

碩士學位論文



濟州大學校 大學院

工學科

高 庸 齊

2005年 12月

**A Design and Implementation of
Integrated Information System based on
Web Services**



Department of Computer Engineering

GRADUATE SCHOOL

CHEJU NATIONAL UNIVERSITY

Supervised by Professor Kwak, Ho Young

指導教授 郭 鎬 榮

高 庸 齊

論文 工學碩士 論文 提出



高庸齊 工學碩士學位 論文 認准

審查 委員長 _____ 印

委 員 _____ 印

委 員 _____ 印

濟州大學校 大學院

2005年 12月

목 차

Summary	1
I. 서론	2
II. 관련연구	3
1. 분산객체기술	3
2. 웹 서비스	6
3. 기존 시스템 현황	14
III. 시스템 분석 및 설계	16
1. 제안 시스템 기본 구조	16
2. 제안 시스템 설계	17
1) Server Application	18
2) 정보통합시스템	27
3) Client Application	28
IV. 시스템 구현	29
1. 제안 시스템 개발 환경	29
2. 제안 시스템 구현 결과	30
1) Server Application 개발	30
2) 정보통합시스템 개발	36
3) Client Application	38
V. 결론	40
참고문헌	41

Summery

One of the main purpose of the internet information systems integration is to provide one-stop services to the customers. However, the service of the current system is inefficient because the systems are managed independently and informations are not shared among the related organizations. At the integrated system, the user-friendly services could be provided by sharing informations between the administrative organizations. This study suggests the integration method of the information systems based on the webservices technology. We design and implement an integration system which allows to integrate many kinds of servers integrally using XML-based webservices which is free from platforms or development languages.



I. 서 론

정보기술(Information Technology)은 컴퓨터의 탄생을 시초로 계속 진화하고 있으며, 그 흐름은 당시 기술과 사회적 필요에 맞는 타당한 방향으로 발전해 왔다. 정보기술 패러다임은 1970년대의 대규모 메인프레임 시스템에서 클라이언트 서버 환경으로, 그리고 다시 분산시스템 환경으로 진화하였다. 최근에 이르러서는 본격적인 서비스 중심의 환경으로 다시 진화하고 있다[1].

이와 때를 같이 하여 구조 및 조직과 행정서비스의 미래 패러다임을 완성시키는 중요한 국가적 사업인 전자정부가 추진되었다. 전자정부의 추진은 전산 보급, 인터넷 보급 및 통합과 연계라는 3단계로 발전되었으며 현재 그 마지막 단계인 통합과 연계의 컨셉이 가장 큰 이슈로 부각되고 있다.

이는 서비스를 제공하는 행정시스템이 백엔드(Back End) 즉 데이터베이스의 통합과 연계가 주가 되어야 하는 시스템 통합에 있어서 그것보다는 프론트엔드(Front End) 즉 사용자(End User)가 직접 대하게 되는 환경에 있어서의 서비스만이 강조되고 있는 현실에 사용자들이 여러 불편을 얻고 있기 때문이다. 기존의 상황을 살펴보면 ERP(Enterprise Resource Planning)를 위시하여, CRM(Customer Relationship Management), SCM(Supply Chain Management), KMS(Knowledge Management System), 기존의 레거시 등 열거하기에도 벅찰 정도로 많은 애플리케이션을 통하여, 다양한 업무들은 처리되며 이렇게 생성되고 처리된 정보들은 부문별로 분리되어 보관, 각 부문의 담당자들만이 손쉽게 정보에 접근하고 활용할 수 있는 것이었다. 이는 다양한 도구 및 전문성의 부재로 정보연계 및 공유에 있어서 큰 장애물이 되었다.

정보 통합 연계의 실질적인 목표라고 할 수 있는 원스톱(One Stop) 행정 서비스 제공을 위해서는 각 부서 내부의 통합이 선행되어야 한다. 이러한 내부 통합의 기반 위에 각 부서 사이의 연계가 유기적으로 이루어지고, 유관 기관들과의 정보 공유에 있어서 원활한 흐름을 보일 때, 진정한 의미의 원스톱 행정 서비스의 제공이 현실화 될 수 있으며 이것이 정보 통합 연계의 목표인 것이다[2].

본 논문에서는 이러한 목표 달성을 위해서 분산 객체 기술과 같은 능력을 가지면서도 플랫폼이나 개발언어에 제약을 받지 않는 웹 서비스 기술(Web Services Architecture)을 이용하여 다수의 정보시스템을 통합하고, 통합된 새로운 정보를 제공할 수 있도록 정보통합시스템을 구현할 필요가 있다. 따라서 본 논문에서는 사용자 서비스 중심의 편의성과 정보의 효율적인 통합을 위해 정보통합시스템을 설계·구현하고자 한다.

본 논문의 구성은 1장 서론에 이어, 2장에서는 논문의 이론적 배경에 관한 내용을 기술하였다. 3장에서는 제안 시스템에 대한 분석과 설계, 4장에서는 설계한 시스템 구현 결과를 고찰하였다. 마지막으로 5장에서는 결론에 대한 내용을 기술하였다.

II. 관련연구

분산된 정보시스템의 통합에 대한 방안으로 CORBA(Common Object Request Broker Architecture), Java RMI(Remote Method Invocation), DCOM(Distributed Component Object Model) 등의 분산 객체 기술을 이용하여 많은 연구가 진행되고 있다. 이러한 분산객체기술은 운영체제나 개발언어에 종속적이기 때문에 다양한 플랫폼을 사용하고 있는 정보시스템을 통합 하는 데는 많은 문제점이 발생하고 있다. 따라서 본 논문은 분산 환경에서의 통합 조건을 만족하고 가장 서비스 중심의 환경에 적합한 웹 서비스 기술을 활용하여 정보시스템의 통합을 보여주고 통합된 정보시스템이 새로운 정보서비스를 제공함을 보여 준다. 이러한 통합방법으로 분산객체 기술을 사용하며 많은 분산객체 기술 중 프레임에 독립적인 웹 서비스 기술을 활용한다.

1. 분산 객체 기술

1) 분산객체기술의 개념

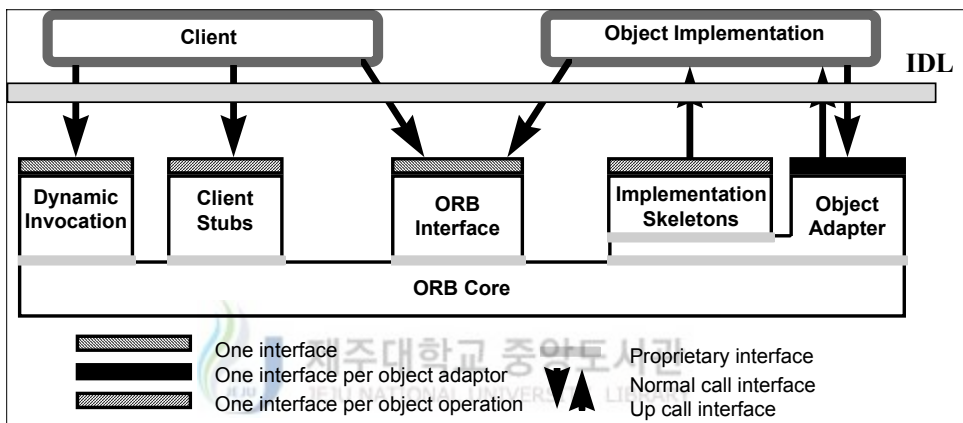
특정 컴퓨터에서 동작하는 객체를 다른 컴퓨터에서 작동하는 클라이언트 어플리케이션들이 사용할 수 있도록 하는 기술이다. 이러한 분산객체 기술은 하나의 기계에서 동작하는 경우에는 필요하지 않은 일들이 클라이언트와 서버로 구성된 분산 환경에서는 반드시 고려되어야 하는데, 이 경우 우선적으로 클라이언트가 자기가 원하는 서버의 연산이 어느 곳에 있는지 알아야한다. 위치가 파악되면 서버의 연산을 호출해야 하는데, 이 때 클라이언트의 연산 호출을 이해한 서버가 처리 결과를 클라이언트에게 전달하기 위해서는 클라이언트와 서버 사이에 공통된 약속이 필요하다. 이러한 약속을 특정 개발자가 나름대로 정의해 사용하면 본인이 개발한 프로그램에서만 적용할 수 있고, 다른 개발자가 이미 개발한 프로그램 또는 이후 개발자가 개발하는 프로그램들과는 통신할 수 없게 된다. 이러한 문제점을 해결 할 수 있는 기술이 분산객체의 개념이다. 따라서 분산객체 기술은 분산기술과 객체지향기술을 접목한 것이다.

분산객체 기술의 핵심은 객체지향 개념을 도입해 구현한 객체들 간의 통신을 원활하게 만들어 주는 일종의 '소프트웨어 버스'라 할 수 있다. 이러한 분산객체 기술로는 CORBA, DCOM등이 널리 알려져 있다. CORBA는 OMG(Object Management Group)에 의해 표준화되어 있고 DCOM은 마이크로소프트사가 제안하고 있다. 또한, RMI(Remot Method Invocation)를 활용하는

EJB(Enterprise Java Bean)는 선마이크로 시스템즈에서 제안하고 있다.

2) CORBA

CORBA [3,6]는 분산객체시스템의 표준구조(Standard Architecture)이다. 분산객체시스템은 분산되고 상호작용하는 객체들의 콜렉션(을 허용한다. [그림1]은 CORBA의 구성도와 [그림2]는 CORBA 클라이언트와 객체 구현사이의 통신의 구조를 보이고 있다.



[그림 1] CORBA 구성도

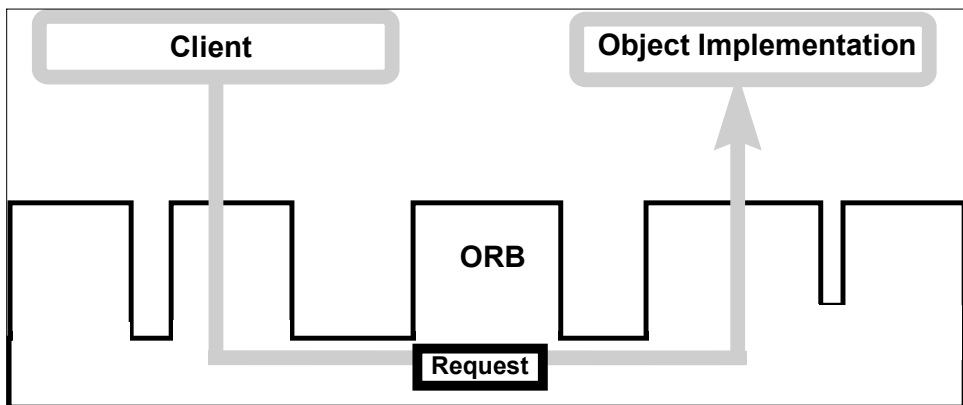


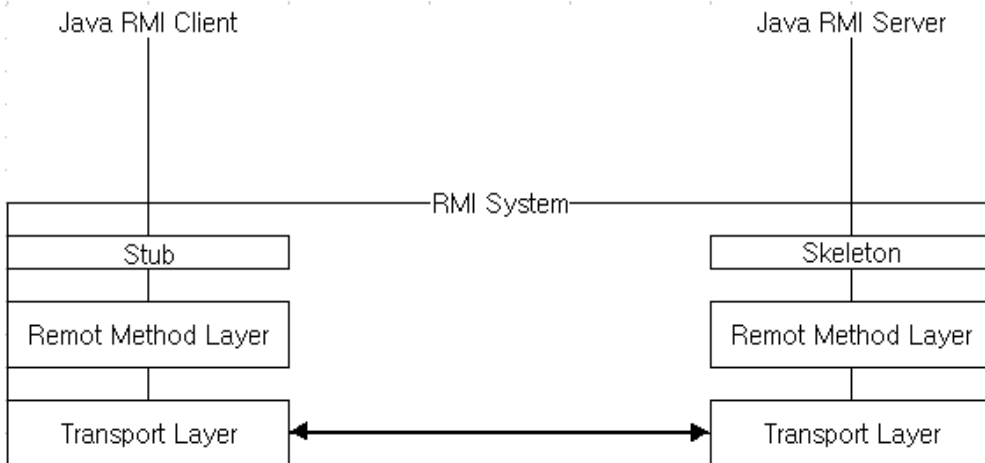
그림 2 CORBA클라이언트와 객체 구현사이의 통신

CORBA는 클라이언트와 서버 객체 간의 통신을 위해 개념화한 소프트웨어 버스라고 생각하면 된다. 외부에 노출된 서버의 인터페이스는 CORBA의 표준 인터페이스 언어인 IDL로 표현되며, IDL로 표현한 인터페이스의 내용을 IDL 컴파일러를 이용해 자바, C++ 등의 언어로 변환해 바인딩할 수 있다. 이러한 방법으로 CORBA가 서로 다른 언어의 한계를 뛰어 넘는다. 이를 통해 프로그래머들은 가장 적절한 언어를 선택해 어플리케이션을 작성할 수 있게 된다. 하지만 CORBA는 제한된 표준화가 제품간의 상호 운영성을 보장할 수 없다는 단점이 있다.

