공학석사 학위논문

SOAP를 이용한 웹 기반의 원격지 데이터 수집 시스템 구현

Implementation of Web based Remote Data Collection System using SOAP

지도교수 임 재 홍

2006년 2월

한국해양대학교 대학원

전자통신공학과

박 시 형

목 차

제 1 장 서 론 ····· 1

제 2 장 오브컴 통신 및 SOAP의 개요 ····· 3

2.1 오브컴 통신 ····· 3

2.2 오브컴 통신의 응용 서비스 분야 ····· 6

2.3 SOAP ····· 15

제 3 장 원격지 데이터 수집 시스템 설계 ····· 32

3.1 E-mail 서버의 구성 ····· 32

3.2 메일 확인 데몬 및 데이터베이스 응용프로그램 ····· 34

3.3 웹 서비스 서버와 클라이언트 ····· 35

제 4 장 원격지 데이터 수집 시스템의 구현 ····· 41

4.1 메일 확인 데몬 ····· 41

4.2 DB 뷰어 ····· 47

4.3 웹 서비스 ····· 48

제 5 장 결 론 ····· 56

참 고 문 헌 ····· 58

Abstract

Orbcomm is a wireless telecommunications company that provides reliable, cost effective data communications services to customers around the world through its unique low-earth orbit satellite network and global ground infrastructure. A diverse customer base, including industry leaders General Electric, Caterpillar Inc. uses the Orbcomm services to track, monitor and control mobile and fixed assets including trucks, containers, marine vessels, locomotives, heavy machinery, pipelines, oil wells, utility meters and storage tanks anywhere in the world. Such the Orbcomm services are using system of Internet communication between programs. But it is important for application development to allow Internet communication between programs. Today's applications communicate using Remote Procedure Calls between objects like DCOM and CORBA, but HTTP was not designed for this. RPC represents a compatibility and security problem. Firewalls and proxy servers will normally block this kind of traffic. A better way to communicate between applications is over HTTP, because HTTP is supported by all Internet browsers and servers. SOAP was created to accomplish this. SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages.

Therefore, all sites of internet is possible access through SOAP. This paper includes characteristic of SOAP in Orbcomm service. Collected data by sensor sends to Orbcomm base station through

Orbcomm Satellite. And server supplies received data through WebService. Client uses SOAP when connect to server. This paper proposes web based remote data collection system using SOAP WebService system.

제 1 장 서 론

전 세계적으로 300개 이상의 회사가 오브컴 서비스(ORBCOMM Service)를 사용하고 있으며, 다양한 종류의 서비스를 제공하고 있다. 오브컴 서비스는 선박, 차량 등의 물류 위치추적 및 감시, 산업시설 감시, 기상관측 및 환경감시, 그리고 메시지전송 등을 들 수 있으며, 이러한 오브컴 서비스는 기본적으로 메일 서비스 시스템(Mail Service System)으로 이루어져 있고, 본 논문에서 구현한 시스템은 오브컴사에서 제공해주는 메일 서비스 시스템 중에서 두 가지를 보완하는데 중점을 두었다.[1]

첫 번째로, 오브컴사에서 제공해주는 메일 서비스는 메일 데이터를 사용하는 사용자가 전문가가 아니라면 데이터 내용을 이해하기 난해하다는 단점이 있다. 본 논문에서 구현한 시스템은 수신된 메일 데이터 내용을 데이터베이스에 필드별로 구분해서 저장하기 때문에 메일 데이터를 사용하는 일반 사용자들도 한눈에 알아보기 쉽게 인터넷으로 데이터를 검색할 수 있도록 되어 있다.

두 번째로, 메일 서비스가 이루어질 때 사용하는 응용 계층 프로토콜(APL; Application Layer Protocol)의 경우 그들만의 약정된 전송 제어 프로토콜(TCP; Transmission Control Protocol) 또는 사용자 데이터그램 프로토콜(UDP; User Datagram Protocol) 포트를 사용하기 때문에 인터넷상에 설치되어있는 방화벽에 많은 제약을 받게 된다.

하지만 단순 객체 접근 프로토콜(SOAP; Simple Object Access Protocol)은 하이퍼텍스트 전송규약(HTTP; Hypertext Transfer Protocol)을 채택함으로써 방화벽의 제약을 받지 않고 불특정 다수의 클라이언트(Client) 또는 인터넷 상의 특정한 서버와의 원격 호출 절차(RPC; Remote Procedure Call)를 효율적으로 수행할 수 있도록 해준다.

이러한 SOAP의 특징은 HTTP 프로토콜을 사용하여 확장성 생성 언어(XML; Extensible Markup Language) 문서를 송수신하며, TCP/80 포트를 사용하여 일반적인 방화벽 통과는 물론, 메시지 전송에 편리하고, XML 사용으로 구조정의 편리 및 정보저장의 용이성 및 확장성이 우수하다. 또한 호환성이 우수하여 다른 프로토콜들과 결합하여 사용 가능하다. 즉 웹으로 접속 가능한 모든 사이트에 SOAP를 통한 액세스가 가능하다[2].

본 논문에서 구현한 시스템은 이러한 SOAP의 특징을 오브컴 서비스에 접목시킨 것이다. 센서에 의해 센싱된 데이터가 오브콤 통신을 통하여 오브컴 지구국에서 수신되고, 수신된 데이터를 웹 서비스 할 때 네트워크상에서 클라이언트와 서버 공급자(Provider)간에 메시지 요청과 응답을 해주는 방법으로 SOAP를 사용한 것이다.

본 논문의 구성은 제 2 장에서 관련연구 분야인 오브컴 통신과 기술관련 동향 그리고 SOAP의 기본구조, 전달과정, 속성 및 형식에 관해서 알아보고, 제 3 장에서는 오브컴 데이터를 이용한 원격지 데이터 수집 시스템을 설계하였다. 그리고 시스템을 설계할 때에는 이메일 서버, 메일 확인 데몬 그리고 데이터베이스에 접근하여 데이터를 검색하고 수정할 수 있는 부분으로 구분해서 설계하였다. 제 4 장에서는 제 3 장에서 설계한 원격지 데이터 수집 시스템을 구현하였고, 구현할 때에는 비쥬얼베이직, 엑세스2003, 닷넷을 사용하여, 웹 서비스를 할 때에는 서버와 클라이언트 상호간에 응용프로그램을 연동할 수 있도록 SOAP를 이용하였다. 마지막으로 제 5 장에서는 본 논문에서 제안한 시스템을 구현할 때 가질 수 있는 시스템의 장점 및 결론에 대하여 기술하였다.

제 2 장 오브컴 통신 및 SOAP의 개요

2.1 오브컴 통신

본 절에서는 전 세계적으로 사용되고 있는 오브컴 서비스란 과연 무엇인지를 정의하고, 오브컴 통신이 어떠한 분야에 적용되고, 서비스되는지에 대해서 설명하겠다.

