해 호출하는 어플리케이션이 되며 이는 브라우저일 수도 있고 사용자 인터페이스가 없는 어플리케이션 또는 웹 서비스일 수도 있다.

- Service Provider : 비즈니스 관점에서, 서비스 제공자는 서비스를 소유하고 있는 개인이나 기업이 되며 서비스를 개발해 서비스 레지스트리에 등록하면 요청된 웹 서비스를 제공한다. 기술적인 관점에서 보면, 서비스에 대한 요청을 접수해 서비스를 실행하고 다시 그 결과를 서비스 요청자에게 돌려주며, 간단하게 서비스에 대한 접근을 제어하는 플랫폼이라고 할 수 있다.
- Service Registry : 서비스 브로커(service broker)라고도 한다. 서비스 제공자가 공개한 서비스에 대한 상세내역을 등록 관리하며 서비스 요청자가 원하는 기능의 서비스를 검색해 바인딩할 수 있는 정보를 제공한다.

웹 서비스를 하기 위해서는 다양한 표준이 필요하다. 웹 서비스를 개발하고 등록해 공개하기 위한 표준부터 시작해 특정 기능의 웹 서비스를 검색하고 자신의 어플리케이션에 연결해 서로 다른 네트워크에 있는 어플리케이션을 실행할 뿐만 아니라 그 결과를 가져와 원하는 형태로 볼 수 있기 위해서는 산업 전반에 걸쳐 공통으로 사용할 수 있는 다양한 표준이 필요하다. 웹 서비스에 기반이 되는 표준들로는 SOAP, UDDI, WSDL 이 있다.

1) SOAP(Simple Object Access Protocol)

웹 서비스가 발전함에 따라 과거의 메시지를 통한 형태에서 나아가 XML 형태의 RPC 프로토콜이 요구됐다. XML-RPC 는 이기종 운영체제에서 인터넷을 통해 다른 환경에서 수행되는 프로시저를 호출할 수 있도록 하기 위해 HTTP 를 통해 XML 형태로 전송하는 것이다. 그러나 XML-RPC 는 상이한 객체 인프라스트럭처 간에 사용하는 경우 쌍방간에 브리지를 제공해야 하는 어려움이 있다. 그리고 각 요청과 응답을 위한 스키마를 생성하는 메커니즘이 필요하기 때문에 각 레이어를 복잡하게 만들고 더욱 더디게 만들기도 한다. SOAP 은 XML-RPC 와 매우 유사하다. SOAP 역시 HTTP 와 XML 문서를 통한 프로시저 호출 형태다. SOAP은 UserLand 와 DevelopMentor, MS 의 협력에 의해 만들어졌다. 초기 SOAP 1.0은 XML-RPC 를 기본으로 하고 HTTP 에서만 가능했지만, SOAP 1.1부터는

HTTP·FTP·SMTP·POP3 에서도 사용할 수 있고, HTTP 확장 프레임워크도 지원하게 됐다. SOAP 은 메쏘드와 인자를 넘기거나 객체를 호출하는 RPC 형태와 어플리케이션 통합이나 EDI 데이터 교환을 위한 Self-describing XML 메시지 형태로도 가능하다. <표 1>는 XML-RPC 와 SOAP 을 비교한 것이다[10].

<표 1> XML-RPC와 SOAP 비교

구분	XML-RPC	SOAP
목표	XML-RPC는 정보를 주고받을 때 간단하고 효과적인 방안을 찾는다. 간결하고 확장성 있는 포맷으로 이기종 시스템이나 다른 환경에서도 동작할 수 있도록 신속하게 연결 할 수 있는 프로토콜을 구현할 수 있다.	SOAP은 사용자 정의 데이터 타입, 수신 지정 능력, 메시지에 따른 프로시저 제어 등의 기능을 구현함으로써 XML-RPC의 한계를 수정하려고 했다.
특징	XML-RPC는 그 형태가 단순하므로 이해·구현·디버그가 매우 쉽고 문법도 간단해 오류를 찾거나 수정하기가 편리하며 안정적이다.	SOAP에서는 개발자들이 임의로 간단하거나 복잡한 데이터 타입을 정의할 수 있으며 커스터마이징이 가능하다.
한계	XML-RPC의 구조를 보면 단순하게 이름과 값이 매치된 형태이므로 인자로 오브젝트를 넘기는 것은 불가능하다. XML-RPC의 단순성이 장점이 되기도 하지만 가장 큰 문제점이다.	XML-RPC에 비해 SOAP은 최근에 나온 개념이고 계속해서 그 형태가 발전해 나가고 있는 상태이다.

2) WSDL(Web Services Description Language)[11]

WSDL 은 기업에서 UDDI 에 등록된 웹 서비스를 어떻게 찾고 알아볼 수 있는지에 관한 표준 XML vocabulary다. WSDL 은 MS SOAP 과 NASSL(Network Accessible Service Specification Language)에 기반을 두고 있다. 웹 서비스를 기술하는 스크립트인 WSDL 은 XML 포맷으로 구성되고 HTTP 를 통해 전달될 수 있으며 인터페이스를 정의하는 IDL 에 해당한다. WSDL 을 이용해 웹 서비스 제공자는 사용자에게 해당 웹 서비스의 정확한 인터페이스와 사용되는 데이터 타입, 전송 프로토콜에 대한 상세 정보를 전달할 수 있다. <표 2>는 WSDL 의 구성을 보여준다.

<표 2> WSDL의 구성

영역	설명
타입	교환된 메시지를 설명하는데 사용된 데이터 형식을 정의
메시지	전송되는 데이터의 추상 정의를 나타내며, 메시지는 논리적인 부분으로 이뤄져 있으며 각 부분은 특정 형식 시스템 내의 정의와 연관
portType	추상 작업의 집합이며 각 작업은 입력 메시지 및 출력 메시지를 참조
바인딩	특정 portType에 의해 정의된 메시지 및 작업에 대해 구체적인 프로토콜과 데이터 형식을 지정
Port	바인딩에 대해 주소를 지정하기 때문에 단일 통신 종점을 정의
서비스	관련된 포트의 집합을 집계하는데 사용

3) UDDI(Universal Description, Discovery and Integration)

2000년 MS · Ariba · IBM이 제안한 UDDI 는 B2B트랜잭션, 전자상거래, 웹 서비스를 자동으로 처리하기 위한 프레임워크로 SOAP 메시징 스키마가 웹 서비스에서 동작하도록 하는 API 조합이다. 기업들은 외부에 공개된 UDDI 레지스트리에 자신들의 제품이나 제공하는 서비스를 등록 · 소개해 인터넷을 통해 사업을 운영하고 고객이나 파트너를 편리하게 검색할 수 있도록 해준다[12]. UDDI 를 사용하면 어플리케이션은 각 디렉터리 서버가 지원하는 한도 내에서 어떤 서비스가 가능한지를 바로 질의해서 얻어낼 수 있다. 데이터들은 표준 분류법을 사용해 분류할 수 있으므로 범주별 정보 검색이 가능하며, 무엇보다도 중요한 점은 UDDI 는 회사의 서비스 기술 인터페이스에 대한 정보를 포함한다는 것이다. 일련의 SOAP 기반 XML API 호출을 통해 디자인 타임과 런타임에 모두 UDDI 와 상호 작용해 기술 데이터를 검색함으로써 이러한 서비스를 호출하고 사용할 수 있도록 할 수 있기 때문에, UDDI 는 웹 서비스를 기반으로 하는 소프트웨어 환경의 인프라 역할을 한다.