3) Java RMI

Java RMI[7]는 네트워크상으로 다른 자바 어플리케이션과 통신할 수 있는 자바 어플리케이션을 만들기 위해 사용된다. 간단히 설명하면 RMI는 어떤 어플리케이션에서 멀리 떨어진 곳에 위치한 어플리케이션의 메소드를 호출하거나, 또는 어플리케이션의 변수에 접근할 수 있게 하고, 객체를 네트워크를 통해 주고 받을 수 있게 하는 것이다. 또한 이렇게 멀리 떨어진 곳에 위치하는 어플리케이션의 경우 완전히 다른 시스템과 다른 자바 환경에서 동작할 수도 있다는 것이다. RMI는 분산자바 객체 사이의 통신을 위한 메커니즘으로써 간단한 소켓 접속보다 좀더 정교한 메커니즘이다.

RMI를 사용하면 어떠한 프로토콜을 사용해서 통신할 것인지를 미리 알지 못하더라도 다른 자바 프로그램과 통신을 할 수 있다[7]. 하지만 RMI는 자바 객체의 직렬화(Java Object Serialization)에 크게 의존하기 때문에, 객체를 일종의 스트림으로 마샬링(Marshaling) 한다. 이러한 자바 객체의 직렬화는 자바의 특징 가운데 하나이기 때문에, 자바 RMI 서버 객체와 클라이언트 객체가 모두 자바일 경우에만 사용할 수 있다. 따라서 다양한 언어로 개발된 여러 시스템을 통합하기에는 적합하지 않다. [그림 3]에서는 RMI시스템상에서 클라이언트와 서버사이의 통신에 대한 구조를 보여주고 있다.



[그림 3] RMI 구조도

4) DCOM/COM+

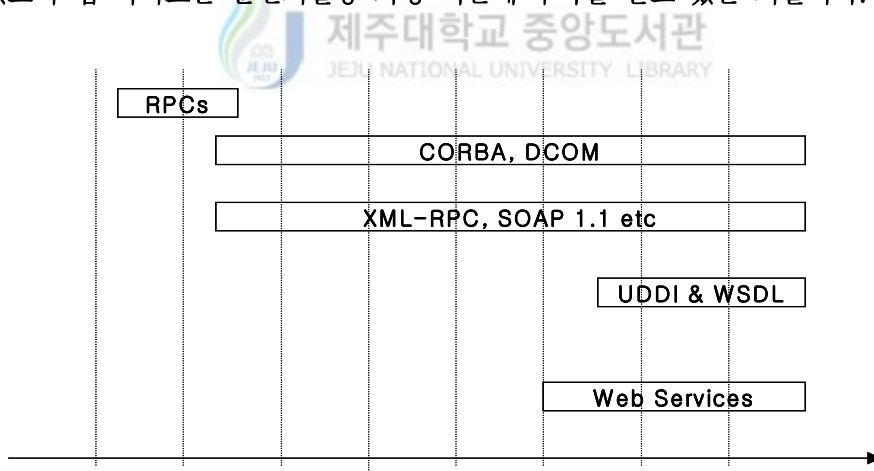
윈도우 환경에서의 분산객체기술은 지난 OLE(Object Linking and Embedding) 기술이 소개된 이후 끊임없이 발전해 COM(Common Object Model) [8], DCOM 환경이 개발되었고, 현재 COM+ 기술이 개발되어 있다. COM+는 3티어 모델 기반의 분산환경에서 비즈니스 로직 처리 부분을 완전히 분리시켜 개발, 실행할 수 있는 미들웨어[MTS(Message Transfer System), MSMQ(Microsoft Message Queuing Services)]를 공급함으로써 보다 쉽고 편리하게 윈도우 분산 환경을 구축할 수 있는 지름길을 제시하고 있다. 하지만 DCOM/COM+ 분산객체기술은 윈도우 환경에서만 그 의미를 가질 수 있다는 단점이 있다. 이는 다양한 이기종 시스템간의 통합에는 적합하지 않은 기술이다.

2. 웹 서비스

웹 서비스[9,10]는 다양한 정보기술 환경 내에서 비즈니스 통합 및 상호 연계라는 필요성 때문에 대두되었다. 서로 다른 하드웨어, 운영 체제, 프로그래밍 언어, 데이터베이스와 여러 형태의 데이터들이 있는 기업 내 및 기업 간의 환경 하에서 표준 기반의 확산 가능한 통합된 환경을 만들기 위해서는 이를 포괄하는 새로운 표준의 필요성이 대두되었고, 이를 구현하여 표준화된 XML 기반의 서비

스 수준 통합의 방법을 제시한 것이 웹 서비스이다.

웹 서비스의 역사는 웹 서비스에 대한 긍정적인 평가들에 비해 상대적으로 짧다. 2000년 5월에 SOAP(Simple Object Access Protocol) 1.1 표준안이 발표되었으며 그 이후로, UDDI(Universal description Discovery and Integration), WSDL(Web Services Description Language) 표준들이 연이어 발표되었다. 비록 웹 서비스의 직접적인 시작 역사는 짧으나, 웹 서비스는 기존의 분산객체기술들에 그 진화의 배경을 두고 있다. 웹 서비스는 기존의 분산객체기술인 CORBA, DCOM, RMI이 진화된 것으로 보아도 무방할 것이다. 기존의 분산객체 기술들은 그간 복잡한 환경설정과 동기식 연결방식, 방화벽과의 충돌, 특정 벤더 수준의 미약한 지지기반 등이 단점으로 지적되어 왔다. 웹 서비스는 상대적으로 단순한 환경설정을 필요로 하고 비동기식 연결을 지원하며, HTTP(Hyper Text Transfer Protocol) 포트를 이용함으로써 방화벽 충돌 문제 등을 해결하였다. 글로벌한 벤더들의 강력한 지지기반을 확보하고, 이를 국제표준화 기구를 통하여 표준화함으로써 그 기반을 넓혀가고 있다. 이러한 웹 서비스의 장점들로 인하여 웹 서비스는 오늘날 현실적이고 가장 유망한 분산처리 기술로서 주목받고 있는 것이다. [그림 4]는 분산객체기술의 발전방향을 보여주고 있으며 웹 서비스는 분산기술중 가장 최근에 주목을 받고 있는 기술이다.



[그림 4] 분산객체기술에서의 웹 서비스 위치

1) 웹 서비스 기술의 주요특징

웹 서비스는 "통합"과 "협업"을 위한 분산 솔루션으로서 다음과 같은 특징들이 있다.

○ 인터넷상에서 제공된다

웹 서비스는 ASP처럼 인터넷 상에서 제공된다. ASP와 다른 점은 사용자가 자신이 웹 서비스를 사용하고 있는지를 인식하지 못한다는 점이다.

○ 인터넷 표준을 지원한다.

웹 서비스는 HTTP, TCP/IP등의 표준뿐만 아니라, 차세대 인터넷 표준인 XML(Extensible Markup Language), UDDI, SOAP, WSDL들을 지원한다. 이를 통해 플랫폼에 독립적이며, 상호 운용성(Interoperability)이 높은 서비스가 가능하게 된다.

○ 비즈니스 로직을 포함하고 있다.

웹 서비스는 기업의 가치사슬(Value Chain)내에서 발생할 수 있는 특정 태스크(예, 재고관리, 주문 등)의 비즈니스 로직 포함하고 있다. 비즈니스 로직은 특정 기업을 위해 개별화(customizing) 된 것이 아니라 모든 기업이 공통적으로 사용하는 표준화된 비즈니스 로직이다. 비즈니스 로직을 보유하고 있기 때문에 웹 서비스 관련 팩터가 변경되었을 때 프로그래밍이 아닌 단순 조작으로 변화에 대처할 수 있다.

○ 객체기술에 기반을 둔 컴포넌트이다.

컴포넌트이므로 산업에 구별 없이, 어떤 기업의 비즈니스 시스템에도 삽입할 수 있으며, 기존의 패키지 소프트웨어나 자체 개발(Custom-Developed)시스템뿐만 아니라 다른 웹 서비스와의 커뮤니케이션이 가능하다.

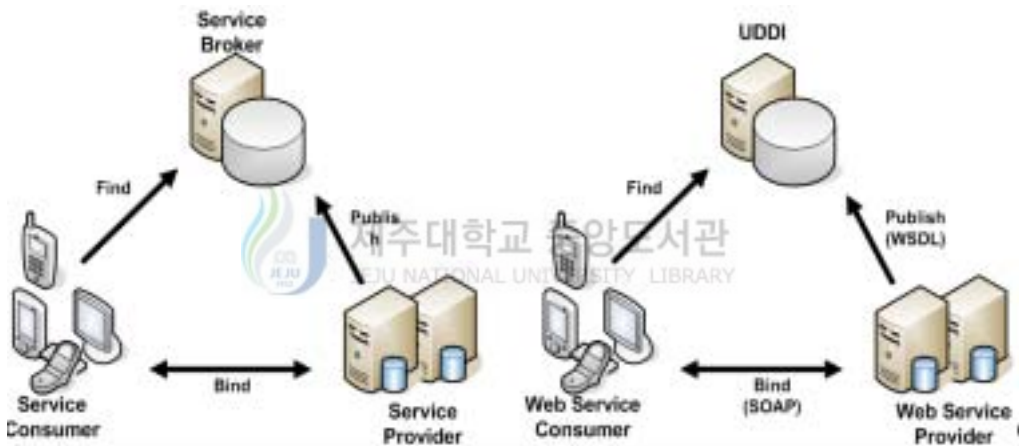
2) 웹 서비스와 서비스 지향 아키텍처

웹 서비스는 기본적으로 서비스 지향 아키텍처(Service Oriented Architectrue) [2]를 갖는다. 서비스 지향 구조란 서비스들의 모임과 이들의 연결성을 강조하는 추상적 개념이다. 서비스 지향 구조는 추상적으로 서비스들의 연결 구조를 서비스의 기술과 서비스 광고 및 발견, 서비스간의 커뮤니케이션 관점에서 제시하는 것으로 이 개념은 오랫동안 연구되어 왔다. 웹 서비스는

이러한 서비스 지향 구조의 개념을 현실적으로 구현한 것이다.