2.1.1 오브컴 통신의 정의

오브컴사는 평균 825km 고도의 저궤도 위성을 이용한 전 세계를 실시간으로 양방향 데이터 및 메시지 통신서비스를 제공하는 통신망운용 및 부대설비 개발 사업자이다. 미국의 위성 및 발사체 제조사인 Orbital Science사와 캐나다 국제통신사업자인 Teleglobe사가 자본 컨소시엄으로 추진하고 있는 사업으로, 현재 190개국과 서비스 제공권을 획득한 바 있고, 현재 35기 오브컴 위성 발사로 세계 약 70개국에서 실용화되어 있다.

저궤도(LEO; Low Earth Orbit) 위성은 적도 상공 약 36,000km의 정지궤도 위성에 비해, 지상 825km 상공의 낮은 궤도에서 VHF대 위성서비스를 제공하므로 종래의 시스템과 비교하면 단말기나 안테나의 소형화 그리고 낮은 가격, 전송시간 단축, 기상조건에 강한 점 등이 장점으로 부각된다.

특히, 향후 계속 발사 예정인 위성은 Orbital Science사가 설계 제조한 페가서스(Pegasus) 수평발사기로 수행되므로 종전의 수직발사 방법과는 비용면에서 1/20정도로 투자비를 절감하고 있어 저요금 서비스가 가능하

다.

오브컴 위성은 지구를 약 100분 주기로 일주하므로, 한국 상공에는 상시 1~2기의 위성이 보인다. 따라서, 직경 5,000km 이내의 모든 도심지역뿐만 아니라 기존의 통신망에서 커버할 수 없는 산간지역 및 해상에서도 서비스를 제공하므로 통신 불감 지역이 거의 없다고 할 수 있다. 주요 용도는 기상 자료수집, 트럭과 선박 등의 이동체의 운행관리, 산업시설의 감시, 원격제어 및 E-메일 송수신용이다.

상기 서비스를 위해 주식회사인 코리아오브컴은 1999년 9월 법인을 설립하여 미국의 오브컴 Global사와 위성사용에 관한 라이선스(License) 계약을 체결하였고, 2000년 2월 12일 정보통신부로부터 기간 위성 휴대 통신(GMPCS; Global Mobile Personal Communication System by Satellite) 사업자로 허가를 받았다. 현재, 경기도 이천시 장호원에 관문 지구국 시설을 구축하여, 시스템 특성 시험을 완료하였으며, 2000년 11월 1일부터 상용서비스를 개시하였다[1].

2.1.2 시스템의 구성

오브컴 시스템은 광범위한 지역에 걸쳐 패킷 교환(Packet Switching)이 가능한 양방향 데이터 통신 시스템으로 가입자 단말기와 오브컴 관문국과의 통신은 저궤도 위성망을 통하여 이루어진다.

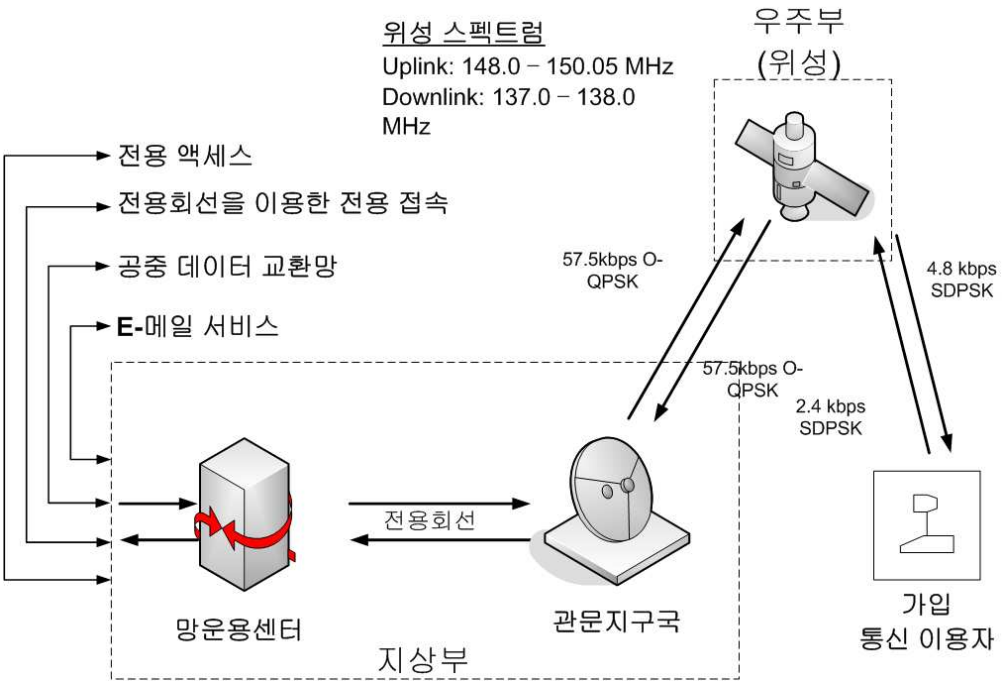
오브컴 관문국은 다이얼 업(Dial-up) 회선, 사설 전용회선(X.25, X.400), 인터넷 등을 통하여 가입자와 연결된다. 오브컴 시스템은 우주부, 지상부, 가입자부로 구성되어 있고, 전 세계 가맹국의 시스템 운용을 지원하는 네트워크 운용센터(NCC; Network Control Center)로 구성된다.

35개의 비정지 저궤도 위성으로 이루어진 우주부, 관문지구국(GES;

Gateway Earth Station)과 각 국에 위치한 망운용센터(GCC; Gateway Control Center)로 이루어진 지상부, 저궤도위성을 통하여 송수신하는 가입자 단말기(SC; Subscriber Communicator)로 이루어진 가입자부로 구성된다.

<그림 2-1>의 오브컴 시스템의 무선통신부는 137~138MHz, 148~150.05MHz의 VHF대역을 이용하며, 오브컴 위성은 4,800/9,600 bps로 가입자에게 패킷 데이터를 전송할 수 있는 송신기와 가입자 단말기로부터 2,400bps의 속도로 신호를 수신할 수 있는 다수의 수신기가 장착되어 있다.