2. 분산 데이터베이스 시스템

분산 데이터베이스는 네트워크에 분포된 다양한 지역 데이터베이스들의 집합을 말한다. 즉 컴퓨터 네트워크상에 각 사이트에 물리적으로 분산되어 있는 공유된 데이터들을 논리적으로 통합한 데이터베이스이다. 또한 분산 데이터베이스를 관리하고, 사용자에게는 투명(transparent)한 분산이 되도록 하게 하는 소프트웨어 시스템을 분산 데이터베이스 관리 시스템이라고 한다. 즉 기존의 중앙 집중식 데이터베이스 시스템은 한 개의 데이터베이스 시스템에 모든 정보가 저장되어 있는 반면, 분산 데이터베이스 시스템은 모든 정보가 일정한 논리에 의해 나뉘어져 관리되고, 사용자는 이렇게 분산된 데이터베이스들이 통합된 하나의 가상 데이터베이스를 사용하여 정보를 이용하게 된다[13, 14]. 이러한 분산 데이터베이스를 이루기 위해서는 분산과 논리적 연계라는 중요한 두 가지 면이 갖추어져 있어야 한다. 여기서 분산이란, 하나의 사이트에만 데이터가 존재하는 중앙 집중형 데이터베이스와 구별되어 여러 사이트에 물리적으로 퍼져 있는 것을 말하며, 논리적 연계란 지역(local) 데이터베이스들을 단순하게 모은 것과는 뚜렷하게 구별된 것으로 데이터들이 서로 연관되는 성질을 갖는 것을 의미한다.

예를 들어 A라는 사람이 S기업의 광주지사 지사장이라고 생각해 보면, S기업에는 ‘광주’이외에도 서울, 전주, 부산, 제주 등의 여러 지사가 있을 것이며, 이러한 지사들로 나뉘어 관리되는 것을 분산이라고 할 수 있겠다. 지사장으로써 효율적인 운영은 회사 전체의 이익이 커야만 광주지사의 모든 직원들에게 이점이 돌아오겠지만, 우선 각각의 지사가 효율적으로 운영될 때, 작게는 각지사의 직원, 그리고 넓게는 S기업이 성공할 수 있는 비결일 것이다.

분산 데이터베이스로 각 지사가 관리된다고 한다면 다음과 같이 두 가지 방법으로 나뉘어 생각할 수 있다. ①전형적으로 케이블을 사용하고 있는 근거리 네트워크(Local Area Network)이다. ②전화선이나 위성들을 사용하는 원거리 네트워크(Wide Area Network)이다. 또한, 두 가지 유형의 네트워크를 혼합해서 사용하는 경우도 있다. 우선 근거리 네트워크로 관리된다면 그림 2와 같이 실제 구조를 표현할 수 있을 것이다.

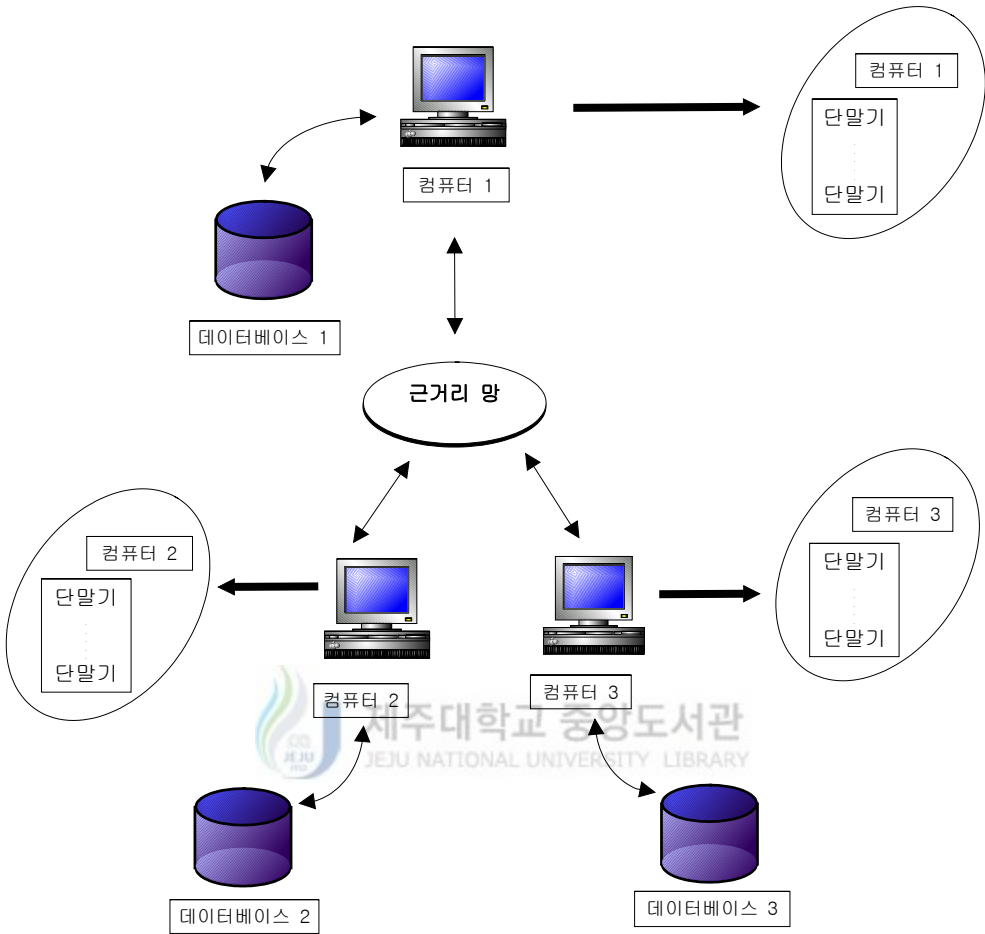


그림 2. 근거리 망에서의 분산 데이터베이스

서울에 있는 본사에는 각 지사의 데이터를 관리하는 Server 에 해당하는 컴퓨터가 존재하며, 이러한 컴퓨터들은 높은 대역폭(high bandwidth)을 갖는 근거리 통신망으로써 연결되어 있다. 이러한 각각의 Server 들에 Client 에 해당되는 각 지점의 창구 단말기가 연결된다. 이렇게 연결된 각각의 컴퓨터와 데이터베이스가 근거리 통신망의 하나의 사이트를 구성하게 되는 것이다. 광주지사에서는 단말기를 통하여 매출현황, 재고과약, 직원관리, 고객관리 등의 업무를 처리 할 수 있으며, 단말기로부터 들어온 작업은 그 지사의 데이터베이스만을 액세스하기

위하여 광주 지사의 컴퓨터에서 실행되는 지역 작업(local application)을 수행한다. 이번에는 원거리 네트워크로 관리된다고 예를 들어본다면 그림 3과 같이 구조가 표현 될 수 있다.