서비스 지향 아키텍처의 관점에서 볼 때, 웹 서비스가 서비스 지향 아키텍처를 실현시킨 유일한 기술은 아니다. 이미 언급된 바 있는 RPC, CORBA, DCOM, RMI등의 RPC 기반 미들웨어시스템들도 서비스 지향 아키텍처를 구현한 기술들이었다.

그러나 이들 기술은 이미 지적한 바 있는 문제점들로 인하여 성공을 거두지 못하였으며, 이제 이러한 이전 기술들의 한계를 극복한 웹 서비스 기술은 이들의 명맥을 이으면서 서비스 지향 아키텍처를 실현시킬 현실적인 대안으로 제시되었다.



[그림 5] (a) 서비스 지향 아키텍처

(b) 웹 서비스

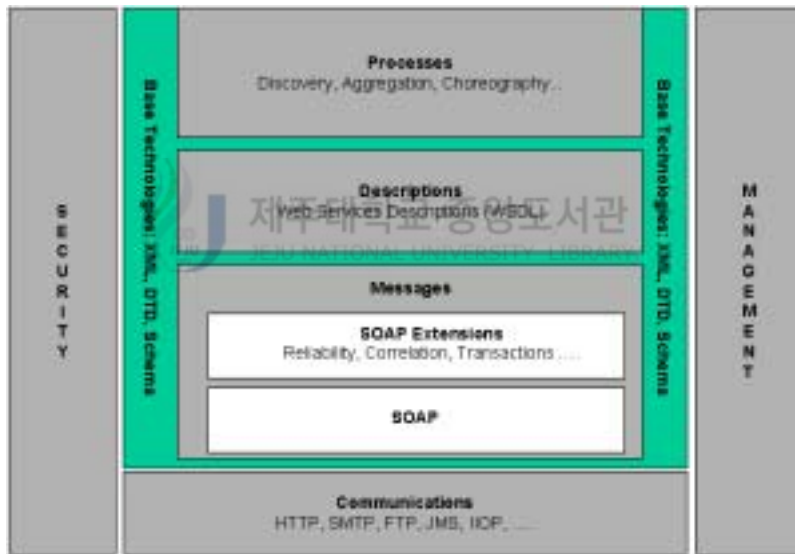
[그림 5]에서와 같이 서비스 지향 아키텍처에서 기본적인 구성요소는 서비스 제공자, 서비스 소비자, 서비스 중개자이며, 기본적인 오퍼레이션은 서비스 등록(Register/Publish), 검색(Find), 연결(Bind)이다. 서비스 제공자는 서비스를 운영하면서 이를 제공하고, 서비스 소비자는 그 서비스를 이용하며, 서비스 중개자는 제공자와 소비자 사이에서 제공자가 서비스를 광고하고 이를 소비자가 검색할 수 있도록 편의를 제공한다.

웹 서비스에서는 서비스 지향 아키텍처의 각 구성요소들과 오퍼레이션들을 구현하기 위하여 WSDL, UDDI, SOAP의 세 가지 핵심기술을 활용한다.

WSDL은 서비스 제공자가 자신의 서비스를 기술할 수 있도록 하며, UDDI는 서비스 중개자가 중개 업무를 하는 메커니즘을 제공한다. SOAP은 서비스 제공자와 소비자가 직접 커뮤니케이션 할 수 있는 방법을 제공한다.

3) 웹 서비스 기술 표준 스택(Web Service Stack)

웹 서비스 기술 표준은 XML, WSDL, SOAP, UDDI 등의 기본적인 표준들은 안정적 활용단계로 접어들었나, 그 이외에 상위 기술표준들(협업 등)과 웹 서비스 관리 및 보안 같은 표준들은 현재 표준화가 함께 진행 중에 있다. 웹 서비스의 대표적인 국제표준화 기구인 W3C는 웹 서비스 기술 표준들을 스택구조로 아래와 같이 표현하고 있다.



[그림 6] web services Architecture의 스택구조

① 커뮤니케이션 계층 (Communications Layer)

커뮤니케이션 계층에서는 네트워크 프로토콜로서 HTTP, HTTPS, SMTP(Simple Mail Transfer Protocol), FTP(File Transfer Protocol) 등의 인터넷 프로토콜들을 사용할 수 있도록 정의되어 있다. 아울러 RMI/IIOP(Internet Inter-ORB Protocol)나 MQSeries도 쓸 수 있다.

② 메시지 계층 (Messages Layer)

SOAP은 XML을 사용한 메시지 프로토콜이다. 모든 SOAP 메시지는 웹 서비스 구조의 출간(Publish), 바인드(Bind), 검색(Find)기능 등을 지원한다. SOAP 메시지는 메시지 콘텐츠를 묘사하기 위한 봉인(Envelope)과 인코딩 규칙, 원격 프로시저 호출을 제공하기 위한 메커니즘으로 구성되어 있다.

③ 기술 계층 (Description Layer)

서비스 기술 계층은 웹 서비스를 호출하는 수단에 대한 상세 정보를 제공한다. WSDL이 기술 층에서 서비스 인터페이스와 실행을 정의하는데 쓰이는 기본 표준이다. 현재 W3C에서 WSDL의 표준화를 추진하고 있으며, 11월에 WSDL 2.0의 Working Draft가 발표되었다.

④ 프로세스 계층 (Process Layer)

Process 계층은 개별적인 서비스의 상위에 존재하여, 각각의 서비스들을 함께 유통하고 활용하는 방법을 제공하는 층이다. 이 층에는 서비스 유통에 필요한 UDDI와 서비스들의 협업을 위한 표준, 비즈니스 정책(Policy)나 프로파일들이 포함된다. UDDI 레지스트리는 웹 서비스 제공자와 이용자 사이에 서비스 유통을 위한 중간 브로커의 역할을 수행한다. 웹 서비스 제공자는 보유한 서비스를 UDDI에 등록함으로써, 이용자들에게 그 서비스를 광고하고 이용하도록 유도할 수 있다. 또한 협업 표준들은 각각의 서비스들이 함께 협업하여 하나의 비즈니스를 수행할 수 있도록 하는 방법을 제공한다. 이외의 정책이나 프로파일 표준들 또한 이러한 서비스들의 협업을 통합 비즈니스를 가능하게 하는 보조적인 표준들 역할을 담당한다.

⑤ 보안 계층 (Security Layer)

웹 서비스 보안 계층은 안전한 서비스 유통을 위하여 필요한 인증(Authentication), 암호화, 권한부여(Authorization) 등의 표준을 다룬다. 현재 웹 서비스 보안은 W3C에서 XML 서명(XML Signature), 암호화(Encryption), 키관리(Key Management) 등에 대한 표준화가 진행중이고, OASIS(Organization for the Advancement of Structured)에서는 SAML(Security Assertion Markup Language), XACML(Extensible

Access Control Markup Language) 등의 표준화가 진행 중에 있다.

⑥ 관리 계층 (Management Layer)

웹 서비스가 개발되고 실제 운영될 때, 그 안정성과 신뢰성을 확보하기 위해서는 웹 서비스에 대한 관리가 중요하다. 관리 층에서는 이러한 웹 서비스의 관리에 대한 표준안을 제공한다. OASIS의 Web Services Distributed Management TC에서는 분산된 서비스들을 관리하기 위한 아키텍처와 기술을 표준화하고 있다.

3) 웹 서비스 기술의 장점

○ XML기반의 인터페이스 정의

웹 서비스 목적 중의 하나는 다른 언어로 개발된 애플리케이션들과 프로그래밍 모델들이 상호 운용되도록 하는 것이기 때문에, 웹 서비스는 XML 기반의 인터페이스 정의 언어 WSDL을 사용하고 있다. 그리고, 이것은 기본 분산 컴퓨팅 프레임워크에 작성되며, 통합 레이어로 웹 서비스를 사용하는 애플리케이션들이 단방향이나 양방향의 브리지 기술을 사용하지 않고 상호 운용되게 한다.



○ XML 기반의 프로토콜

웹 서비스는 XML기반의 표준 RPC 또는 XML프로토콜 이라고 불리면서 표준으로 발전하고 있는 SOAP을 사용해서 통신을 한다. SOAP/XMLP(XML Protocol)는 다른 분산 컴퓨팅 프로토콜에 비해서 두 가지 이점을 갖고 있다. CORBA의 IIOP, J2EE의 RMI-over-IIOP, 그리고 DCOM/COM+ 프로토콜과 비교시 우선 동적인 포트 선택 메커니즘을 사용하지 않고, 쉽게 표준 방화벽을 통해 실행될 수 있다 그리고 XML기반의 프로토콜로서, 구현되는 기본 시스템 유형, 프로그래밍 언어, 그리고 분산 컴퓨팅 프레임워크 등과는 상관없이 애플리케이션간의 인수를 마셜링하기 위해서 사용된다.

○ By-How 네이밍 서비스

웹 서비스 검색과 호출을 위해, 웹 서비스는 CORBA의 COSNaming이나 J2EE의 JNDI(Java Naming and Directory Interface) 서비스 등이 제공하는 네임 기능의 단순 바인드보다 훨씬 더 정교한 바인딩 기능을 제공하고 있다. 특히, UDDI는 표준 검색 메커니즘을 통해 발견한 후에, 호출할

수 있는 WSDL정의를 등록하기 위해서 계층적인 XML-네임스페이스의 개념을 사용하고 있다. 이것은 “by-name” 이 아니라 “by-how” 호출이라고 불린다.

표 1.에서는 분산객체기술들(CORBA, DCOM, 웹 서비스) 사이의 차이점을 보여주고 있다.

표 1. CORBA, DCOM, 웹 서비스의 비교

구 분	CORBA	DCOM	웹 서비스
전송계층	Internet Inter ORB Protocol	DCE-RPC	HTTP, SMTP 등
인터페이스	IDL	IDL	WSDL
데이터 인코딩	CDR (Common Data Representation)	NDR (Network Data Representation)	SOAP, XML
서비스 발견	명명, 레지스트리 서비스 (Naming, Registry Service)	레지스트리 (Registry)	UDDI
시스템 호환성	부분적	윈도우 계열	이기종 호환성 높음
Preferred Language	Java, C++	C++, Visual Basic	C#, Java, C++, Visual Basic 등
접속형태	접속 비지향	접속지향	접속지향
방화벽친화도	낮음	낮음	높음

4) 웹 서비스 기술의 현황

웹 서비스 기술은 단순한 정보를 활용하기 위한 웹 서비스에서 협업적 웹서비스로 발전하고 있다. 또한 유용한 정보를 가진 기업들은 웹 서비스를 활용하여 새로운 비즈니스 모델을 발굴해 이를 수익과 연결시켜 가고 있다. 협업적 웹 서비스는 웹 서비스를 단순 서비스하는 형태에서 진일보 하여 웹 서비스를 합성한 것을 말한다. 이러한 협업적 웹 서비스는 복잡한 워크플로우를 투명하게 구축할 수 있고 다수의 이질적인 웹 서비스와 사용자가 직접 상호작용하는 복잡한 과정을 단순화 할 수 있다. 이러한 협업적 웹 서비스에 대한 표준들은 BPEL4WS(Business Process Execution Language, MS, IBM, BEA) 와 WSCI(Web Services Choreography Interface, HP, BEA, SAP, Sun)를 들 수 있다. 기업의 측면에서 볼 때 웹 서비스를 활용하여 얻을 수 있는 장점을 다음과 같이 들 수 있다. 첫째, 웹 서비스는 기업 내부의 여러 부서 혹은 외부 기업과의 유연한 협력과 통합을 가능하게 한다. 둘째, 웹 서비스는 기업이 필요한 때에 필요한 기능만을 사용하는 것이 가능하므로 IT투자비용을 절감하여 효과적인 IT운영이 되도록 한다. 셋째, 기업의 입장에서 무엇보다도 매력적인 웹 서비스의 장점은 웹 서비스의 아키텍처가 개방형으로 되어 있고 모듈화 되어 있어서 기업이 웹 서비스를 도입하는데 위험부담이 적다는 것이다. 마지막으로 웹 서비스 사용자들의 일차적 목적은 IT비용의 절감이나 효율성 증대에 있지만 궁극적으로는 이것이 기업의 수입 증대에도 공헌 할 것이다. 웹 서비스로 인한 정보기술의 획기적인 변화는 필연적으로 기업의 비즈니스 패러다임에 적지 않은 변화를 몰고 올 것이다.