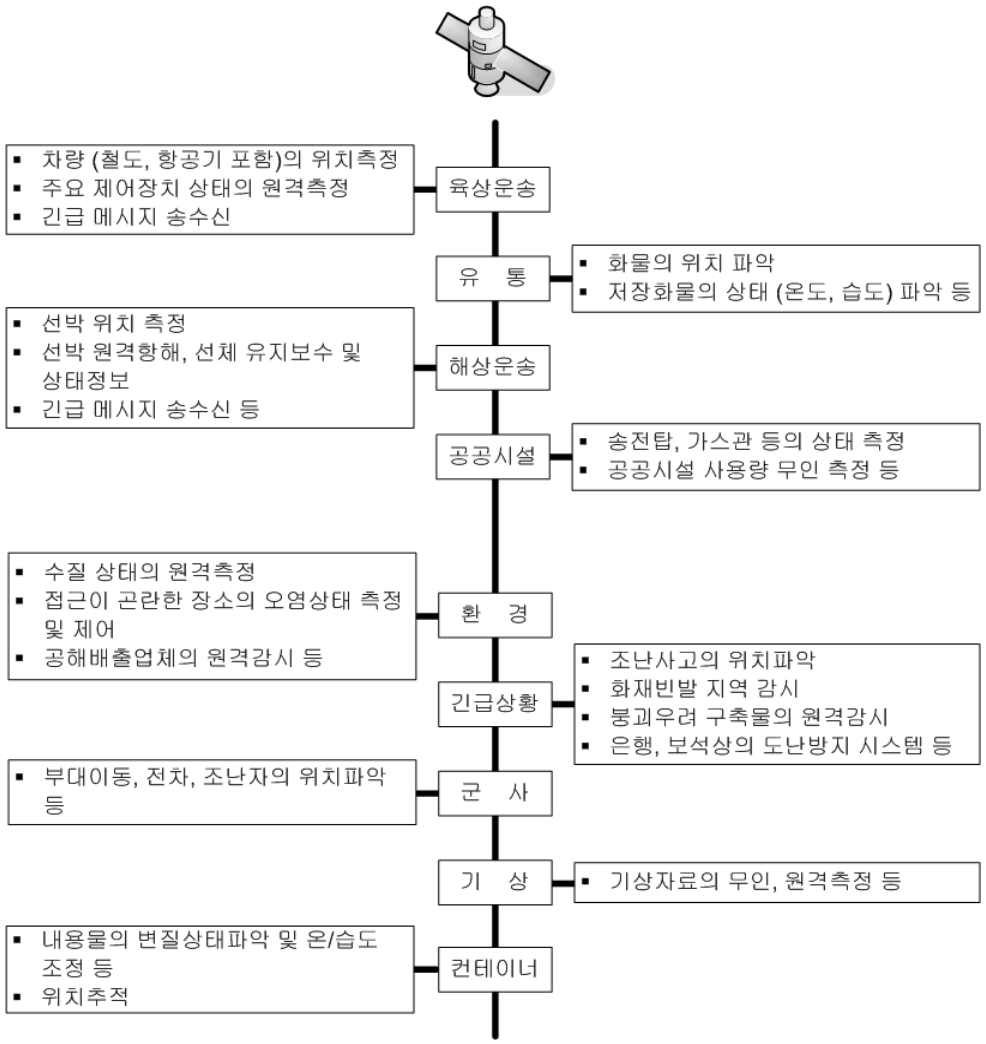
오브컴 위성망과 관문국간 링크가 불가능한 대양에서는 축적전송(Store and Forward)방식으로 위성 무선데이터 통신서비스를 제공할 수 있다.



<그림 2-1> 오브컴 통신 시스템의 구성도

2.2 오브컴 통신의 응용 서비스 분야

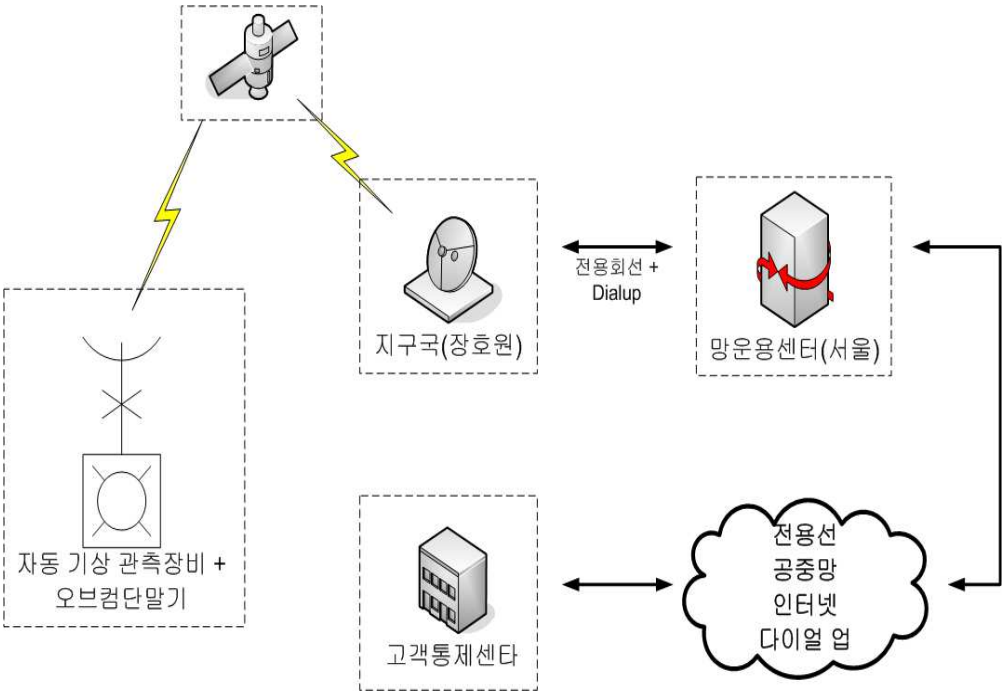
오브컴 통신 시스템의 서비스 응용 분야는 <그림 2-2>와 같이 원격지로 측정이 어려운 지역, 전기·통신시설이 미비한 지역, 비행장, 해양, 상공, 군 작전지역, 산악지역 등의 위치 기상정보에 사용될 수 있다.



<그림 2-2> 오브컴 통신 시스템의 서비스 응용 분야

2.2.1 기상

기온, 기압, 풍속, 풍향, 강수량, 습도, 대기오염정도 등을 알 수 있으며, 기대효과로는 유선이나, 육상 RF 시스템과 비교하여 장비설치가 용이하며, 요금이 저렴하고, 도서지역이나 산악지역, 해양. 상공등 기상관측이 힘든 지역의 관측이 용이하며, 원격으로 장비제어가 가능하다.



<그림 2-3> 오브컴 통신 시스템의 기상 서비스 분야

<그림 2-3>은 오브컴 통신 시스템의 기상 서비스 분야에서 오브컴 단말 시스템을 사용하는 예이다.