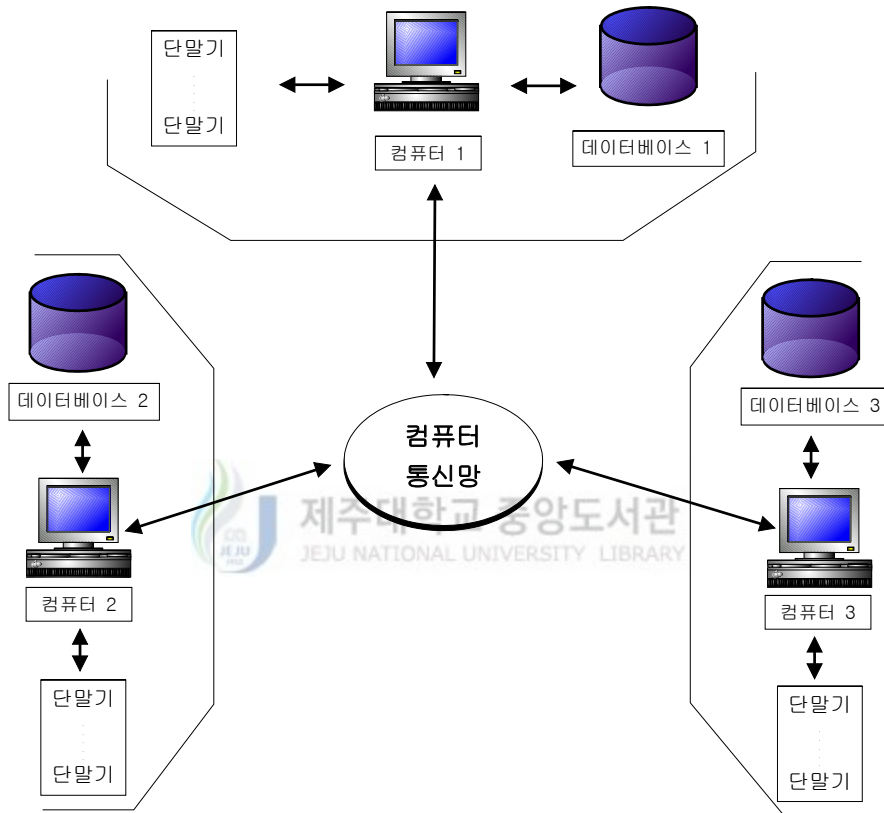


그림 3. 지역적으로 분산된 데이터베이스

그림 3에서 보인 것과 같이 원거리 네트워크에서는 각 지사에 Server 컴퓨터를 두고 이 Server 가 그 지사의 창구 단말기들과 계정 데이터베이스를 관장하게 된다. 그리고 각 지사의 Server 들은 통신망을 통하여 연결되는 것이다. 근거리 네트워크와 구조적인 차이는 있지만, 각 지사의 업무를 처리하는 면에서나, 데이터베이스 관리 차원에서 볼 때 지역 작업을 수행한다는 것은 다르지 않다.

다만, 근거리 네트워크에서는 국지성(locality)이 작업을 실행하는 컴퓨터들이 지역적으로 떨어져 있는 경우에 적용되지 않고, 어떤 컴퓨터의 데이터베이스에 국한한다는 면을 나타내는 것만이 다르다.

여러 지사들이 서로의 데이터를 공유하는 것을 논리적 연계라고 할 수 있을 것이다. 예를 들면 독자는 광주지사를 운영하는데 있어서 타 지사의 데이터를 필요로 하는 일이 종종 있게 될 것이다. 만약 독자의 지사에서는 품질 된 제품을 고객이 찾을 때 다른 지사에 재고가 있는지 알아야할 것이며, 또한 다른 지역에서 광주로 이사 온 고객의 효율적인 관리를 위하여 고객의 취향 등의 자료도 얻을 수 있다면 더욱 효율적일 수 있다. 이러한 데이터의 공유를 위해서는 분산 작업(distributed application)이라고도 불리는 전역 작업(global application)이 존재하여 각 지사에 있는 데이터들을 액세스할 수 있을 때 지역 데이터베이스 집합과 구별된다. 전역 작업에 있어서, 서울지사에 하나밖에 남아있지 않은 제품을 광주지사에 요구되어 이미 보내졌고, 제주지사에서는 갱신되지 않은 서울지사의 재고현황만을 믿고 고객에게 매매약속을 하게 된다면 기업의 신용에 큰 영향을 미치게 될 것이다. 또한 직원관리에 있어서 직원의 실적을 바로 갱신해 줄 수 없다면, 직원의 인사에 막대한 착오를 일으킬 수도 있을 것이고, 급여가 누락되는 경우도 일어날 수 있을 것이다. 물론 이러한 전역 작업에 대한 요건을 보장하는 것이 쉬운 일은 아니다. 하지만 이러한 전역 작업이 하나의 트랜잭션으로 간주되어 트랜잭션의 성질을 모두 갖고 있어야만 한다. 이상의 내용을 종합하여 분산 데이터베이스의 정의를 한다면, 컴퓨터 통신망 내의 서로 다른 사이트에 분산되어 있는 데이터들의 집합이며, 컴퓨터 통신 내의 각 사이트는 자치적으로 처리할 수 있는 능력을 가지고 있고, 지역 작업을 실행할 수 있다. 각 사이트는 또한 적어도 하나 이상의 전역 작업을, 즉 통신을 통하여 여러 사이트에 저장된 데이터를 액세스하는 작업을 실행하는 데에 관여한다.

분산 데이터베이스를 이용하기 위한 연결 표준으로는 일반적으로 ODBC(Open Database Connector), OLE-DB(Object Linked Environment-Database), DCOM(Distributed Component Object Model), 그리고 OODB(Object Oriented Database)를 위해 OMG(Object Management)에서 제안한 OMA(Object Management)모델의 컴포넌트인 ORB(Object Request Broker)의 명세를 따르는

CORBA(Common Object Request Broker Architecture)가 데이터베이스 시스템 간 통신을 위해 널리 사용되고 있는 바, 이는 벤더 중심적인 프로토콜인 성격과 접속 지향형 통신의 성격을 갖는다. 따라서 인터넷을 통한 ERP(Enterprise Resource Planning)와 전자상거래 솔루션과 같은 분야에서 사용하기 어렵다[15].



3. 웹 서비스 향후 발전 방향

웹 기술의 발전 단계를 구분할 때 웹 서비스 기술의 출현 이전을 제 1세대 웹으로, 웹 서비스 출현 이후를 제 2세대 웹으로 분류하고 있다[16].

<표 3> 웹 서비스의 진화

1세대 웹 : 웹 서비스 출현 이전	2세대 웹 : 현재의 웹 서비스	3세대 웹 : 시맨틱 웹서비스
<ul style="list-style-type: none"> ■ 브라우저 기반의 이용 ■ 사용자 중심 ■ 콘텐츠, 어플리케이션 지향 ■ URL을 이용한 수동적 접근 	<ul style="list-style-type: none"> ■ 어플리케이션 기반의 이용 ■ 사용자, 어플리케이션 중심 ■ 서비스 지향 ■ 레지스트리를 이용한 검색, 바인딩 	<ul style="list-style-type: none"> ■ 에이전트 기반의 이용 ■ 사용자, 어플리케이션 중심 ■ 목적 지향 ■ 온톨로지 기반의 서비스 검색, 실행, 조합

현재의 웹 서비스는 SOAP, WSDL, UDDI 와 같은 기술 표준으로 구성된다. 이러한 웹 서비스는 기존의 사용자 중심, 어플리케이션 지향의 웹 환경을 변모시키며, 기존의 RPC, CORBA 와 같은 분산 컴퓨팅 모델을 대체하면서 빠르게 도입되고 있다. 또한 인터넷과 웹은 전자적 마켓의 빠른 성장과 함께 물리적인 마켓을 보완하여 기존의 전통적인 비즈니스 모델에서 제품과 서비스를 전달하는 새로운 방법을 형성하는 동적인 e-Business 로의 혁신적인 변화를 가져왔다.

무선통신 기술과 GPS(Global Positioning System) 기술을 이용해 차량에 위치기반 시스템(Location-based Services, LBS)을 비롯한 다양한 콘텐츠 및 커뮤니케이션 서비스를 제공하는 텔레매틱스(telematics)는 자동차 산업 이외에 통신 산업, 유무선 콘텐츠 산업, 컴퓨팅 디바이스 및 소프트웨어 산업 등 다양한 산업이 모이는 접점으로 이들 다양한 산업의 관심을 한 몸에 받고 있으며, 일반적인 네비게이션 시스템에서 각종 무선 콘텐츠 및 커뮤니케이션 서비스까지 자동차 속에서 할 수 있는 활동의 범위가 점점 넓어지고 있다[17]. 지역적으로 분산된 컴퓨터, 대용량 스토리지, 거대 실험장치 등과 같은 고성능 자원들을 고성능 네트워크로 연동하여 사용자가 단일 시스템처럼 그리드에 연결된 모든 자원들을 쉽게 사용할 수 있는 정보통신 인프라인 그리드(Grid)는, 차세대 인터넷의 급속

한 발전과 Globus 와 같은 미들웨어의 기술 개발로 성능 저하의 주요 원인이었던 저속의 네트워크 문제와 통합의 어려움이 해결되고 있다. 이에 따라 미국과 유럽은 국가 슈퍼컴퓨터 자원들과 고성능 실험 장비들을 결합하여 네트워크로 연결된 그리드 환경을 구축하고 있으며, 국내에서도 정보통신부가 주관하여 국가 그리드 계획을 발표하는 등 국가 그리드 구축사업이 추진중에 있다[18].