3. 기존 시스템의 현황 및 문제점

기존의 정보시스템(도정정보시스템, 재해재난정보시스템, 관광정보시스템, 취업정보시스템, 기상정보시스템, 교통정보시스템)들은 서로의 정보를 공유하지 못하고 개별적으로 운영되고 있다. 이러한 정보들은 상호 연관성을 가지고 있어 이를 통합하고자 하여도 백엔드 수준의 통합이 어렵다. 예를 들면 [그림 7]에서 보는 것과 같이 단순한 링크를 통해서 정보를 공유하고 이를 다시 제공하고 있다. 또한 이러한 시스템들을 분산객체기술을 사용하여 통합한다고 해도 6개의 정보시스템의 플랫폼이 모두 상이하여 하나의 시스템으로 통합하기 위해서는 전

체 시스템을 재구성해야만 가능하다. 이는 많은 예산과 시간이 소요되어 효율성을 오히려 반감시킬 수 있다.



[그림 7] 제주시청 웹서버

III. 시스템 분석 및 설계

1. 제안 시스템 기본 구조

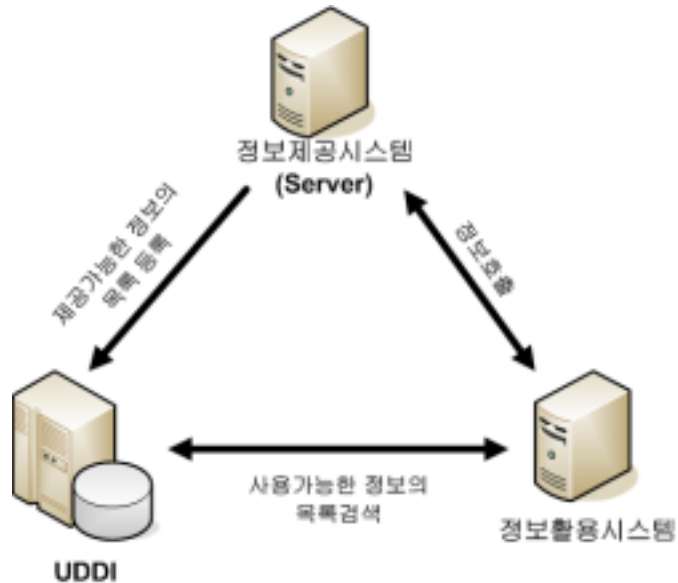
본 논문에서는 기존 정보시스템들이 정보를 제공하는 방식을 웹 서비스 기술을 적용하여 개선한다. 이는 정보시스템을 효과적이며 손쉽게 통합할 수 있도록 할 뿐만 아니라 인터넷에 접근할 수만 있으면 어떤 시스템에서도 정보를 상호 활용할 수 있어 정보의 활용이 극대화 될 수 있다. 또한 이렇게 구성된 각각의 정보시스템들을 통합하여 새로운 통합정보를 제공하는 정보통합시스템을 구현하고 통합된 새로운 정보들이 다시 다른 시스템에게 제공될 수 있도록 설계하고 구현하였다. 예를 들어, 특정 관광지까지의 도로교통상황에 대한 정보, 재해·재난 발생지의 날씨현황 등 각기 다른 시스템의 정보들이 통합되어 새로운 정보로 제공된다. 본 논문에서 제안한 웹 서비스 기술을 적용한 정보통합시스템의 전체적인 기본 구조는 [그림 8]과 같으며, 정보제공자시스템인 Server, 정보 활용시스템인 통합시스템 및 타 시스템들, 제공되는 정보 및 기능에 대한 정보를 저장하는 UDDI로 구성될 수 있다.

서버 측의 기본적인 역할은 각 서버의 정보를 타 시스템에 제공하는 것이다. 이를 위해 서버는 웹 서비스 공개와 함께 UDDI 저장소에 서버에서 제공되는 각종 기능에 대해 WSDL 문서를 등록하여 정보를 활용하고자 하는 타 시스템에서 이를 활용할 수 있도록 한다.

UDDI 저장소는 서버 측에서 제공하는 정보에 대한 WSDL 문서를 저장하고 정보 활용자의 요청에 의해 제공하는 역할을 한다. 제공된 WSDL 문서는 정보활용 시스템에서 각종 기능 및 정보를 활용하기위해 사용된다.

정보활용시스템은 UDDI 저장소로부터 정보제공시스템의 WSDL 문서를 이용하여 각종 기능 호출에 필요한 정보와 함께 서버 측에서 공개한 웹 서비스를 호출하여 필요한 기능을 사용 할 수가 있다.

이와 같이 웹 서비스 기술을 이용하면 인터넷에 접근 가능한 어떠한 기기(PC, Notebook Computer, PDA, Mobile Phone 등)에서도 필요한 정보를 웹 서비



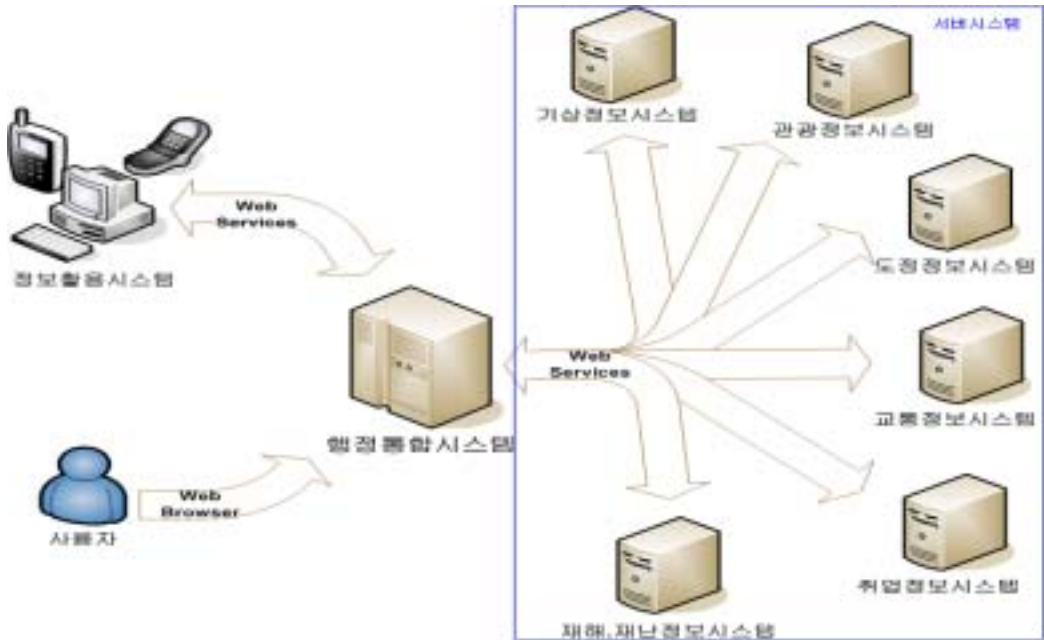
[그림 8] 제안시스템의 기본적인 구조

스가 적용된 어플리케이션으로 개발하여 정보제공시스템으로부터 정보서비스를 받을 수 있게 되는 것이다. 또한 어떠한 프로토콜로도 호출이 가능하기 때문에 방화벽 내부에 있는 서버에 대해서도 아무런 제약 없이 접근이 가능하게 된다.

2 제안 시스템 설계

1. 제안 시스템의 기본 구조는 기본적인 웹 서비스 모델의 활용을 보여준다. 이는 정보시스템이 단일 시스템인 경우 그 시스템에 웹 서비스 연계(Web Services interface) 모듈만 작성하고 정보를 이용하는 방법이다. 이러한 기본적인 구조를 바탕으로 각각의 정보 시스템을 통합하여 정보통합시스템을 구현하고 통합된 새로운 정보를 다시 다른 정보활용시스템에 제공할 수 있도록 구현하였다. [그림 9]는 제안시스템의 구조도를 보여주고 있다.

제안 시스템은 크게 정보를 제공하는 서버시스템에 웹 서비스로 구현된 Server Application, 정보시스템들 사이의 각종 정보를 통합하여 정보를 제공하는 정보통합시스템, 통합된 정보를 웹 서비스로 다시 활용하는 정보활용시스

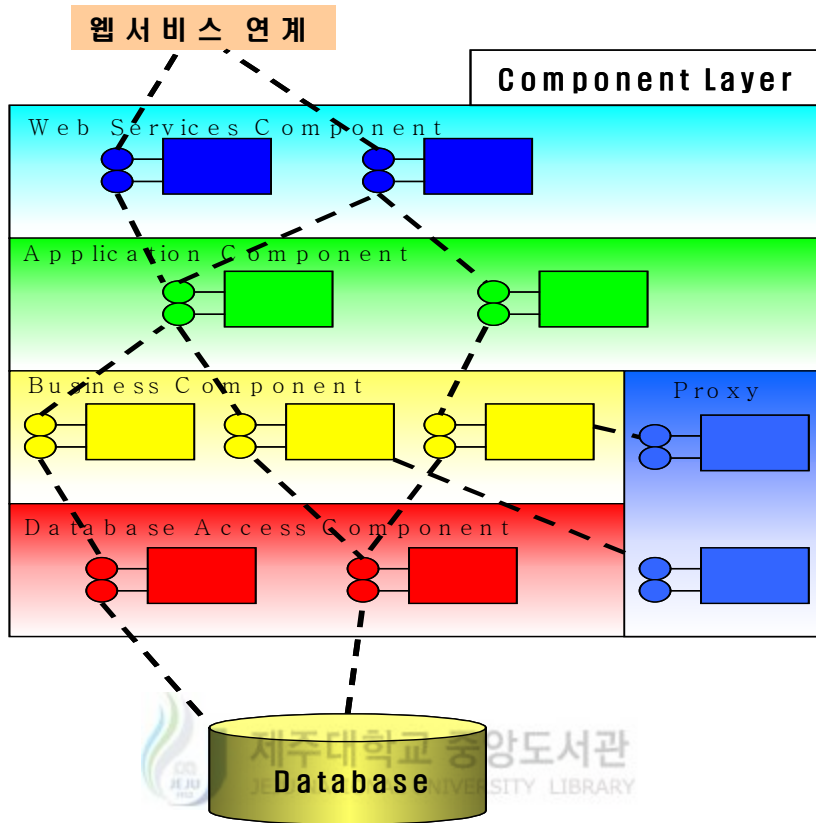


[그림 9] 제안시스템의 구조도

템의 Client Application 그리고 정보통합시스템을 이용하는 사용자 부분으로 나누어 설명할 수 있다.