2.2.2 환경 감시

오늘날 현대 사회는 급속한 산업화 및 도시화가 이루어지면서 환경문제가 더욱 심각해졌다. 따라서 이러한 문제를 해결하기 위하여 환경문제가 되는 지역을 감시하는 오브컴 위성 시스템을 사용하여 방안을 제시할 수 있다. 그 감시 대상으로는 댐, 저수지, 강, 하천, 관개시설, 공장폐수 등의 수질오염감시 등이 있으며 수위, 수온, 강수량, 화재 등의 감시, 유독물 탱크, 에너지 등을 감시하고 제어할 수 있다. 또한, 장비의 효율성 증대, 데이터 신뢰성 향상, 비용절감, 원격장비 제어 가능, 이상이 발생했을 때 센터에 통보하는 것 등을 들 수 있다.

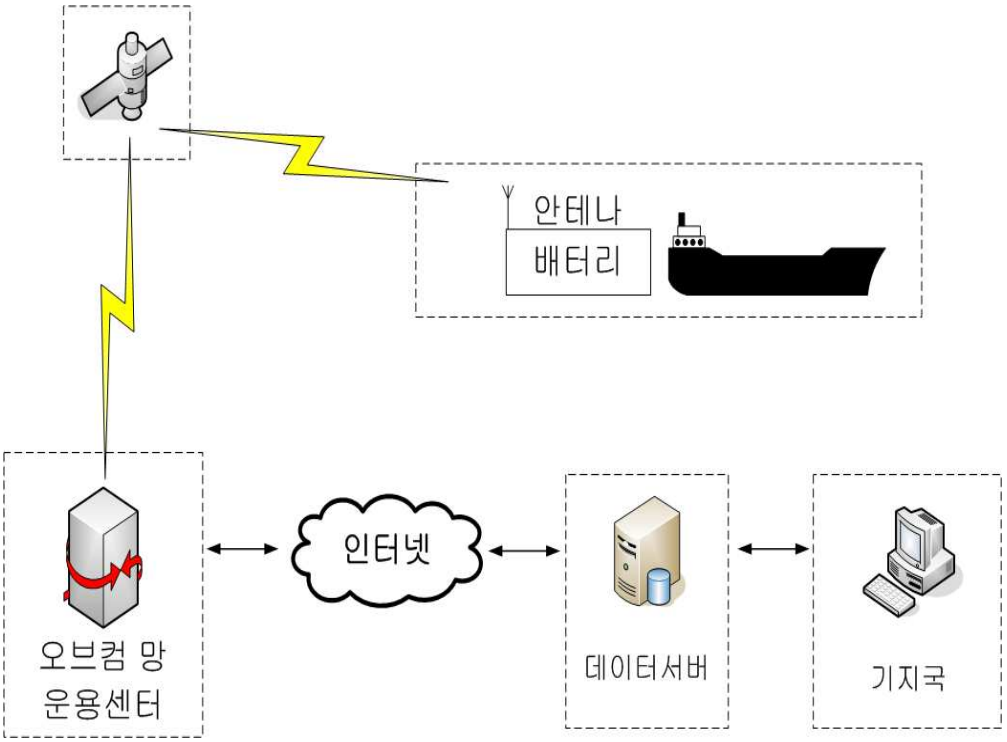
오브컴 위성망을 이용하면 수자원과 관련된 정보를 모니터링 할 수 있는 시스템으로, 이 시스템은 측정장비(센서포함), 오브컴 단말기, 태양열 전원장치로 구성되었으며, 수위, 수온, 강수량 및 수질상태 등을 측정하여 정기적으로 데이터를 송신할 수 있고, 원격으로 운용 파라미터(Parameter)값을 변경시킬 수 있다. 그리고 전원은 태양열 시스템을 이용하여 사용하고 있다.

2.2.3 물류・해양

오브컴 단말기의 위성위치확인시스템(GPS; Global Positioning System) 기능을 이용한 위치정보 및 메시지 전송을 이용하여 물류 시스템, 선박의 위치추적 등의 정보제공을 할 수 있다. 차량, 철도, 선박 등 운송수단의 위치정보, 주요 제어장치 상태의 원격측정, 화물의 위치파악, 저장화물의 온도・습도파악 및 원격조정, 양방향 메시지 전송 등의 측정 요소가 있으며, 운송수단 위치 및 정보탐지, 날씨, 교통상황, 화물정보 등의 유용한 정

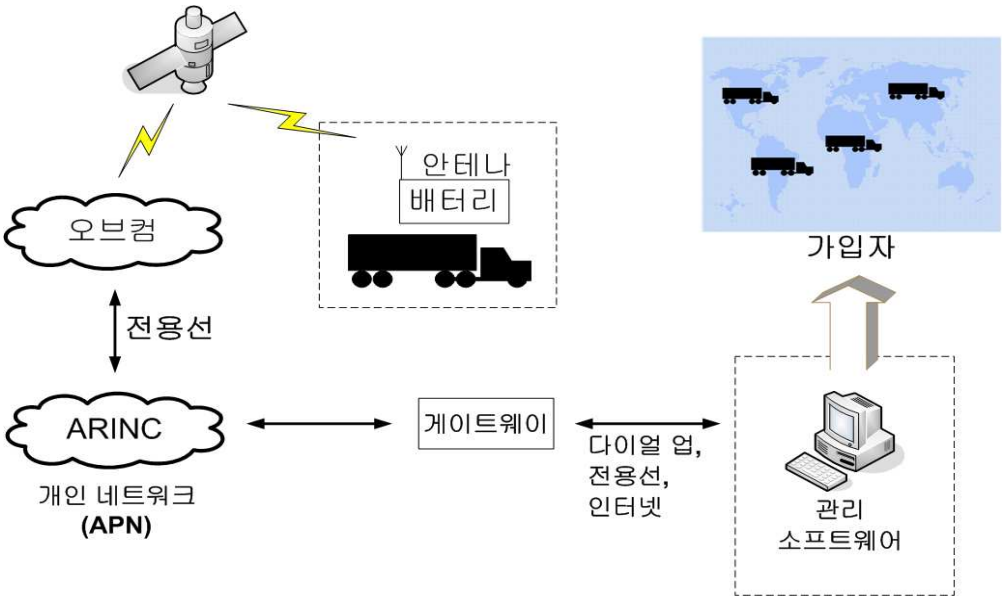
보제공으로 물류 관리의 효율성 증대의 기대효과를 얻을 수 있다.

<그림 2-4>와 같이 선박위치 추적 시스템은 다양한 콘텐츠(Contents)와 기능을 갖춘 독립된 웹 공간에서 다양한 서비스를 제공한다.