향후 미래의 웹 서비스는 SOAP, WSDL, UDDI, 이외의 추가적인 기술 표준들을 통해 ebXML, RosettaNet 과 같은 B2B 관련 표준과 유사하거나 동일한 수준의 기능을 제공할 것으로 예측되며, 이러한 B2B 관련 표준과의 경쟁과 연계를 통해 보다 다양한 기능을 제공하는 방향으로 진행될 것이고, 시맨틱 웹 기술의 도입을 통한 자동화된 서비스 검색 및 비즈니스 플로우의 구성 등이 가능해 질 것으로 전망된다[19].



Ⅲ. 제안된 시스템

1. 제안된 시스템 모델

본 논문에서 제안하는 시스템은 그림 4와 같다. 기업형 비즈니스를 기업 내 또는 기업 간 동질적이고 이질적인 데이터베이스를 통합 운영하기 위한 시스템은 Management 부분과 Client 부분, 그리고 웹 서비스 부분으로 크게 구성된다.

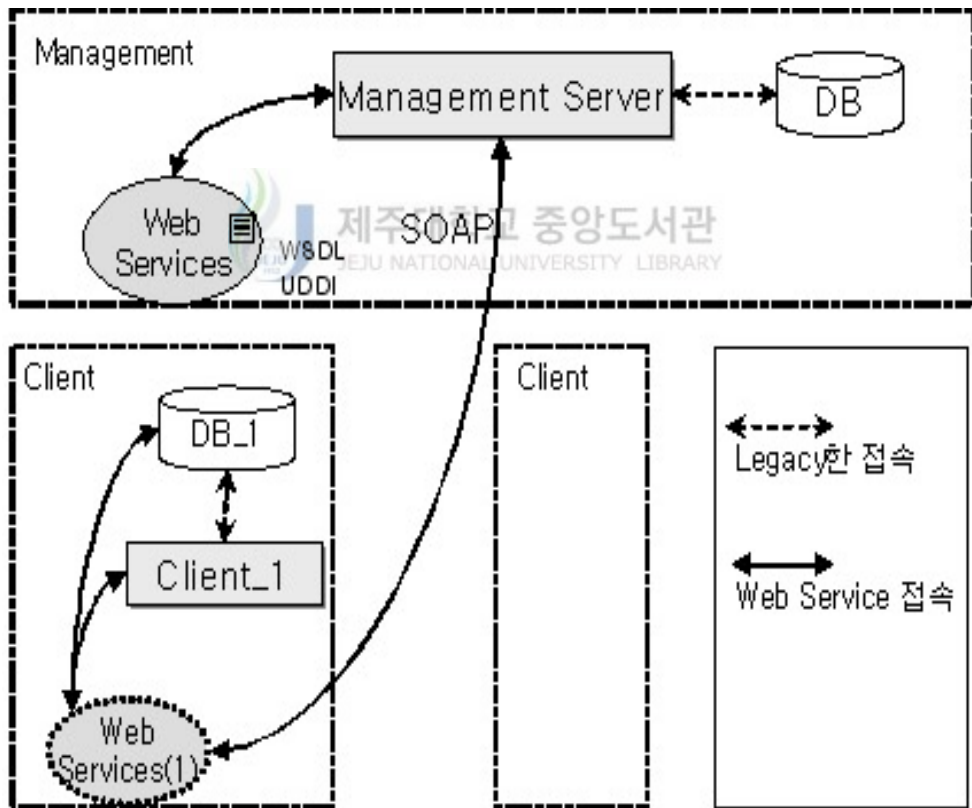


그림 4. 제안된 시스템 모델

2. 제안된 시스템 구조

1) Management

Management 는 기업 내 또는 기업 간 동질적이고 이질적인 데이터베이스를 운영하는 분산 처리기(distributed processor)이다. 즉, 지리적, 물리적으로 분산된 지역 데이터베이스 또는 타 기업의 지역 데이터베이스 자원을 상호 운영하기 위해 웹 서비스 정보를 이용하는 시스템이다. 이 시스템의 구조는 Management Server 를 구동하는 어플리케이션과 데이터베이스 트랜잭션을 위한 웹 서비스 서버, 그리고 각각의 지역 데이터베이스의 위치와 웹 서비스 정보를 보관하기 위한 데이터베이스들로 구성이 된다. Management Server 를 구동하는 어플리케이션에서는 서버가 참조하고 있는 웹 서비스의 역할을 대신 처리하는 Proxy 대리 객체를 생성한다. 생성된 Proxy 대리 객체는 지역 데이터베이스의 Proxy 대리 객체와 SOAP 메시지를 서로 송수신하면서 데이터베이스에서 처리 수행되는 삽입, 수정, 삭제, 검색 등의 트랜잭션을 수행한다. 트랜잭션 처리 시 데이터베이스에 저장되어 있는 지역 데이터베이스의 위치 정보와 지역 데이터베이스가 참조하는 웹 서비스의 위치를 이용하여 Proxy 대리 객체가 통신을 하게 된다.

예를 들어, 지역 데이터베이스의 검색을 수행하는 경우, Management Server 가 참조하고 있는 웹 서비스의 Proxy 객체가 생성되고 해당 지역 데이터베이스 분산 처리기 또한, 웹 서비스 정보를 이용하여 지역 데이터베이스 분산 처리기의 웹 서비스 Proxy 객체를 생성한다. Management Server 에서는 Proxy 객체를 이용하여 지역 데이터베이스 Proxy 객체에 질의(query)를 요청을 한다. Management Server 의 Proxy 객체는 요청된 질의를 SOAP 메시지 형태로 변환을 한다. 변환된 SOAP 메시지는 지역 데이터베이스의 Proxy 객체로 전송이 된다. 그리고 지역 데이터베이스의 Proxy 객체가 트랜잭션 수행 결과 또한 SOAP 메시지로 통해 Management Server Proxy 로 전송이 된다. 이러한 수행이 중앙 집중 데이터베이스를 사용자가 이용하듯이 사용가능하게 한다.

2) Client

Client는 Management Server 의 상호 운영에 직접적으로 연결된 지역 데이터베이스를 관리하는 모듈로써, Management Server 와 상호 운영에 직접적으로 연결된 지역 데이터베이스를 관리하는 분산 처리기(distributed processor)이다. 이 Client 의 구조는 시스템을 구동시키는 어플리케이션과 지역 데이터베이스에 여러 질의를 수행하기 위해 참조되는 웹 서비스 서버, 그리고 데이터베이스로 구성이 된다. 구동 어플리케이션은 데이터베이스에 데이터를 삽입, 수정 삭제, 검색 등의 질의 수행과 Management Server 와 상호 통신을 할 수 있는 Proxy 객체를 생성한다. Proxy 객체를 생성할 때에는 Management Server Proxy 객체로부터 지역 데이터베이스가 참조 하려고 하는 웹 서비스의 위치 정보를 검색하고 검색되어진 위치 정보를 이용해 WSDL 을 다운로드 받아 Proxy 대리 객체를 생성한다.

지역 데이터베이스에는 사용자가 입력한 데이터가 저장되며, 저장되어진 데이터의 모든 연산은 생성된 Proxy 대리 객체에 의해 이루어진다. 물론 Management Server 와의 상호 통신 또한 Proxy 대리 객체를 통해서 이루어진다.