1) Server Application

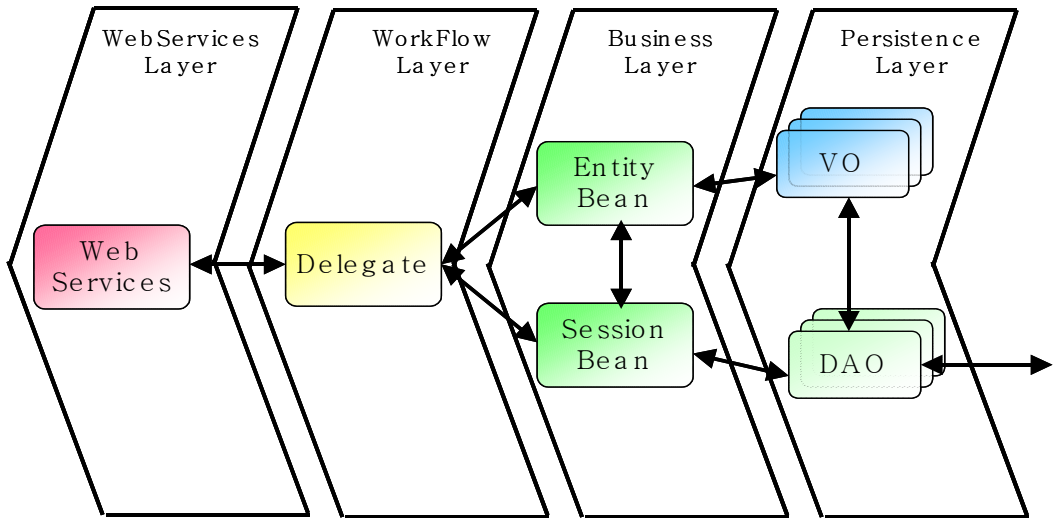
각 정보시스템의 Server Application은 [그림 10]과 같은 컴포넌트 아키텍처 스타일을 가지고 있다. 이러한 구성은 각각의 정보시스템에 웹 서비스로 구현 되게 된다.



[그림 10] 컴포넌트 아키텍처 스타일

각각의 컴포넌트 계층은 데이터베이스를 직접 접근하여 처리하는 데이터베이스 접근 컴포넌트(Database Access Component), 로직을 담고 있는 비즈니스 컴포넌트(Business Component), 웹 서비스 컴포넌트(Web Services Component)와 연계되는 어플리케이션 컴포넌트(Application Component), 그리고 클라이언트에서의 각종 정보요청에 대한 인터페이스를 담당하는 웹 서비스 컴포넌트(Web Services Component)로 구성되어 있다. 또한 이 컴포넌트 스타일은 [그림 11]과 같은 4계층의 소프트웨어 아키텍처 계층을 사용하여 설계하였다.

- 웹 서비스 계층 (Web Services Layer): 어플리케이션 사용자의 액션

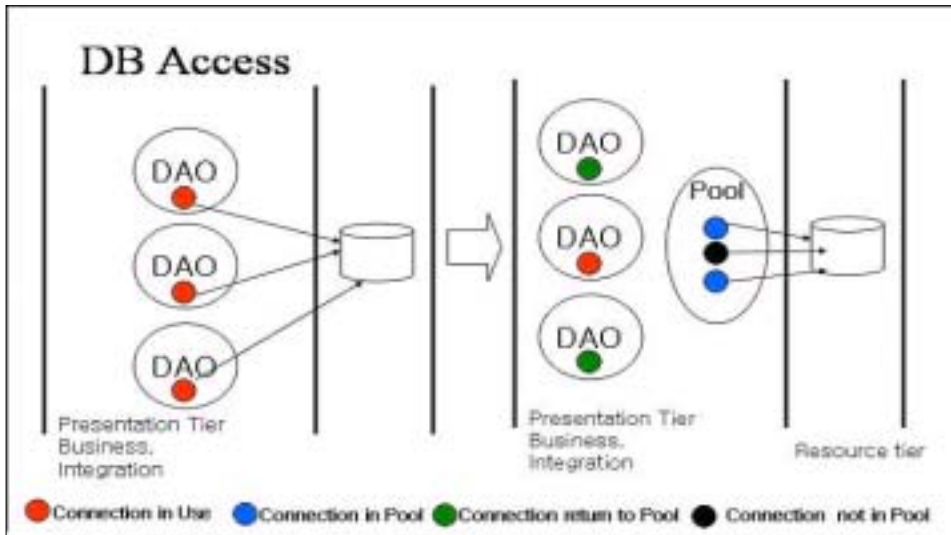


[그림 11] 소프트웨어 아키텍처 스타일

처리가 이루어지는 계층이다. 이 Layer에서는 웹 서비스 기술을 이용해 구현되며, 사용자의 요구에 따라 서비스를 요청하는 계층

- 워크 플로우 계층 (Work Flow Layer): 서비스 요청 메시지를 전달하는 책임을 수행하는 계층
- 비즈니스 계층(Business Layer): 비즈니스 로직을 가지고 있으며 요청 메시지에 따라 해당 기능을 수행하는 계층
- 퍼시스턴스 계층(Persistence Layer): 해당 서비스의 정보를 지속적으로 유지시켜주는 역할을 수행하는 계층

Server Application의 데이터 접근방식은 직접 데이터베이스에 접근하는 방식을 연결 풀(Connection Pool)을 이용하여 [그림 12]과 같이 설계하고 데이터베이스의 연결성능을 향상시켰다. 그리고 정보시스템의 접근 보안에 대한 설계는 HTTP Basic 인증과 SSL을 적용하여 보안설계를 하였다.



[그림 12] 데이터베이스 접근방식

* Server Application의 실제 구축된 함수내역

(1) 교통정보시스템 제공 함수

함수명	설명
getBuildingCodeList	첨단시설물의 종류 코드 목록 조회
getCategory	교통지역의 대분류 (제주시, 서귀포시, 서부산업도로) 조회
getLinkRoadList	대분류에 속하는 중분류 도로명 조회
getRoadList	상세도로에 대한 정보 조회
getTrafficList	소통정보 리스트 정보 조회
getOccurList	돌발발생 리스트 정보 조회
getControlPlanList	공사통제 리스트 정보 조회
getBuildingList	첨단시설물의 설치 위치정보 조회
getCCTVList	CCTV의 정지영상 위치정보 조회
getRoadWeatherList	도로기상정보 조회
getTrafficOccurList	소통돌발정보 리스트 조회

(2) 관광정보시스템 제공 함수

함수명	설명
1. 코드정보 함수	
SearchCodeGroupCode	시스템 코드 호출 함수
SearchDataKindCode	자료 종류 시스템 코드 호출 함수
SearchCategoryCode	시스템 분류 코드 검색
ViewCategoryCode	시스템 분류 상세보기 함수

SearchOperationCode	검색방식 시스템 코드 호출 함수
SearchZoneCode	지역코드 검색 호출 함수
ViewZoneCode	지역코드 상세보기 위해 호출 함수
2. 관광정보통합 검색	
SearchComplexData	관광정보 복합 조회 함수
SearchData	관광정보 단순 조회 함수
ViewData	자료를 상세보기 위해 호출하는 함수
SearchDataTop	관광최신 정보 조회 함수
SearchTotal	관광정보 중 관광지, 숙박지, 음식점, 교통에서만 조회 함수
GetPointTour	네티즌추천으로만 관광정보 조회함수
3. 게시판 관련 함수	
GetBoardList	관광지 소식 게시판의 게시물 함수
GetBoardView	관광지 소식 게시물 상세보기 함수
4. 문화/축제 정보	
SearchFestival	문화/축제정보 검색 함수
ViewFestival	문화/축제정보 상세보기 함수
5. 관광지 정보	
SearchTour	관광지 정보 검색 함수
ViewTour	관광지 정보 상세 보기 함수
6. 숙박지 정보	
SearchLodging	숙박지 정보 검색 함수
ViewLodging	숙박지 정보 상세보기 함수
7. 레저 정보	
SearchLeisure	레저정보 검색 함수
ViewLeisure	레저정보 상세보기 함수
8. 음식 정보	
SearchFood	음식점 정보 검색을 위해 호출 함수
ViewFood	음식점 정보를 상세보기 위해 호출 함수
9. 교통 정보	
SearchTransportCompany	교통업체(택시, 렌터카, 버스)를 검색하기 위해 호출 함수
SearchTransportEquipment	교통시설(공항, 항공, 터미널)을 검색하기 위해 호출 함수
SearchTransportFacilities	교통편의시설(여행사, 카센터, 주유소등)을 검색을 위해 호출 함수
ViewTransportCompany	교통업체(택시, 렌터카, 버스)를 상세보기 위해 호출 함수
ViewTransportEquipment	교통 편의 시설(공항, 항공, 터미널)을 상세보기 위해 호출 함수
ViewTransportFacilities	교통 편의 시설(여행사, 카센터, 주유소등)을 상세보기 위해 호출 함수
10. 버스 정보	

SearchBusStop	버스정류장관련 자료 검색 함수
ViewBusPath	버스노선 관련 자료를 출력한다
11. 예약 상품정보	
SearchGoodsByComplex	예약 상품을 복합적으로 검색하기 위해 호출하는 함수
SearchGoodsByPrice	예약 상품을 가격별로 검색하기 위해 호출하는 함수
SearchGoodsByGoodsCode	예약 상품 코드별 검색 함수
SearchGoodsCode	예약 상품 검색 함수
ViewGoods	예약상품 상세보기 위해 호출하는 함수
SearchGoodsCodeGroupCode	
RegistFreeReservation	자유예약 등록 함수
12. 자료평가 정보	
SearchDataRation	자료 평가 검색을 위해 호출하는 함수
ViewDataRation	자료를 상세보기 위해 호출하는 함수
RegistDataRation	자료 평가 등록을 위해 호출하는 함수
DeleteDataRation	
13. 주인장 한마디 정보	
SearchOwnerNotify	주인장 한마디 검색 함수
ViewOwnerNotify	주인장 한마디 상세보기 함수
14. 법정동 주소 정보	
SearchGroundCode	법정동 검색 함수
ViewGroundCode	법정동 코드 상세 보기 함수

(3) 도정정보시스템 제공 함수

서비스명	설명
1. 게시판정보 서비스	
listLoginUserItem	신고센터 로그인 사용자 작성 게시물 목록 조회 함수
listNotLoginUserItem	신고센터 비로그인 사용자 작성 게시물 목록 조회 함수
listHotItemByDate	게시판의 최신 게시 목록 조회 함수
getSearchList	통합 게시판 검색 목록 조회 함수
listItem	게시판의 게시물 목록 조회 함수
showItem	게시판의 게시물 상세정보 조회 함수
showItems	답변글이 있는 게시물에 대해 답변글 조회 함수
showEncItem	게시판의 게시물 상세정보 조회 함수(게시물 정보 중 비밀번호 암호화)
addItem	게시물 등록 함수

addEncItem	게시물 등록 함수(게시물 정보 중 비밀번호 암호화)
modifyItem	게시물 수정 함수
removeItem	게시물 삭제 함수
2. 포럼정보 서비스	
getCurrentForumList	현재 진행 중인 토론 목록 조회 함수
getOldForumList	지난 토론 목록 조회 함수
getForumContentList	토론 내용 목록 조회 함수
showForum	토론 상세 내용 조회 함수
showForumContent	등록된 토론 내용 조회 함수
addForumContent	토론 내용 등록 함수
modifyForumContent	토론 내용 수정 함수
removeForumContent	토론 내용 삭제 함수
3. 페이지정보 서비스	
getPage	게시되어 있는 페이지 목록 조회 함수
getPageList	페이지 상세 정보 조회 함수
4. 설문조사정보 서비스	
getSurveyList	설문조사 목록 조회 함수
getSurvey	설문조사 상세정보 조회 함수
voteSurvey	설문결과 목록 조회 함수
getSurveyResult	설문 응답 함수
5. 회원정보 서비스	
checkDupId	사용자 UserID 중복 확인 함수
checkAuth	사용자 로그인 함수
showUser	사용자 정보 조회 함수
showPostInfos	우편번호 목록 조회 함수
addUser	사용자 정보 등록 함수
modifyUser	사용자 정보 수정 함수
modifyUserPasswd	사용자의 비밀번호 수정 함수
removeUser	사용자 정보 삭제 함수
6. 첨부파일 서비스	
sendFileToServer	첨부파일을 클라이언트로부터 서버로 업로드 하는 함수
receiveFileFromServer	첨부파일을 서버로 부터 클라이언트로 다운로드하는 함수