<그림 2-4> 선박 위치 추적 시스템

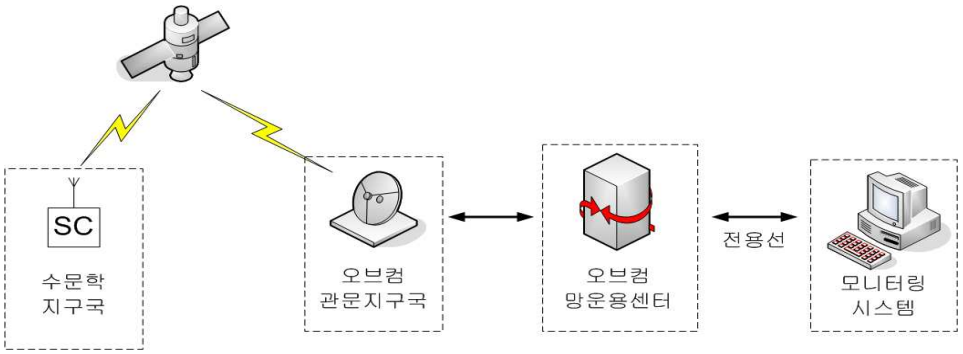
<그림 2-5>는 물류·환경 분야의 위치·제어시스템으로서 철도, 화물차량 등을 오브컴 위성망을 이용하여 위치추적, 감시, 제어할 수 있는 시스템이 있으며, 모니터 지도상에서 위치정보를 조회 및 이동경로를 알 수 있다. 냉동차량에 센서를 부착하여 냉동차의 온도, 화물적재 상태, 배터리 상태 등도 감시할 수 있다.



<그림 2-5> 철도, 화물차량의 위치·제어 시스템

2.2.4 수자원 관리

댐, 강, 하천, 해양, 수자원을 이용한 발전소 등 원격제어 지역 등의 장소에서 조류, 폭풍, 적조류 관측, 수량, 유속 등을 관측할 수 있다.

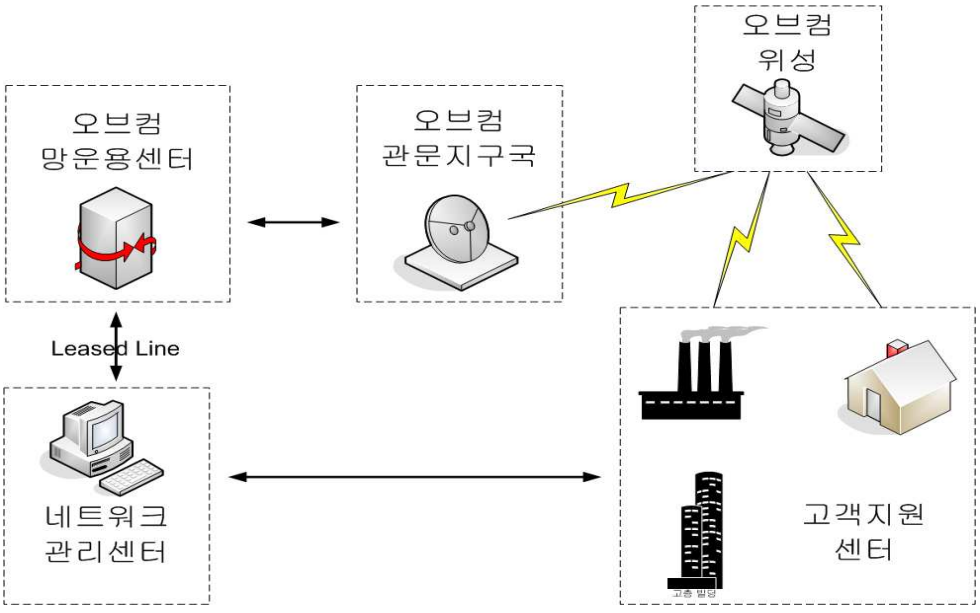


<그림 2-6> 수자원 관리 시스템

이러한 관측으로 인하여 수질 오염에 대한 대비, 학술적 연구 자료 제공, 이상 현상에 즉시 대비, 실시간 정보 제공, 긴급 재해 방지 등의 효과를 기대할 수 있다. 국가적 차원에서 한정된 수자원의 효율적 관리는 체계적이고 신뢰성 있는 통신망 구축이 필수적으로 요구되는 분야로 <그림 2-6>은 수자원 관리 시스템의 구성도이다.

2.2.5 전기가스 원격검침

<그림 2-7>의 전기가스 원격검침 시스템은 위험요소가 많은 전력계통의 서비스를 안전하게 관리하고 높은 신뢰성을 유지하며 통합 관리함으로써 저렴한 비용으로 관리경비 및 인건비 절감 효과와 업무의 실시간 자동화를 이루며 전산화로 기업의 경쟁력을 높일 수 있다.



<그림 2-7> 전기가스 원격검침 시스템

가정용, 산업용 전력 검침 지역, 누수가 잦은 곳의 안전사고 다발 지역, 수요가 적고 인건비가 높은 곳 등의 지역에 오독 방지, 에너지 사용량의 감시, 자동 검침 및 필요 시 수시 검침, GPS를 이용한 사고지점 표시 사고 기록・해석, 누전・가스 누출로 인한 위험지대 감시・통보, 정전 여부 검사, 다양한 데이터 수집, 부하 제어기능 갖추고 자동으로 전기사용량을 조절 등을 제어함으로써 인명 피해 방지, 실시간 정보로 에너지 절약, 낙뢰・태풍 등의 원인으로 계통 발생했을 때 순간적으로 사고차단, 건전 계통에 파급 방지, 인건비 절약 등의 효과를 기대할 수 있다.

2.2.6 SCADA

감시 제어 데이터 수집 시스템(SCADA; Supervisory Control And Data Acquisition) 시스템은 소규모 원격 단말기 장치(RTU; Remote Terminal Unit)와 원격감시, 제어, 계측하는 시스템으로써 수자원관리, 에너지 관리에서 교통관제, 변・발전소, 가스・송유관 원격감시, 제어, 측정에 이르기까지 전 분야에 이용 할 수 있다.