3) Web Service

Web Service 는 기업 내 또는 기업 간 제공되는 웹 서비스 사항을 Publish, Bind 하며, 웹 서비스를 사용하고자 하는 모듈들에 Find 정보를 제공한다. 이를 위해 웹 서비스 표준으로 자리 잡고 있는 WSDL, UDDI 가 자원의 정보와 위치를 기술하고 저장되며 Service Requester 가 Find 하는데 각각 사용된다. 물론 그림4에서는 웹 서비스가 각각 로컬에 운영중인 형태로 도식하였지만 전체 하나의 웹 서비스로 운영하는 것도 가능하다. 서비스간의 통신은 SOAP 이 사용된다.

IV. 시스템 구현 결과

1. 개발환경 및 서비스 환경

개발환경은 .NET Framework Version 1.1, C#, ASP .NET, 그리고 Web Services 기반이며, 제안 시스템 서비스 환경은 <표 4>와 같다.

<표 4> 개발 환경 및 서비스 환경

구분	OS	Func.	DB
Management System	MS 2000 Pro.	Client 추가, 수정, 삭제 해당 Client별로 데이터 검색, 통계 모든 Client 데이터 검색, 통계	Oracle
Client-1	MS XP Pro.	데이터 입력, 수정, 삭제(입고/ 판매)	MsSQL
Client-2	MS XP Pro.	데이터 입력, 수정, 삭제(입고/ 판매)	MySQL

2. Management

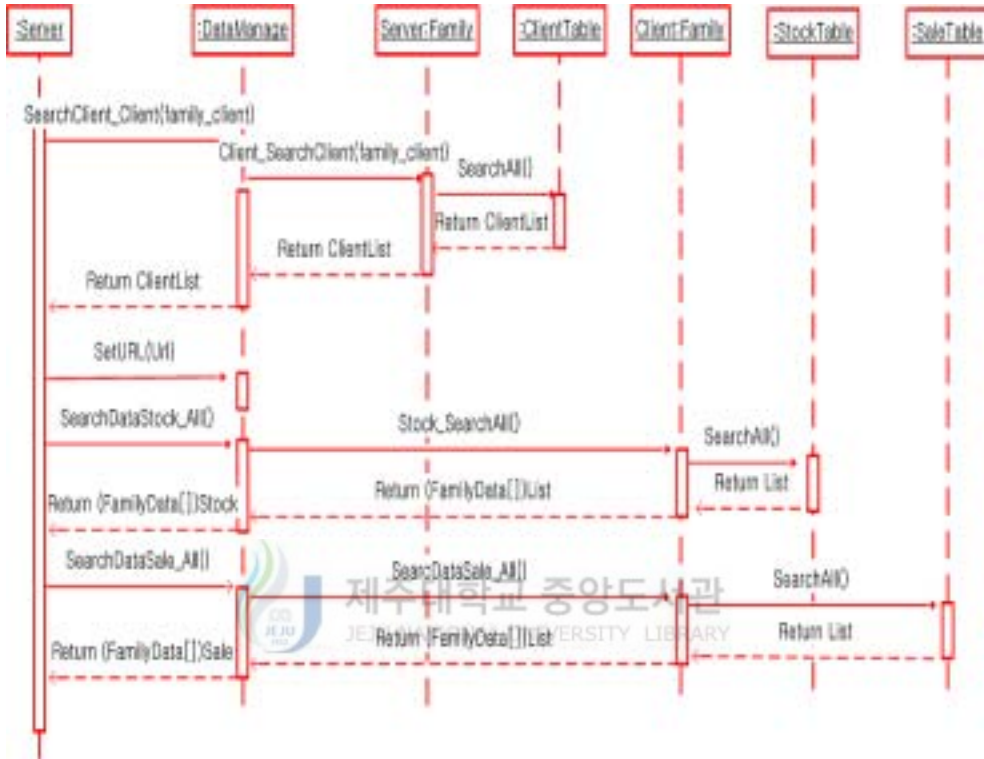


그림 5. Management에서의 트랜잭션 처리도

그림 5는 Management 에서의 Management Server 의 검색 트랜잭션이 수행되는 과정을 나타낸 트랜잭션 처리도이다. DataManage 객체는 생성된 Proxy 객체에 데이터베이스 접근 트랜잭션에 관한 역할을 할 수 있도록 역할을 부여한다. 즉, 지역 데이터베이스 연결과 검색, 삽입, 수정, 삭제 등의 여러 질의에 관한 행위들을 하는 클래스 객체이다. Server:Family 객체, 여기서는 Proxy 객체가 되겠다. 즉, DataMange 객체에서 역할 부여한 행위들을 Clinet:Family Proxy 객체와 SOAP 메시지로 상호 통신을 한다. 그리고 StockTable 객체와 SaleTable 객체는 지역 데이터베이스에서 데이터베이스를 핸들링 할 수 있는 클래스 객체이다.

3. Client

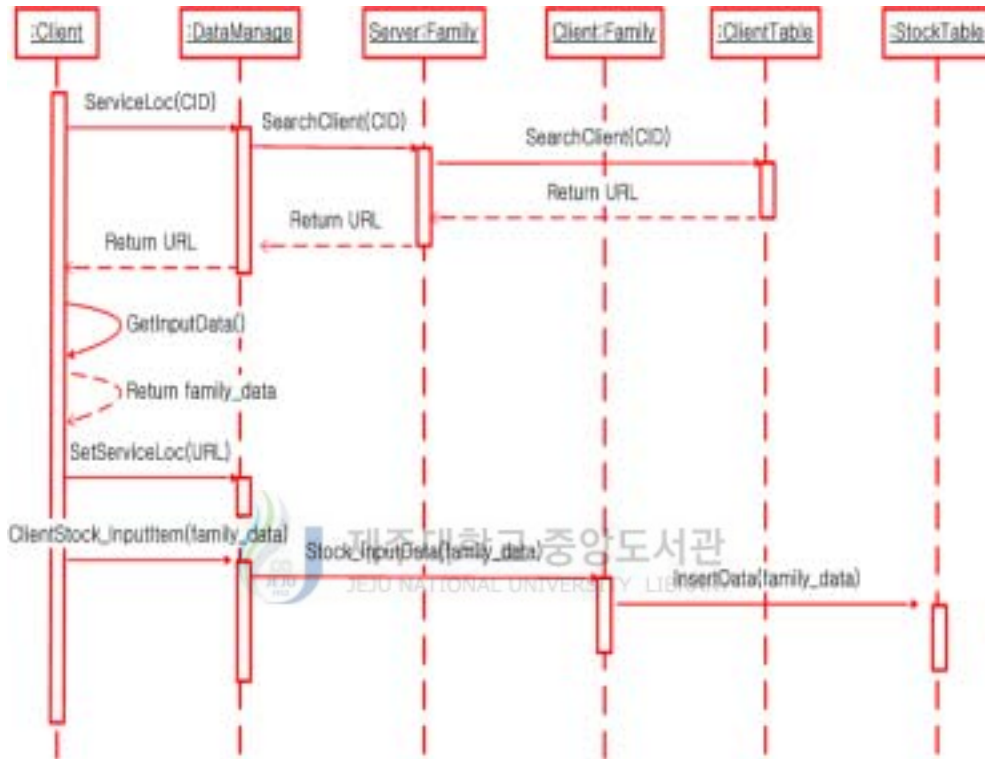


그림 6. Client 에서의 데이터 삽입 트랜잭션 처리도
서의 데이

그림 6은 지역 데이터베이스를 관리하는 Client 에서의 데이터 삽입 트랜잭션 처리도이다. DataManager 객체는 Management Server 에서 관리되는 DataManager 객체와 역할이 같다. 지역 데이터베이스에서는 먼저 Proxy 객체를 생성하기 위해 Client 에 Proxy 객체로부터 지역 데이터베이스가 참조하고 있는 웹 서비스의 위치 정보를 검색한다. 그리고 검색된 웹 서비스의 위치 정보를 이용하여 Proxy 객체를 생성하여 지역 데이터베이스에 Proxy 객체를 사용하여 데이터 삽입의 행위를 수행한다.