(4) 재해정보시스템 제공 함수

함수명	설명
1. 강우량 정보	제주도내에 위치한 자동강우량 관측소로 부터의 강우량 정보
GetRtuNameList	강우량 관측소명 리스트 보기
GetRealRain	시간강우량 보기
GetDayRain	일간강우량 보기
GetSeasonRain	기간강우량 보기
2. 게시물 정보	재해대책본부에서 발표하는 대국민보고서, 공지사항에 대한 게시물 정보
GetAccHotList	최근소식리스트 보기
GetAccBoardList	게시판 리스트 보기
GetAccBoardView	게시물 상세보기
3. 재해행동요령 정보	재해시 행동요령에 대한 정보
GetActionPointCodeList	재해행동요령 코드 보기
GetActionPointData	재해행동요령 상세 보기
4. 재해대책기관 정보	재해대책기관 목록 및 상세정보
GetAccOrganizaionList	재해대책본부 목록 보기
GetAccOrganizaionDetailList	재해대책본부 상세 보기
5. 재해영상 정보	재해 예방을 위한 웹카메라 목록 및 상세정보
GetCameraList	웹카메라 코드 목록보기
GetCameraDetailList	웹카메라 상세 보기

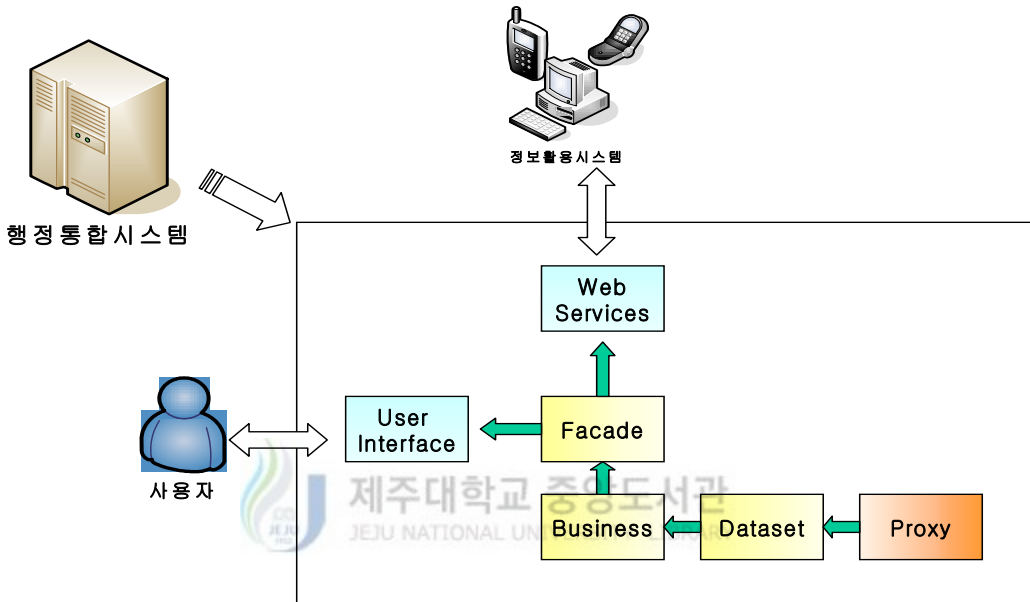
(6) 취업정보시스템 제공 함수

서비스명	설명
1. 회원정보 서비스	
1.1 개인회원 정보	
showMemberLogin	회원의 로그인 정보 확인
checkDupMemberID	회원 아이디 중복 여부 확인
checkDupMemberNum	회원 주민번호 중복 여부 확인
addUser	회원정보 등록
removeUser	회원정보 삭제
showUser	회원정보 조회
modifyUser	회원정보 수정
showMemberResume	회원 이력서 등록 여부 확인
showIDThroghNum	주민등록번호를 이용해서 회원ID 조회
showMemberPass	회원 비밀번호 조회
showMemberID	이름과 주민등록번호를 이용해서 회원ID 조회
1.2 기업회원 정보	
removeCompany	기업회원 정보 삭제
showCompanyPass	이름과 회사번호를 이용해서 기업회원 비밀번호를 조

	회
showCompanyID	이름과 회사번호를 이용해서 기업회원ID를 조회
modifyCompany	기업회원 정보 수정
showCompanyOffer	기업회원의 구인정보 등록 확인
showCompanyLogin	회원의 로그인 정보 확인
showCompany	기업회원 정보를 조회
checkDupCompanyNum	기업회원 회사등록번호 중복 확인
addCompany	기업회원 정보 등록
checkDupCompanyID	기업회원 ID 중복 확인
2. 게시판 서비스	
2.1 구인 정보	
showOfferList	구인 게시판 게시물목록 조회
showOffer	구인 게시판 게시물 조회
addOffer	구인 게시판 구인정보 등록
modifyOffer	구인 게시판 구인정보 수정
removeOffer	구인 게시판 구인정보 삭제
2.2 구직 정보	
showSeekerList	구인 게시판의 게시물 목록
showSeeker	구직 게시판의 게시물 조회
addSeeker	구직 게시판의 게시물 등록
modifySeeker	구직 게시판의 게시물 수정
removeSeeker	구직 게시판의 구직정보 삭제
2.3 설문 조사	
showPollList	설문조사 게시판의 게시물목록 조회
showPoll	설문조사 게시판의 설문내용, 설문답변, 결과 조회
addPollAns	설문에 응한 결과 삽입 함수
2.4 취업 달력	
showCalendar	취업캘린더 게시판의 년도, 월별 목록 조회
2.5 공통 코드	
showCodeInfoList	공통코드 목록 조회 ([별첨] 2. 취업정보 공통 코드)
showCodeInfo	공통코드 상세정보 보기 ([별첨] 2. 취업정보 공통 코드)
showCommonCodeList	공통코드 상세 목록 조회([별첨] 2. 취업정보 공통 코드)
showCommonCode	공통코드 상세정보 보기 ([별첨] 2. 취업정보 공통 코드)
showOccupationList	직종 코드 목록 조회([별첨] 2. 취업정보 공통 코드)
showOccupation	직종 코드 상세정보 조회([별첨] 2. 취업정보 공통 코드)
3. 첨부파일 서비스	
receiveFileFromServer	서버에서 파일 수신
sendFileToServer	서버로 파일 전송

2) 정보통합시스템

정보통합시스템은 여러 정보시스템에서 제공하는 정보를 통합하여 새로운 정보를 사용자에게 서비스하고, 통합된 정보를 다른 정보활용시스템에 서비스 할 수 있도록 Server Application도 포함되어 설계되었다. [그림 13]은 정보통합시스템의 S/W Architecture를 보여주고 있다.



[그림 13] 정보통합시스템 구성도

○ Proxy

Server Application의 웹 서비스 계층(web services Layer)과 통신을 담당 부분으로 웹 서비스의 SOAP을 바탕으로 통신을 한다.

○ Dataset

서버시스템으로부터 수신된 XML 자료를 언마샬링(UnMarshaling) 하여 데이터셋(Dataset)으로 매칭시켜 주는 역할을 담당한다.

○ Business

수신된 정보를 필요에 따라 가공하거나 타 시스템의 정보와 연계하여 새로운 정보를 만드는 역할을 담당한다.

○ Facade

생성된 정보를 사용자 인터페이스에 넘겨주거나 타 시스템에게 제공될 수 있도록 정보의 미들웨어 역할을 수행한다.

○ User Interface

각각의 정보시스템에서 수신된 정보를 제공하기 위해 구성된 화면

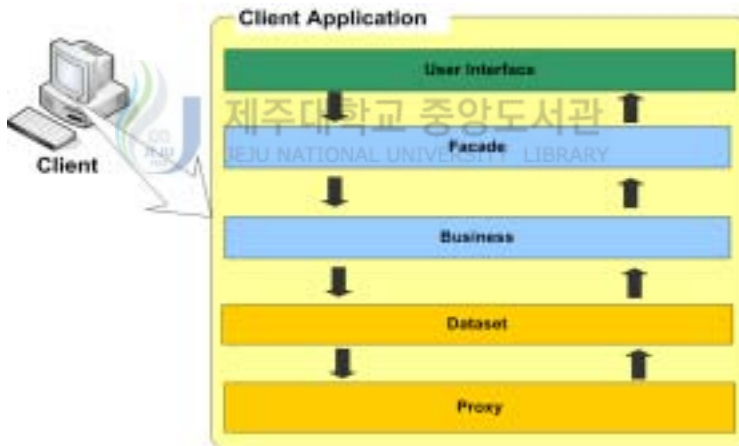
○ web services

각각의 정보시스템에서 수신된 정보를 재가공 및 연계를 통한 새로운 정보를 다른 정보활용시스템에 제공할 수 있도록 하는 역할을 수행한다.

3) Client Application

Client Application은 정보통합 시스템에서 제공하는 웹 서비스를 이용하여 개발할 수 있다. 즉 인터넷에 연결 가능한 단말기에 작동하는 개발언어를 이용하여 웹 서비스 Application을 개발하여 통합정보를 활용할 수 있다. Client Application에서는 필요한 UI에 대한 구현과 필요로 하는 정보의 목록을 UDDI를 통해 얻고 해당 정보제공시스템으로부터 필요한 정보를 가져와서 이용하면 된다. 본 논문에서 UDDI는 정보통합시스템에 구축되어 있다.