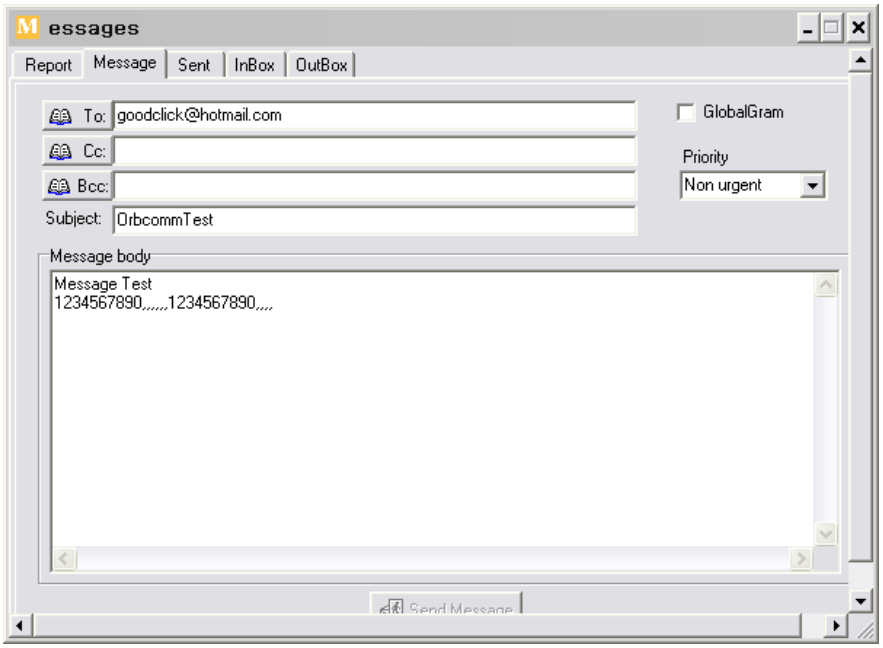
또한, 국제적 환경에 맞추어 표준화된 하드웨어 구조와 운용체제를 채택하고 분산화를 통해서 유연성을 확보하고 이중화구성을 통해서 시스템 안정성을 향상 시키며 오브컴 단말기를 통한 실시간 데이터베이스와 관계형 데이터베이스의 사용으로 시스템 및 업무의 효율을 높일 수 있다.

SCADA는 <그림 2-8>과 같이 중앙 처리장치, 데이터 처리장치(RTU), 통신 장치(오브컴 단말기)로 구성된다. 통합 관리 시스템(원격 검침・전력계통), 분산되어 원격 감시 및 제어가 필요한 곳(가스・송유관, 산업용 파이프라인), 효율적인 자동화 제어 시스템이 필요한 곳, 실시간 데이터처리 및 원격 데이터 제어, 처리가 필요한 곳, 사고 발생률이 높아 원격 관리가

필요한 곳 등에 이용할 수 있으며, 원격 검침, 전력 계통 원격관리, 원격지 현장 상태 감시, 긴급사고시 경보·이벤트 처리, 변·발전소의 비상 발전 제어, 수시·주기적 데이터 보고를 함으로써 실시간 데이터 취득 및 처리로 상시 모니터링 가능, 위성 데이터 통신으로 데이터 손실방지, 기상 악화에도 안정적인 데이터 전송, 효율적, 안정적인 시스템 관리 등이 가능하다.

2.2.7 E-메일 송수신

E-메일 송수신이 현재 사용되고 있는 인터넷에 가장 기본이 되면서 생활의 일부로 중요한 부분을 차지하고 있다. 일반 메시지의 전송은 공중망이나 기타 인터넷 상에 의존이 절대적이다.

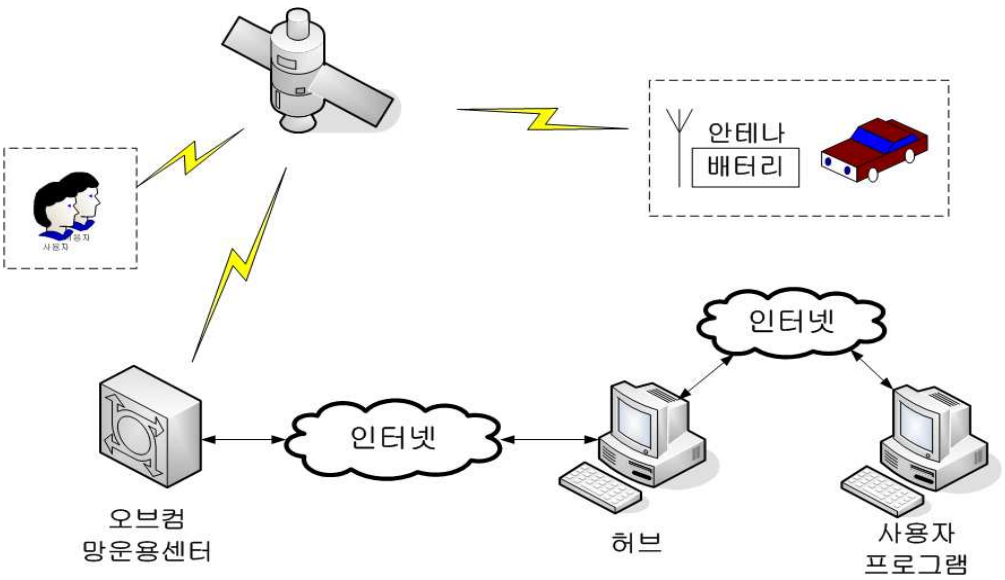


<그림 2-8> E-메일 송수신 서비스

<그림 2-8>과 같이 오브컴은 세계 오지 등 언제 어디서나 E-메일 송수신 서비스를 제공한다. 해상, 산악지역, 타 유선망이 설치되지 않은 지역, 중요 데이터 전문이 필요할 때, 긴급 구조를 요청할 경우와 실시간 정보 송수신이 필요할 때에 이용되며 단말기에서 단말기로 전송 가능하며, 윈도우 상태의 실행으로 편리하다. 또한 정보유실을 방지할 수 있으며, 그룹 호출도 가능하고, 각종 운영체제에서 실행 가능(Windows NT, Windows 95, Windows 98, Windows CE 지원) 하다. 또한 통신 음영지역을 해소할 수 있으며, 실시간 정보 송수신이 가능 하다.

2.2.8 위치추적 및 감시

오브컴은 단말기에 내장된 GPS를 이용한 인적, 물적 자산의 위치 추적 및 감시한다.



<그림 2-9> 위치추적 및 감시 시스템

각종 정보를 실시간으로 센터에서 파악할 수 있게 하는 기능과 수집된 여러 정보를 데이터베이스화하여 관리 할 수 있게 하는 기능이 제공 되며 시스템의 구성은 <그림 2-9>와 같다.

차량(화물, 여객), 선박, 해상, 산악 등의 지역에서 양방향 메시지를 이용한 차량의 위치 추적·운행 관리, 무인 기지국의 상태 감시의 이상 동작 시 센터에 경고, 무인 자산 관리에 대한 24시간 자동 감시·보고 등을 할 수 있다.