4. 웹 서비스 구성 요소

1) SOAP

(1) Sale_SearchAll2 서비스의 SOAP Message



그림 7. Sale_SearchAll2 서비스의 SOAP Message

그림 7은 Sale_SearchAll2 라는 검색 질의에 대한 해당 XML 기반의 SOAP Request 메시지와 Response 메시지이다.

(2) Stock_InputData 서비스 SOAP Message



그림 8. Stock_InputData 서비스 SOAP Message

그림 8은 Stock_InputData 라는 삽입 질의에 대한 해당 SOAP Request 메시지와 Response 메시지이다.

2) WSDL

(1) Stock_InputData의 WSDL

```
<?xml version='1.0' encoding='utf-8' ?>
<definitions xmlns:http='http://schemas.xmlsoap.org/wsdl/http/'
xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/' xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xmlns:tns='http://tempuri.org/' xmlns:soapenc='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:tm='http://microsoft.com/wsdl/mime/textMatching/'
xmlns:mime='http://schemas.xmlsoap.org/wsdl/mime/' xmlns='http://schemas.xmlsoap.org/wsdl/'
targetNamespace='http://tempuri.org/'>
<types>
<schema elementFormDefault='qualified' targetNamespace='http://tempuri.org/'>
<element name='Stock_InputData'>
<complexType>
<sequence>
<element minOccurs='0' maxOccurs='1' name='data' type='tns:FamilyData' />
</sequence>
</complexType>
</element>
<complexType name='FamilyData'>
<sequence>
<element minOccurs='1' maxOccurs='1' name='ID' type='xsd:int' />
<element minOccurs='0' maxOccurs='1' name='ClientID' type='xsd:string' />
<element minOccurs='1' maxOccurs='1' name='Date1' type='xsd:dateTime' />
<element minOccurs='1' maxOccurs='1' name='Date2' type='xsd:dateTime' />
<element minOccurs='0' maxOccurs='1' name='Kind' type='xsd:string' />
<element minOccurs='0' maxOccurs='1' name='Barcode' type='xsd:string' />
<element minOccurs='1' maxOccurs='1' name='Price' type='xsd:int' />
<element minOccurs='1' maxOccurs='1' name='Amount' type='xsd:int' />
<element minOccurs='0' maxOccurs='1' name='Etc' type='xsd:string' />
</sequence>
</complexType>
</element>
</complexType />
</schema>
</types>
<message name='Stock_InputDataSoapIn'>
<part name='parameters' element='tns:Stock_InputData' />
</message>
<message name='Stock_InputDataSoapOut'>
<part name='parameters' element='tns:Stock_InputDataResponse' />
</message>
<portType name='FamilySoap'>
<operation name='Stock_InputData'>
<input message='tns:Stock_InputDataSoapIn' />
<output message='tns:Stock_InputDataSoapOut' />
</operation>
</portType>
<binding name='FamilySoap' type='tns:FamilySoap'>
<soap:binding style='document' transport='http://schemas.xmlsoap.org/soap/http' />
<operation name='Stock_InputData'>
<soap:operation soapAction='http://tempuri.org/Stock_InputData' style='document' />
<input>
<soap:body use='literal' />
</input>
<output>
<soap:body use='literal' />
</output>
</operation>
</binding>
<service name='Family'>
<port name='FamilySoap' binding='tns:FamilySoap'>
<soap:address location='http://209.259.207.64/FamilyWeb/Family.asmx' />
</port>
</service>
</definitions>
```

그림 9. Stock_InputData의 WSDL

그림 9는 지역 데이터베이스에서 데이터를 입력시 사용되는 Stock_InputData 서비스, 즉 삽입 질의에 대한 명세인 WSDL 이다.

(2) Sale_SearchAll2의 WSDL

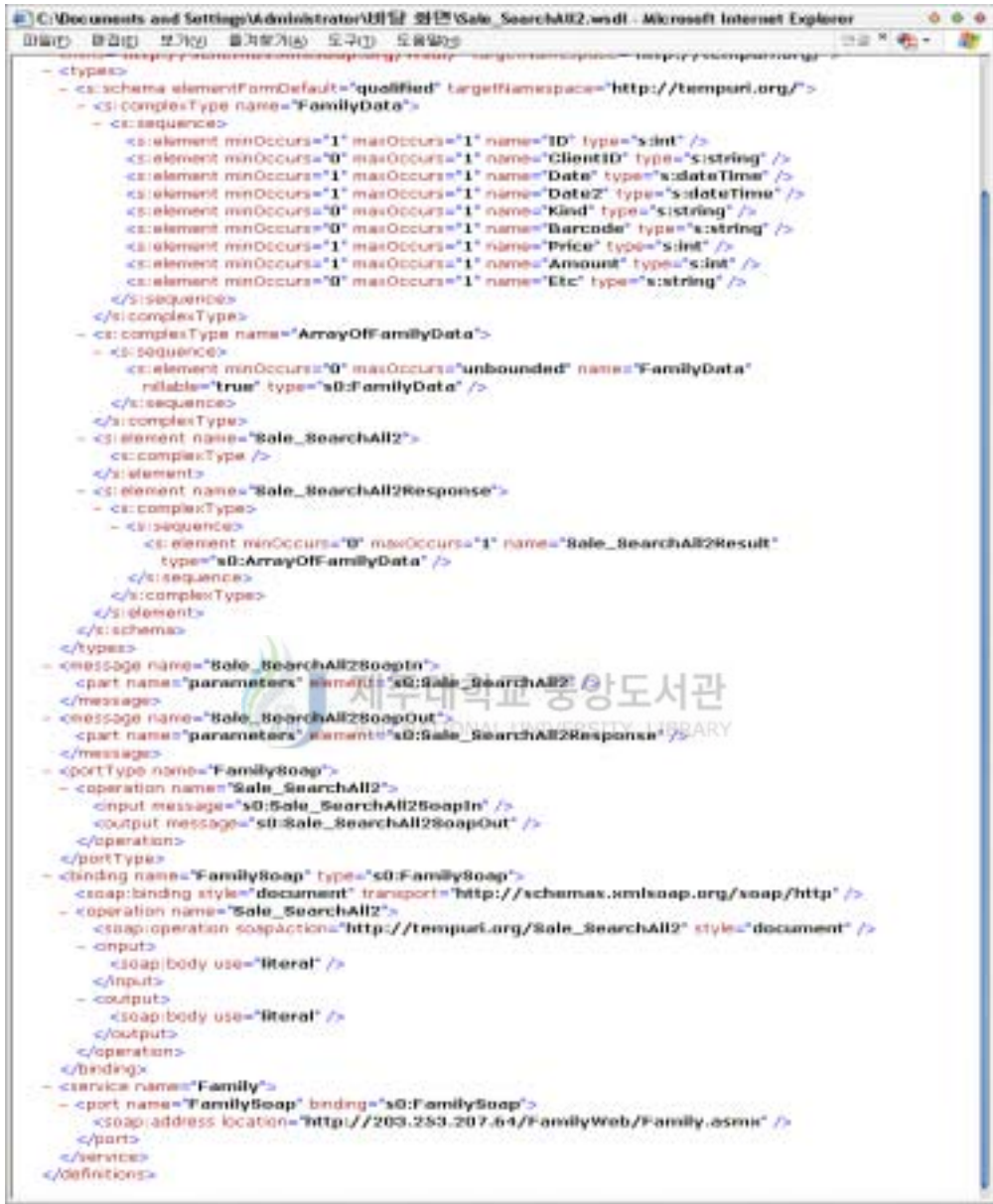


그림 10. Sale_SearchAll2의 WSDL

그림 10은 Management 에서 서버 프로그램 상에서 검색 질의에 해당하는 Sale_SearchAll2 서비스의 WSDL 이다.