[그림 14]에서는 Client Application의 기본 구조를 나타낸것이다



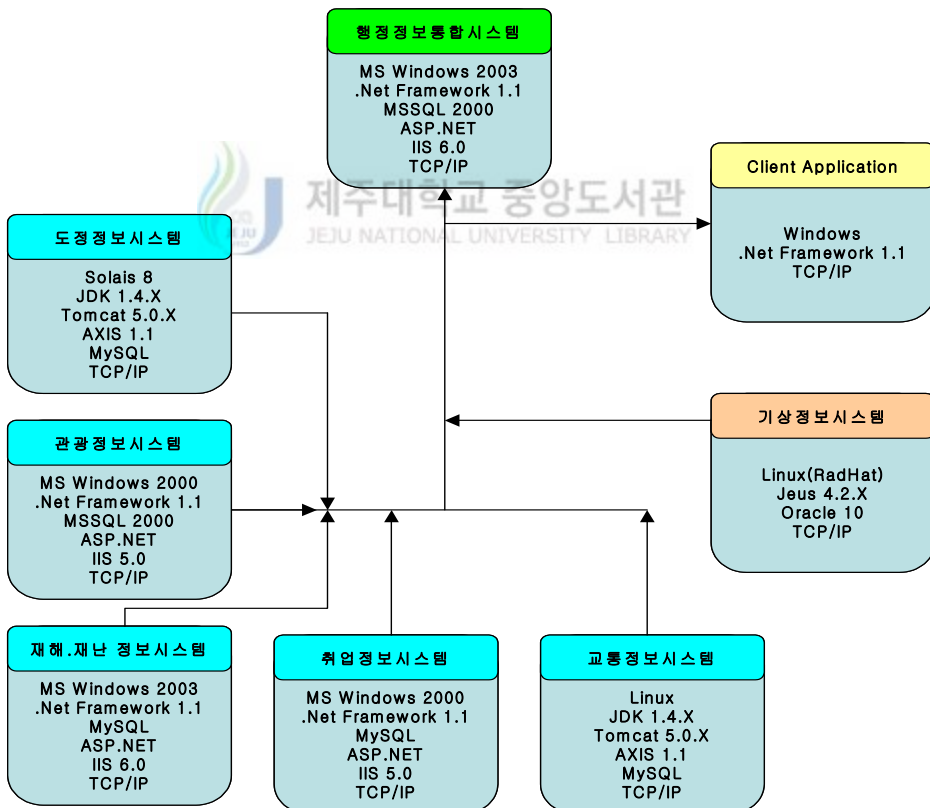
[그림 14] Client Application의 구조

[그림 14] Client Application 구조도는 정보통합 시스템의 구조도와 유사한 형태를 갖추고 있다. 물론 Client 시스템 또한 각종 정보를 또 다른 시스템에 제공할 수 있으며 이때 Client 시스템에 웹 서비스를 제공하는 모듈만 추가하면 된다.

IV. 시스템 구현 결과

1. 제안 시스템 개발 환경

본 논문에서 제안한 정보통합시스템은 도정정보시스템, 관광정보시스템, 재해·재난정보시스템, 취업정보시스템, 교통정보시스템의 정보를 얻기 위해 Server Application을 개발하고 Server Application이 기개발 되어 있는 기상정보시스템등 전체 6개 정보시스템의 정보를 통합하여 정보통합시스템을 구현하였다. 또한 구현된 통합정보를 Client Application으로 제공하여 정보통합시스템의 활용까지 구현해 보았다. 전체 S/W 아키텍처는 [그림 15] 과 같다.



[그림 15] 전체 시스템 S/W 아키텍처

2. 제안 시스템 구현

제안 시스템의 구현을 위해 본 논문에서는 크게 Server Application, 정보통합시스템, Client Application으로 구분하여 수행하였다.

1) Server Application 개발

Server Application은 표 2- 표6에서 보이는 것과 같이 각 정보시스템별로 웹 서비스 모듈을 구성하고 있다. 각 모듈들은 각각의 정보시스템 내에 구현되어 웹 서버에서 서비스되고 있다.

표 2. 교통정보 서버의 웹 서비스 모듈(Server Application)

서비스명	웹 경로	설명
관광정보 서비스	http://211.184.197.174:8080/its/services/TrafficService?WSDL	교통정보시스템에서 제공하는 각종 서비스들을 웹 서비스로 제공

표 3. 재해.재난정보 서버의 웹 서비스 모듈(Server Application)

서비스명	웹 경로	설명
관광정보 서비스	http://bangjae.jeju119.go.kr/accweb/service/accfacade.asmx	재해.재난정보시스템에서 제공하는 각종 서비스들을 웹 서비스로 제공

표 4. 관광정보 서버의 웹 서비스 모듈(Server Application)

서비스명	웹 경로	설명
관광정보 서비스	http://cyber.jeju.go.kr/tourweb/service/tur.asmx?wsdl	관광정보시스템에서 제공하는 각종 서비스들을 웹 서비스로 제공

표 5. 취업정보 서버의 웹 서비스 모듈(Server Application)

서비스명	웹 경로	설명
취업정보 회원관련 서비스	http://job.ejeju.net/wsJobMemService/MemberFD.aspx?wsdl	취업정보서버에서 사용되는 개인회원 및 기업회원에 관련된 비즈니스 로직을 웹 서비스로 구현
게시판 정보관련 서비스	http://job.ejeju.net/wsJobService/BoardFD.aspx?wsdl	취업정보서버에서 사용되는 각종 게시판(채용소식, 이력서작성등)의 비즈니스 로직을 웹 서비스로 구현
첨부파일 관련 서비스	http://job.ejeju.net/wsJobFileService/AttachFileFD.aspx?wsdl	이력서 작성 시 개인의 사진을 첨부하기 위한 웹 서비스(Microsoft DIME 활용)

표 6. 도정정보 서버의 웹 서비스 모듈(Server Application)

서비스명	웹 경로	설명
게시판 정보 서비스	http://www.jeju.go.kr:8080/ctz/services/BoardService?wsdl	도정정보서버에서 사용되는 게시판 관련 웹 서비스
포럼 정보 서비스	http://www.jeju.go.kr:8080/ctz/services/ForumService?wsdl	도정정보서버에서 사용되는 정책마당등의 포럼 관련 웹 서비스
페이지 정보 서비스	http://www.jeju.go.kr:8080/ctz/services/PageService?wsdl	도정정보서버에서 제공되는 게시판, 메뉴 및 페이지에 대한 목록 조회 웹 서비스
설문조사 정보 서비스	http://www.jeju.go.kr:8080/ctz/services/SurveyService?wsdl	도정정보서버에서 제공하는 설문조사관련 웹 서비스
회원	http://www.jeju.go.kr:8080/ctz/se	도정정보서버 회원관련

정보 서비스	rvices/MemberService?wsdl	웹 서비스
첨부파일 서비스	http://www.jeju.go.kr:8080/ctz/services/AttachService?wsdl	도정정보서버 게시판 첨부파일 관련 웹 서비스

각각의 행정정보시스템에는 표 7.에서 보이는 것과 같이 Server Application 구현되었다. 표 7.은 정보시스템중 도정정보시스템에 구현된 소스의 구축현황을 보여주고 있다.

표 7. 도정정보시스템의 Server Application의 구축현황.

URL	www.jeju.go.kr		설치디렉토리	/data3/web_service/ctz		
구성정보	- AttachService, MemberService, BoardService, BoardService, PageService를 Deploy시킨다.					
	- http://www.jeju.go.kr:8080/ctz/services/AttachService?WSDL					
	- http://www.jeju.go.kr:8080/ctz/services/MemberService?WSDL					
	- http://www.jeju.go.kr:8080/ctz/services/BoardService?WSDL					
	- http://www.jeju.go.kr:8080/ctz/services/PageService?WSDL					
provin/wsitp	ctz	attach	첨부파일을 관리하는 디렉토리			
		bat	deploy.bat	시스템에 웹 서비스를 Deploy시키는 배치 구성 파일		
			deploy.wsdd	Deploy될 내용을 담고 있는 구성파일		
			server-db.xml	tomcat의 server.xml에 DB부분을 추가해야되는 내용		
			server-ssl.xml	tomcat의 server.xml에 ssl부분을 추가해야되는 내용		
			tomcat-users.xml	베이직 인증을 위해 tomcat.xml에 사용자를 등록하는 내용		
		logs	로그 기록 디렉토리			
		secret	SecretKey.ser	SSL을 위한 보안키		
		ssl	client.keystore	클라이언트 보안키		
			exportkey.bat	키를 내보내는 배치 구성파일		
			genkey.bat	키생성 배치 구성파일		
			importkey.bat	키를 가져오는 배치 구성파일		
			server.cer	서버 보안키		
			server.keystore	서버측 키보관소		
			ssl_readme.txt	SSL설정을 위한 도움말		
		WEB-INF		server-config.wsdd	서버 환경설정을 위한 구성 파일	
			classes	컴파일된 모든 내용의 바이너리 파일을 담고 있다.		
			lib	필요한 라이브러리를 가지고 있다.		
		src	provin/wsitp		AccessHandler.java	웹 서비스 접근시 처리되는 내용을 담고 있는 구성파일
					ReqHandler.java	요구(Request)시 처리되는 내용을 담고 있는 구성파일
	ResHandler.java			응답(Response)시 처리되는 내용을 담고 있는 구성 파일		

				AttachDAO.java	첨부파일 DB 구성 파일
			attachfiledo	AttachDAOException.java	첨부시 오류의 발생을 처리하는 구성파일
				AttachSQL.java	첨부파일의 SQL생성 구성파일
			attachfilebean	AttachBean.java	첨부 비즈니스 구성파일
				AttachIOException.java	첨부의 입출력시 발생하는 오류처리 구성파일
			attachfiledelegate	AttachDelegate.java	웹 서비스 접근시 처리되는 내용을 담고 있는 구성파일
			attachfilevo	AttachVO.java	첨부파일 Value Object 구성파일
			boardbean	BoardBean.java	게시판 비즈니스 구성파일
				BoardException.java	게시판 오류처리하는 구성파일
				CategoryException.java	카테고리 오류처리하는 구성파일
			boarddao	BoardDAO.java	게시판 DB구성파일
				BoardSQL.java	게시판 SQL생성 구성파일
				CategoryDAO.java	카테고리 DB구성파일
				CategorySQL.java	카테고리 SQL구성파일
			boarddelegate	BoardDelegate.java	웹 서비스 접근시 처리되는 내용을 담고 있는 구성파일
			boardvo	CategoryVO.java	카테고리 Value Object 구성파일
				ItemVO.java	게시판 아이템 Value Object 구성파일
				ListItemVO.java	게시판 리스트 아이템 Value Object 구성파일
				ListVO.java	게시판 리스트 Value Object 구성파일
				SearchResultItemVO.java	검색결과 아이템 Value Object 구성파일
				SearchResultListVO.java	검색결과 리스트 Value Object 구성파일
				common/appconf	ctz.properties
			common/utills	ChiperUtil.java	비밀번호 암호화 관련 구성파일
					ConfigUtil.java

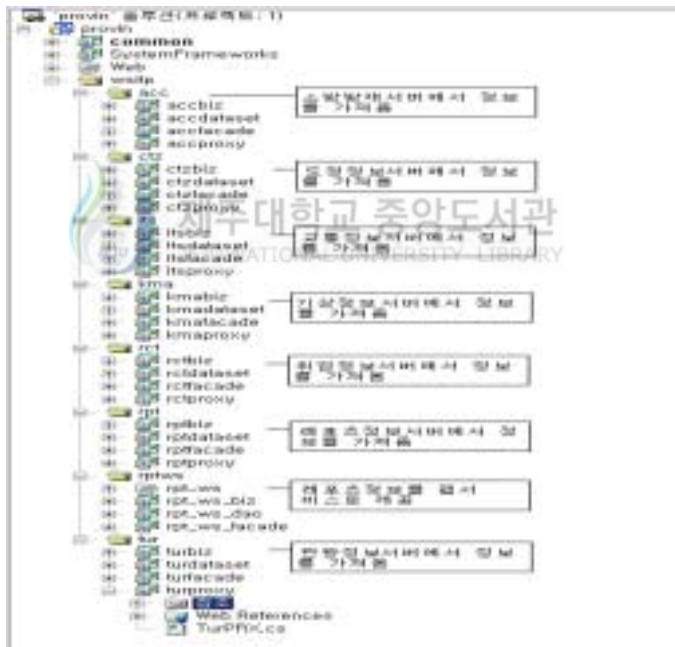
					일
				CryptUtil.java	암호화 관련 구성파일
				DBUtil.java	DB 다루는 구성파일
				EncUtil.java	Encoding 관련 구성파일
				FileUtil.java	파일관련 구성파일
				Guid.java	GUID생성과 관련된 구성파일
				Log4jInit.java	로그를 다루는 구성파일
				StringUtil.java	스트링 관련 유틸리티 구성파일
				TimeUtil.java	시간 유틸리티 구성파일
			forum	ForumException.java	포럼 오류처리 구성파일
			forum/ bean	ForumBean.java	포럼 비즈니스 구성파일
			forum/ dao	ForumDAO.java	포럼 DB구성파일
				ForumSQL.java	포럼 SQL 구성파일
			forum/ delegat e	ForumDelegate.java	웹 서비스 접근시 처리되는 내용을 담고 있는 구성파일
			forum/ vo	ForumContentSummary VO.java	포럼 내용의 요약 Value Object 구성파일
				ForumContentVO.java	포럼 내용 Value Object 구 성파일
				ForumSummaryVO.java	포럼 요약 Value Object 구 성파일
				ForumVO.java	포럼 Value Object 구성파 일
				ListForumContentVO.ja va	포럼 내용 리스트 Value Object 구성파일
				ListForumVO.java	포럼 리스트 Value Object 구성파일
			membe rbean	MemberAuthFailExcept ion.java	회원 권한 관련 오류 처리 구성파일
				MemberBean.java	회원 비즈니스 구성파일
				MemberException.java	회원 관련 오류 처리 구성파 일
				NameCheckBean.java	이름 검사 비즈니스 구성파 일
			membe rdao	MemberDAO.java	회원 DB구성파일
				MemberSQL.java	회원 SQL 구성파일
			membe	MemberDelegate.java	웹 서비스 접근시 처리되는 내용을 담고 있는 구성파일