2.3 SOAP

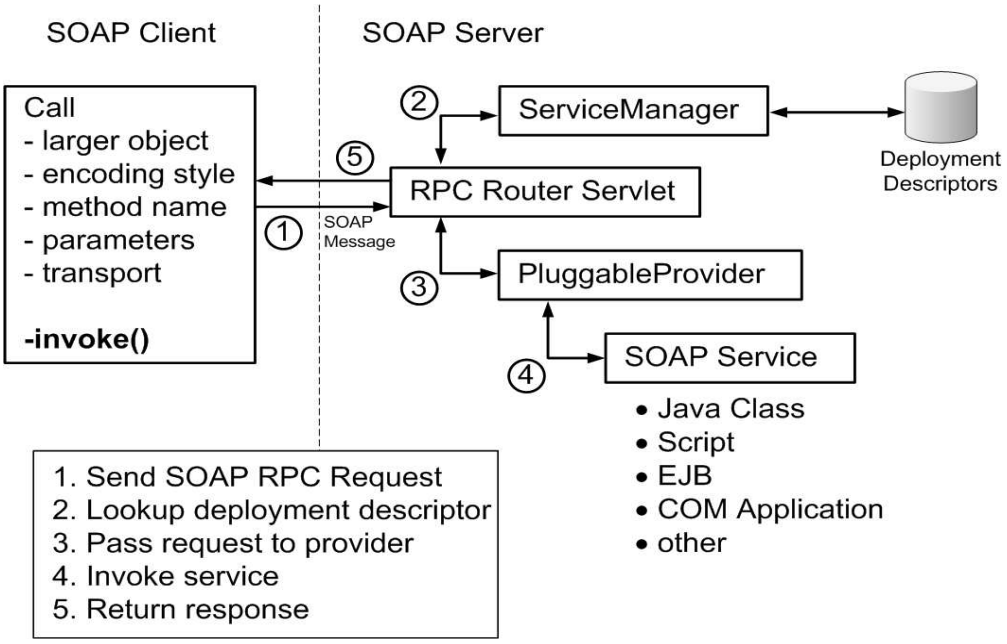
SOAP는 전자적 커뮤니케이션의 기본 프로토콜로서 XML로 작성된 간단한 요청과 응답 메시지들을 위한 모델이며 월드 와이드 웹 컨소시엄(W3C; World Wide Web Consortium)의 표준으로서, 분산처리 환경에서 정보의 교환을 목적으로 설계되었다. SOAP는 어떠한 전송 프로토콜과도 함께 사용될 수 있으며, 현재 HTTP가 가장 많이 사용된다. SOAP 메시징은 가끔씩 플랫폼·중립적인 원격 호출 절차 메카니즘으로 모델링되기도 하지만, 어떤 형태의 XML 정보를 교환하는데도 사용될 수 있다. W3C의 XML 프로토콜 담당 그룹은 앞으로 W3C의 추천사양이 될 SOAP 1.2라고 명명된 새로운 표준을 만들고 있는 중이다.

SOAP는 어떻게 XML 데이터가 모든 플랫폼 상의 미들웨어(Middlewere)와 하드웨어(Hardware)에 의해, 그리고 모든 프로그램 언어(Programming Langage)로, 모든 네트워크 프로토콜을 통해, 모든 객체 모델(Object Model)을 기반으로 지원될 수 있는 표준 메커니즘을 이용하여 교환될 수 있는지 대답해 준다.

SOAP의 “상호작용(loosely coupled)” 모델은 다양한 방법으로 e-비즈

니스 통합 문제를 해결한다. 예로, SOAP를 이용하여 서비스 요청자와 제공자 모두를 위한 개발을 단순화시킨다. 메시지 컨텐츠에만 집중함으로써, 밑단의 구현 절차에 대한 특별한 지식이 없어도 모든 서비스 요청자와 제공자 사이에서의 협업을 실현할 수 있으며, HTTP가 구성의 일부가 되기도 하지만, SOAP는 어떠한 프로토콜과도 사용이 가능하다.

웹 서비스 구현절차는 기존의 코볼 응용프로그램(COBOL Application), 신규 엔터프라이즈 자바 시스템, C++로 작성된 제품의 인터페이스, 또는 요청 메시지를 해석하고 적절한 응답을 제공하는 제반적인 것들도 포함된다. 이와 유사하게, 서비스 제공자도 요청자의 구현절차에 대해 아무런 정보도 알 필요 없으며, 서비스 요청자와 서비스 제공자 모두는 요청 및 응답 메시지의 형태와 내용만 알면 되므로, 바로 이것이 작업을 단순화시켜주는 요인이 된다.

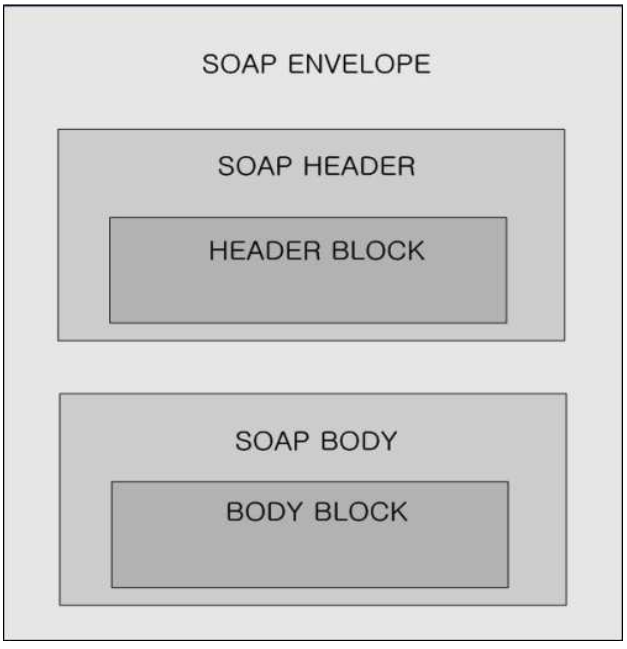


<그림 2-10> SOAP를 통한 클라이언트와 서버간의 통신 구조

<그림 2-10>의 메소드명(Method Name)과 아규먼트(Augment)를 포함하는 메시지의 원격 호출 절차 프로토콜로 사용되며, 객체-지향의 프로그래밍을 웹 기반의 원격 오브젝트로 확장하는 능력까지 지니고 있다. 또한 SOAP는 모든 형태의 XML 데이터가 포함된 문서(Document) 교환의 경우에도 사용될 수 있다. SOAP는 모든 형태의 시스템으로부터, 그리고 기업 내부와 비즈니스 파트너 간의 코드의 재활용을 완전하게 수행하며, 모든 형태의 시스템에서의 명쾌한 통합을 촉진한다.

2.3.1 SOAP 메시지의 기본 구조 및 전달 과정

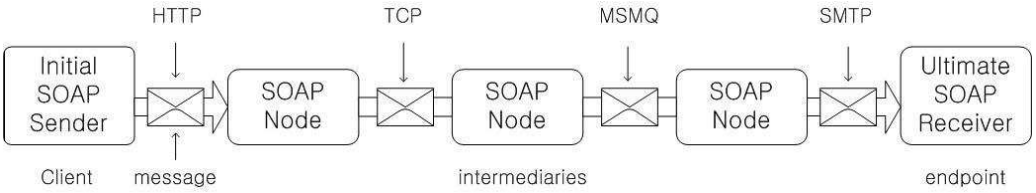
SOAP 메시지는 엔빌로프(Envelope), 헤더(Header), 바디(Body)의 3개 주요 파트로 이루어지며 <그림 2-11>의 구조로 구성된다.