3) UDDI



그림 11. 웹 서비스 항목

Management 상의 Management Server 와 지역 데이터베이스인 Client 가 동적 참조할 수 있는 웹 서비스 항목들이다. 즉, UDDI 를 말한다. 이를 이용하여 서비스 개발자는 지원하는 웹 서비스의 종류를 검색하여 시스템을 개발할 수 있다.

5. 상호운영 트랜잭션 결과

1) Management Server

ClientID	날짜	종류	바코드	가격	수량	Etc
kodol	2003-09-13	화장지	1234555	23000	5	첫 상품 입고완료
kodol	2003-09-27	츄파춥스	12300	200	1000	이벤트상품
kodol	2003-09-27	홀런볼	12	700	100	새로나온 홀런...
kodol	2003-09-27	쏘세지	13	500	1000	통통해요~
kodol	2003-09-27	화장지	1234555	1500	50	--
kodol	2003-09-27	쏘세지	500	500	200	큰사발
kodol	2003-09-27	망고주스	800	3000	70	롯데
kodol	2003-09-27	망고주스	800	3000	100	냉장보관
kodol	2003-09-27	화장지	1234555	1500	100	알록달록
kodol	2003-09-27	샌드위치	111	2000	20	냉장보관
kodol	2003-09-27	코카콜라(페트)	700	1000	100	Coke
kodol	2003-09-27	삼각김밥(소...	222	500	30	삼각김밥세일~!
kodol	2003-09-27	망고주스	800	3000	50	500
kodol	2003-09-27	홀런볼	12	700	5	맛있다~
kodol	2003-10-06	볼펜	1111A...	500	50	사무용 볼펜
kodol	2003-11-02	망고주스	800	3000	50	맛있는 망고주스
kodol	2003-09-27	아이스크림	577	1000	10	이씨판매완료
kodol	2003-09-27	아이스크림	577	1000	20	배써가한번에사...
kodol	2003-09-27	스타킹	100	1500	2	커피색
kodol	2003-09-27	아이스크림	577	1000	5	맛있다
kodol	2003-11-06	망고주스	800	1000	2	색 다른 맛

그림 12. Management Server에서 검색 질의를 수행한 후의 결과

Management Server 에서의 어플리케이션 모습이다. 위 그림은 특정 지역 데이터베이스가 관리하고 있는 데이터베이스의 데이터를 Management Server 에서 검색 질의를 수행한 후의 결과이다.

FamilyMartServer

클라이언트 : kodol

날 짜 : 2003-11-06 오전 12:00:00 ~ 2003-11-06 오전 12:00:00

ClientID	날짜	종류	바코드	총액	수량	Etc
kodol	2003-09	화장지	1234555	115000	5	첫 상품 입고완료
kodol	2003-09	츄파춥스	12300	200000	1000	이벤트상품
kodol	2003-09	홀런볼	12	73500	105	새로나온 홀런...
kodol	2003-09	쏘세지	13	500000	1000	통통해요~
kodol	2003-09	화장지	1234555	225000	150	---
kodol	2003-09	쏘세지	500	100000	200	큰사발
kodol	2003-09	망고주스	800	660000	220	롯데
kodol	2003-09	샌드위치	111	40000	20	냉장보관
kodol	2003-09	코카콜라(페트)	700	100000	100	Coke
kodol	2003-09	삼각김밥(소...	222	15000	30	삼각김밥세일~!
kodol	2003-10	볼펜	1111A...	25000	50	사무용 볼펜
kodol	2003-11	망고주스	800	150000	50	맛있는 망고주스
kodol	2003-09	아이스크림	577	35000	35	이씨판매완료
kodol	2003-09	스타킹	100	3000	2	커피색
kodol	2003-11	망고주스	800	2000	2	색 다른 맛

제주대학교 중앙도서관
JEJU NATIONAL UNIVERSITY LIBRARY

그림 13. 지역 데이터베이스의 검색 질의 수행 후 통계

그림 13은 지역 데이터베이스가 관리하고 있는 지역 데이터베이스의 검색 질의 수행 후, 통계 기능을 수행 했을 때, 입력 데이터 항목별로 통계를 수행한 결과이다.

2) Client

(1) Client 접속

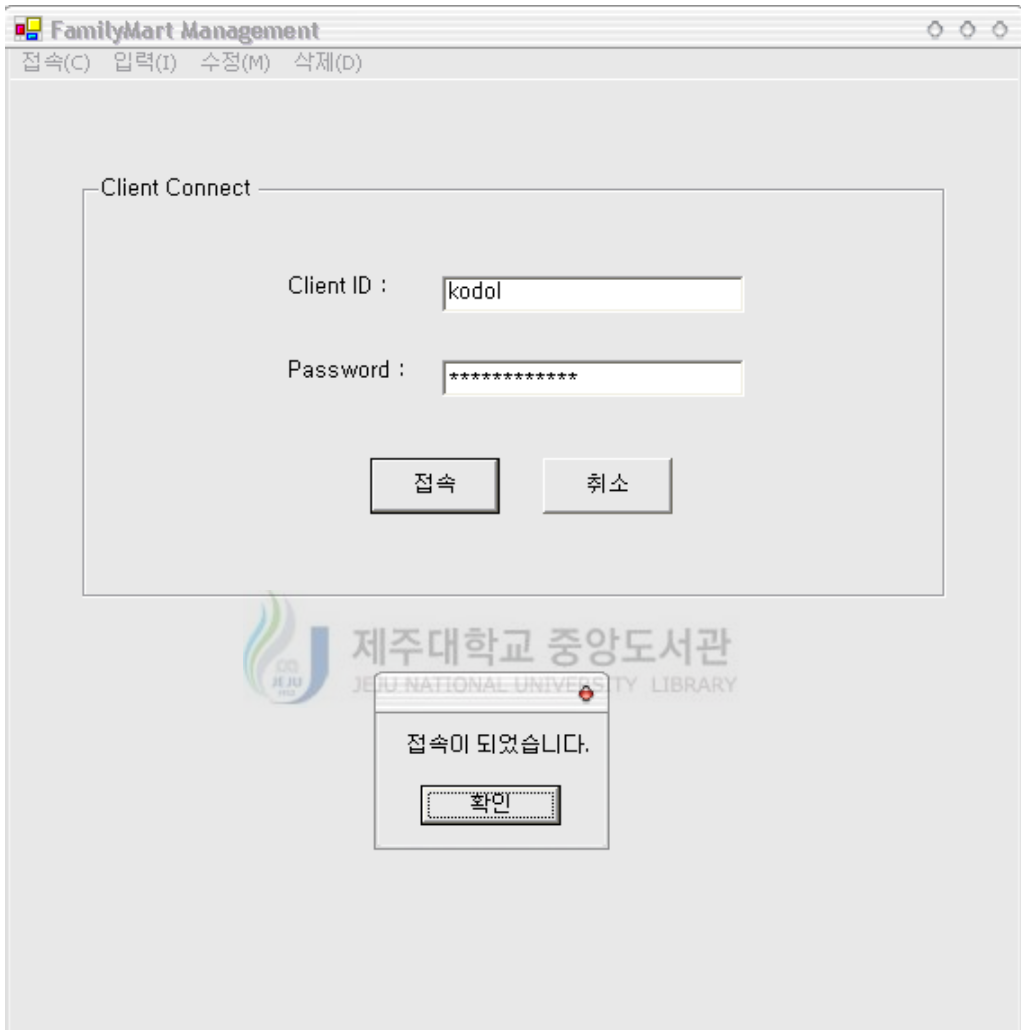


그림 14. Client의 접속

그림 14는 지역 데이터베이스의 Client 가 활성화되기 위해 접속을 하는 부분이다. 즉, Client 는 Management Server 로부터 웹 서비스의 위치 정보를 알기 위해 로그인을 한다. Management Server 는 Client 로 해당 웹 서비스의 위치 URL 을 리턴 시켜주고, 받은 URL 을 참조 하여 Client 는 Proxy 객체를 생성하여 지역 데이터베이스에 관한 여러 연산을 수행한다.