			rdelegat		
			membe rvo	MemberVO.java	회원 Value Object 구성파 일
				PostInfoVO.java	주소 Value Object 구성파 일
			pages	PageException.java	페이지 관련 오류 처리 구성 파일
			pages/ bean	PageBean.java	페이지 비즈니스 구성파일
				PageDAO.java	페이지 DB구성파일
			pages/ dao	PageSQL.java	페이지 SQL 구성파일
			pages/ delegat e	PageDelegate.java	웹 서비스 접근시 처리되는 내용을 담고 있는 구성파일
			pages/ vo	PageListVO.java	페이지 리스트 Value Object 구성파일
				PageSummaryVO.java	페이지 요약 Value Object 구성파일
				PageVO.java	페이지 Value Object 구성 파일
			survey	SurveyException.java	설문조사 오류처리 구성파일
				SurveyDAO.java	설문조사 DB 구성파일
			survey/ dao	SurveySQL.java	설문조사 SQL구성파일
			survey/ delegat e	SurveyDelegate.java	웹 서비스 접근시 처리되는 내용을 담고 있는 구성파일
			survey/ vo	SurveyAnswerItemVO.j ava	설문응답 아이템 Value Object 구성파일
				SurveyAnswerVO.java	설문응답 Value Object 구 성파일
				SurveyItemVO.java	설문 아이템 Value Object 구성파일
				SurveyListVO.java	설문 리스트 Value Object 구성파일
				SurveyResultCountVO.j ava	설문 결과 개수 Value Object 구성파일
				SurveyResultItemVO.ja va	설문 결과 아이템 Value Object 구성파일

					SurveyResultVO.java	설문 결과 Value Object 구성파일
					SurveySummaryVO.java	설문 요약 Value Object 구성파일
					SurveyVO.java	설문 Value Object 구성파일

2) 정보통합시스템 개발

[그림 16]는 정보통합시스템의 구조를 나타내고 있다. 제안 시스템의 구현은 .Net을 이용하여 구현하였고 각각의 행정정보시스템에 있는 웹 서비스 모듈 (Server Application)과 통신하기 위해 정보통합시스템에서 proxy를 생성하여 사용하였다.



[그림 16] 정보통합시스템 구조

[그림 17]은 실제 구축된 정보통합시스템의 사용자 인터페이스 화면이다. 본 화면에서 보듯이 교통정보, 관광정보, 취업정보, 기상정보들이 하나의 시스템으로 통합되었고 [그림 18]은 관광정보와 기상정보를 통합하여 제공하는 화면을 보여주고 있다.



[그림 17] 정보통합 시스템



[그림 18] 관광정보와 기상정보가 통합된 정보 제공 화면

3) Client Application 개발

Client Application의 개발은 웹 서비스를 지원하는 어떠한 언어로도 개발이 가능하다. 개발자는 사용자 인터페이스 부분만 신경을 쓰고 그 후 필요한 정보는 웹 서비스를 호출하여 사용자 인터페이스 부분에 연결함으로써 개발이 간단히 개발 할 수 있다. 완료되게 된다. 정보통합 시스템으로부터 정보서비스를 받도록 구현된 사용자 인터페이스다. [그림 19]는 C#으로 개발된 Client Application을 보여준다. 여기서는 기상정보와 실시간 재해정보, 실시간 도정정보를 웹 서비스로 제공받아 보여주고 있다. 그리고 [그림 20] 은 JSP로 구축된 사이트로 웹 서비스를 활용하여 관광정보를 활용하여 시스템을 구축하였다.



[그림 19] C#으로 개발한 Client Application

V. 결 론

본 논문에서는 각기 이기종 시스템에 산재되어 있는 정보시스템들의 정보를 분산객체기술중 웹 서비스를 활용하여 백엔드 수준으로 통합하여 시스템간의 통합과 상호 호환성에 대해 연구하였다.

정보시스템 통합의 대상은 관광정보시스템, 도정정보시스템, 교통정보시스템, 취업정보시스템, 재해.재난정보시스템 그리고 기상정보시스템등 6개의 정보시스템을 통합하였다. 이러한 정보시스템의 통합에는 웹 서비스 기술을 이용하였다. 이렇듯 웹 서비스를 이용하여 정보시스템을 통합한 결과 여러 행정 부서 혹은 외부 기관과의 유연한 협력과 통합이 가능하게 된다. 이를 바탕으로 부처간 협업제고 및 정보시스템의 질적 고도화를 위한 정보의 공동 활용 및 시스템 통합/연계를 통해 국민 중심의 정보서비스를 제공할 수 있다. 이렇듯 대 국민 행정 서비스 중 부처의 협조를 얻어서 처리하는 민원처리 시, 부처간 시스템이 통합.연계되는 경우 업무처리 기간단축으로 이어져 곧 정부의 생산성 향상과도 직결되어 정보의 활용도의 극대화가 기대된다.

이러한 정보시스템에 대한 통합은 단지 시스템 통합의 차원을 넘어선 국가경쟁력과도 밀접한 연관을 갖고 있다. 정보시스템간의 상호 호환성/상호운영성이 보장되는 정보시스템 상호간의 통합을 통해 정보서비스의 질을 극대화 할 필요가 있다.

향후 추가적으로 고려해야 될 사안으로 웹 서비스 활용 시 발생하는 부하에 대한 최소화 방안과 시스템의 보안에 대한 강화, 그리고 XML을 기반으로 한 SOAP나 WSDL문서의 암호화 및 압축에 대한 추가 보완이 필요하다. 마지막으로 추후 웹 서비스를 확장한 시맨틱 웹 서비스(Semantic Web Services)기술을 적용하여 통합된 정보의 검색등의 기능을 고려해야한다.

[참고 문헌]

- [1] 권기현 " 우리나라 전자정부와 행정개혁 : 패러다임.모형 그리고 개혁 " - 1999년 3월
- [2] 허정희, 김은주 "공공정보화 웹 서비스 도입방안 연구" - 2003년 12월
- [3] Aeleen Frisch, "Essential System Administration", 1995, O' Reilly & Associates, Inc.
- [4] 박남섭, 김태운 "CORBA 기반 원격 서버 관리 시스템의 설계 및 구현", 한국정보처리학회 춘계학술발표논문집Vol.9, No.1, [2002].
- [5] Microsoft Strategy for Distributed Computing and DCE Services, <http://www.microsoft.com/TechNet/Analpln/dce.asp>
- [6] CORBA Specification, Ver 2.4.2, OMG, <http://www.omg.org>
- [7] Java, RMI and CORBA, OGM, <http://www.omg.org/news/compare.html>
- [8] Richard Grimes, Professional DCOM Programming, Wxor Press.
- [9] 이경하, 이규철, "웹 서비스의 표준화 동향과 발전 방향", 데이터베이스연구 회지, VOL.19, No.01, pp.0080~0087, 2003. 03.
- [10] "Web Services Architecture", "<http://www.w3.org/TR/ws-arch>", 2004. 2. 11
- [11] SOAP Version 1.2 Part 0 Primer, <http://www.w3.org/TR/soap12-part0/>, 2004. 6. 25
- [12] Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, <http://www.w3.org/TR/wsdl20/>, 2004. 8. 3.
- [13] UDDI Technical White Paper, UDDI.ORG, http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf
- [14] David Orchard, "Web Services Pitfalls", http://www.xml.org/pub/a/2002/02/06/web_services.html, XML.com, Feb 2002.
- [15] Fabio Casati, Ming-Chien Shan, "Models and Languages for Describing and Discovering E-Services", Tutorial, Semantic Web Working Symposium, Stanford, USA, July 2001.
- [16] Heather Kreger, "Web Services Conceptual Architecture 1.0.

- <http://www-4.ibm.com/software/solutions/webservices/pdf/WSCA.pdf>, 2001, 4
- [17] Donald F. Ferguson, Tony Storey, Brad Lovering, John Shewchuk , “안전하고 신뢰성 있는 웹 서비스” , IBM DeveloperWorks, <http://www-903.ibm.com/developerworks/kr/web/services/library/ws-securtrans.html>, 2003, 10.
- [18] Eric Newcomer, “Understanding Web Services” , Pearson Education, 2002, 12
- [19] Henry Bequet, Meeraj Moidoo Kunnumpurath, Sean Rhody, Andre Tost, “Beginning Java Web Services” , Wrox Press, 2003, 5 .
- [20] S. Jeelani Basha, Scott Cable, Ben Galbraith, Mack Hendricks, Romin Irani, James Milbery, Tarak Modi, Andre Tost, Alex Toussaint, “Professional Java Web Services” , Wrox Press, 2002, 11.
- [21] Ray Lai, “J2EE Platform Web Services” , Prentice Hall, 2003, 12, 24
- [22] Harvey M. Deitel, Paul J. Deitel, Jonathan P. Gadzik, Kyle Lomeli, Sean E. Santry, Su Zhang, “Java Web Services for Experienced Programmers” , Pearson Education, 2003, 9, 17.
- [23] Steve Graham, Simeon Sumeonov, Toufic Boubez, Doug Dais, Glen Daniels, Yuichi Nakamura, Roy Neyama, “Building Web Services with Java (Making Sense of XML, SOAP, WSDL, and UDDI)” , SAMS Press, 2002.
- [24] David Chappell, Tyler Jewell , “Java Web Services” , O'Reilly, 2002, 5
- [25] Mark Endrei, Jenny Ang, Ali Arsanjani, Sook Chua, Philippe Comte, Pal Krogdahl, Min Luo, Tony Newling, “Patterns: Service-Oriented Architecture and Web Services” , IBM Red Books, 2004, 4
- [26] Java 2 Platform, Micro Edition(J2ME) Web Services A Technical White Paper, Sun Microsystems, http://java.sun.com/j2me/reference/whitepapers/Web_Svcs_wp072904.pdf
- [27] Jon Ellis, Mark Young, J2ME Web Services 1.0 Final Draft, Sun Microsystems, http://sunsdlc1-25.sun.com/servlet/EComFileServlet/j2me_web_services-1_0-fr-spec.pdf