(2) 삽입 질의를 수행

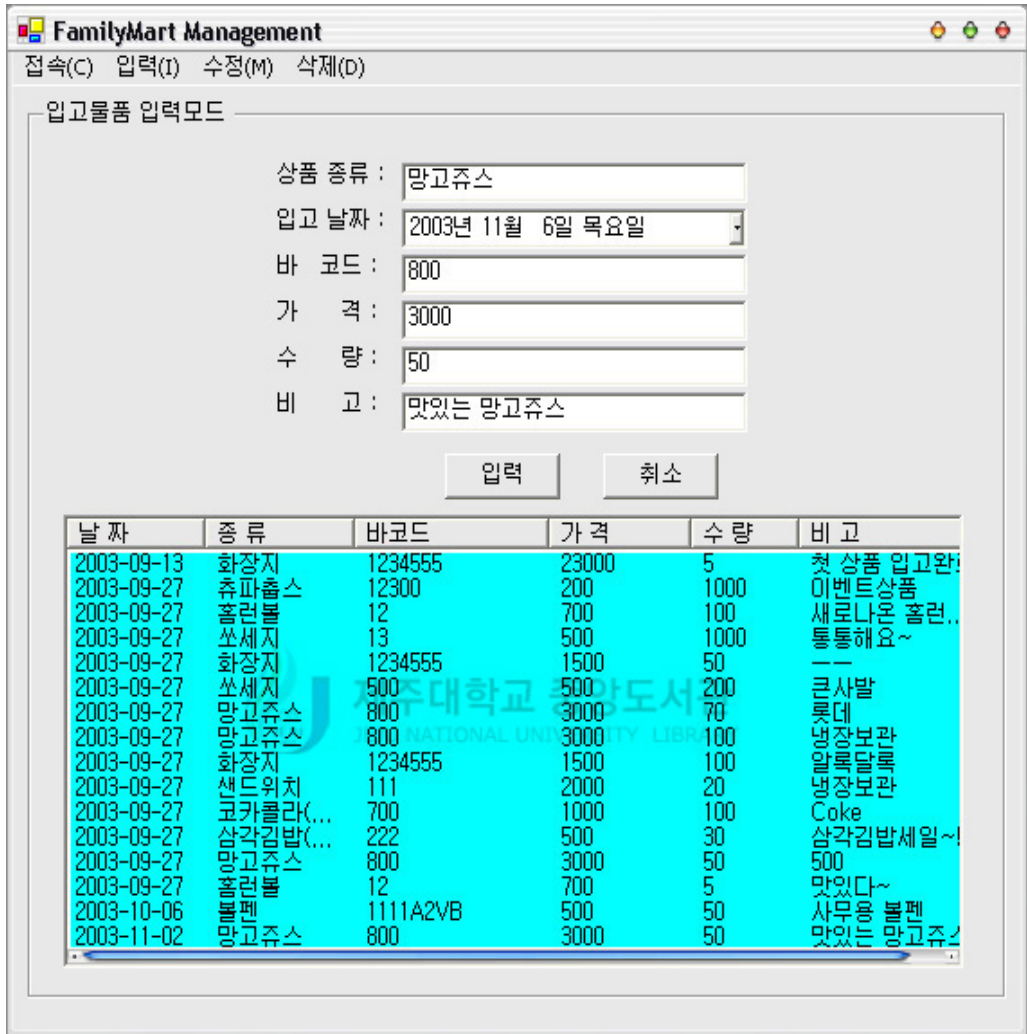


그림 15. Client에서 삽입연산 수행

그림 15는 Client 에서 지역 데이터베이스에서 할 수 있는 여러 연산중에 삽입 연산을 수행한 모습이다. 즉, 삽입을 하면 삽입 항목이 FamilyData Array 형식의 SOAP 메시지로 지역 데이터베이스 시스템인 Client 가 참조하고 있는 웹 서비스 서버에 전송이 되며, 받은 SOAP 메시지는 디코딩되어 지역 데이터베이스에 삽입을 하게 된다.

V. 결론 및 향후 연구

본 논문에서는 기업 내에서 또는 기업 간 비즈니스와 관련된 데이터들을 관리하고 유지하기 위해 웹 서비스 기반의 분산되면서 이질적인 데이터베이스를 상호 운영할 수 있는 시스템을 제안하고 설계하였다. 제안된 시스템은 자사의 지역 데이터베이스를 중앙에서 통합 관리할 수 있는 기반이 되며 서로 다른 운영 환경하의 이질적 분산 데이터베이스를 활용하고 관리할 수 있는 방안을 제시하였다. 또한 제안된 시스템은 기업 내 또는 기업 간 트랜잭션 처리를 하기 위해 분산 데이터베이스 연결은 기존의 연결 표준인 ODBC, OLE-DB, ADO, CORBA 기반을 배제하고 이를 웹 서비스 기반인 HTTP 를 사용하게 되며, 분산 데이터베이스의 단점으로 지적되고 있는 이미 존재하고 있는 데이터베이스에 기반 한 응용 프로그램에 대한 투자비용의 감소와 이질적 분산 데이터베이스를 통합하는데 있어서 조직적인 어려움들의 해결 가능성, 그리고 전체적인 분산 데이터베이스의 상호 호환성을 증가시켰다.

향후 연구과제는 분산 데이터베이스 시스템과 제안된 시스템 모델간의 정량적인 성능 평가 및 순수한 데이터베이스 관리 시스템으로의 웹 서비스에 대한 심도있는 연구가 필요하다.

참 고 문 헌

- [1] Reash Trivedi, "Web Service Architecture Models", RCG IT, 2002, 4
- [2] eXtensible Markup Language(XML) 1.0
<http://www.w3.org/TR/2000/REC-xml-20001006/>
- [3] James Snell 외 2인, Programming Web Services with SOAP, O'Reilly & Associates, Inc. 2002
- [4] M.Birdeck 외 Professional XML, Wrox Press, 2002
- [5] UDDI.org : UDDI Executive White Paper, Nov. 2001
- [6] 김선영, 김지영, 김행곤, 웹 서비스 기반의 e-Business 시스템 개발을 위한 아키텍처 메타모델 및 설계 패턴 식별, 정보과학회 학술대회, Vol. 30. No. 1. 2003
- [7] Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts, Fourth Edition", McGraw-Hill, 2003
- [8] Paul Allen, Semantic Web Service Architecture", American Association for artificial Intelligence, 2001
- [9] 박성진, XML데이터베이스, 인터넷정보학회지 제 2권 3호(2001. 9)
- [10] Simple Object Access Protocol (SOAP) 1.1
<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [11] Web Services Description Language (WSDL) 1.1
<http://www.w3.org/TR/2001/NOTE-wsdl>
- [12] UDDI Version 2.04 API Specification UDDI Committee Specification, 19 July 2002
http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm#_Toc25137690-20010315/
- [13] David Bell, et al. Distributed Database System Addison-Wesley, 1992
- [14] M. Tamer Ozsu, Principles of Distributed Database System, Prentice-Hall, 1999

- [15] United Modeling Language(UML), Version 1.5, Complete specification,
<http://www.omg.org/docs/formal/03-03-01.pdf>
- [16] Fabio Casati, Ming-Chien Shan, "Models and Languages for Describing
and Discovering E-Services", Tutorial, Semantic Web Working
Symposium, Stanford, USA, July 2001.
- [17] 시장분석, 텔레메틱스, Software Market News, 한국소프트웨어진흥원
(2002. 11)
- [18] 김동균, 이필우, 황일선, 그리드 컴퓨팅, 정보과학회지, 제 20권 2호(2002. 2)
- [19] 이경하, 이규철, 웹 서비스의 향후 발전 방향, 정보처리학회지, 제 9권 4호
(2002. 7)