<그림 2-11> SOAP 메시지의 구조

SOAP 메시지의 구성 요소의 첫 번째인 엔빌로프는 메시지의 시작과 끝을 정의하고, 또 다른 요소인 헤더는 메시지의 모든 조건적 속성들을 포함한다. 마지막 구성 요소인 바디는 전송될 메시지를 포함한 모든 XML 데이터를 포함한다.

SOAP 메시지는 <그림 2-12>와 같이 SOAP Sender에 의해 생성되어 여러 SOAP 노드(Node)들을 거치면서 임의적으로 SOAP Header에 대한 부분적인 처리를 하게하고 최종적으로 SOAP Receiver에게 전달되어 SOAP Header와 SOAP Body에 대한 특정 처리 및 작동을 촉발시킨다. SOAP의 스펙에는 어떻게 해서 SOAP 메시지의 전달 경로가 결정되며 어떻게 SOAP 메시지가 그 경로를 따라 전달되게 되는지에 대해서는 명시하지 않는다.



<그림 2-12> SOAP 메시지의 전달·처리 시나리오

2.3.2 SOAP Header 속성

SOAP Header내에는 하드웨어 처리에 대한 정보를 전달하는데 사용되는 몇 개의 속성(Attribute) 들이 정의 되어 있다.

가. role 속성

SOAP 메시지의 전달에 가담하는 SOAP 노드들은 각 메시지에 대한 몇 가지 특정 규칙들에 포함 되어 규칙에 따라 정해진 특정 기능을 수행

할 수 있다. 규칙은 헤더블록(Header Block)내에 “env:role”(env는 SOAP Envelope의 name space prefix) 속성의 값으로 지정된다. SOAP 노드는 role header의 값으로 해당 헤더블록이 자신이 처리해야 하는 블록인지를 판단한다.

<표 2-1>에서 모든 SOAP 노드들은 “next”라는 규칙에 디폴트로 속해 있다. 디폴트로 지정되어 있는 SOAP 규칙들은 아래 표와 같으며 SOAP 규칙들은 SOAP 응용프로그램의 필요에 따라 추가 정의 될 수 있다.

<표 2-1> 디폴트로 지정되어 있는 SOAP 규칙들의 표

Short-name	Name	Description
next	“http://www.w3.org/2003/05/soap-envelope/role/next”	각 SOAP 중개물과 최종 SOAP receiver는 이 role를 수행 한다.
none	“http://www.w3.org/2003/05/soap-envelope/role/none”	SOAP 노드들은 이 role를 수행하지 않아야 한다.
ultimateReceiver	“http://www.w3.org/2003/05/soap-envelope/role/ultimateReceiver”	최종 receiver는 이 role을 수행한다.
user-defined	user-defined	사용자가 지정

나. mustUnderstand 속성

<표 2-2>에서 현 SOAP 노드가 처리해야 하는 SOAP Header block중 반드시 SOAP 노드가 처리해야 하는 즉, 처리하지 못한다면 폴트를 발생시켜야 하는 헤더 블록에는 mustUnderstand 속성을 가지며 그 값을

“true”로 한다. SOAP 노드는 처리해야 하는 헤더 중 mustUnderstand 속성이 “true″인 헤더 블록은 반드시 처리할 수 있어야 한다.

다. relay 속성

SOAP 메시지가 여러 매개체 들을 지나가는 동안 만약 한 헤더 블록이 한 SOAP 노드에서 처리 되도록 되어 있었다면 그 헤더 블록은 외부로 가는(Out-Bound) 메시지에서는 지워지게 되어 다음 매개체에게는 전달되지 않아야 한다. 이때 relay 속성이 “true”로 설정되어 있으면 해당 헤더 블록은 한 SOAP 노드에서 처리 되도록 되어 있었지만 처리되지 않은 경우 지워지지 않고 다음 매개체에 전달된다.

<표 2-2> SOAP Header block의 relay 속성 값의 표

ROLE		**Header block**	
Short-name	**Assumed**	**Understood & Processed**	**Forwarded**
next	Yes	Yes	No, unless reinserted
		No	No, unless relay=“true″
user-defined	Yes	Yes	No, unless reinserted
		No	No, unless relay=“true″
	No	n/a	Yes
ultimateReceiver	Yes	Yes	n/a
		No	n/a
none	No	n/a	Yes

모든 SOAP 노드가 속해있는 next role의 경우 mustunderstand속성이 “true″이고, 프로세서 되면 전송(Forward) 되지 않으며, mustunderstand

속성이 “true″이고, 프로세서 되지 않더라도 relay 속성이 설정되지 않았거나 거짓으로 설정되어 있으면 전송되지 않는다. 사용자 정의된 규칙의 경우 특정 SOAP 노드가 그 규칙에 속하지 않은 경우는 해당 헤더블록을 처리하도록 가정되어 있지 않으므로 그 헤더 블록을 처리하지 않고 포워드 한다. 최종 수신 SOAP 노드는 전송할 곳이 없다. 규칙이 정의 되어있지 않은 헤더블록은 무조건 전송된다.

2.3.3 문서형식과 RPC 형식의 SOAP 메시지

가. SOAP 메시지의 형식 종류

SOAP 1.1 과 SOAP 1.2의 과거 2001년 12월 버전의 스펙에서는 <그림 2-13>과 같이 SOAP 메시지를 문서형식과 RPC 형식의 두 종류 형식으로 나누었다.

* 문서형식
 SOAP Body의 내용을 하나의 XML 문서로 처리
* RPC 형식
 SOAP Body의 내용을 RPC 호출을 위한 특정 형식에 맞게 처리

<그림 2-13> SOAP 메시지의 두 종류

그러나 최근 버전의 SOAP 1.2 스펙에서 문서형식이라는 용어는 사라졌으며, 대신에 RPC 호출과 응답의 SOAP 메시지 내 전송을 위한 동일한 표현만을 정의한다.

나. SOAP RPC

```
<xml version='1.0'>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/
soap-envelope" >
<env:Header>
...
</env:Header>
<env:Body>
<m:chargeReservation
env:encodingStyle="http://www.w3.org/2003/05/soap-
encoding" xmlns:m="http://travelcompany.example.org/">
<m:reservation xmlns:m=
"http://travelcompany.example.org/reservation">
<m:code>FT35ZBQ</m:code>
</m:reservation>
<o:creditCard xmlns:o=
"http://mycompany.example.com/financial">
<n:name xmlns:n=
"http://mycompany.example.com/employees">
</n:name>
<o:number>123456789099999</o:number>
<o:expiration>2005-02</o:expiration>
</o:creditCard>
</m:chargeReservation>
</env:Body>
</env:Envelope>
```

<그림 2-14> SOAP RPC 호출의 예

RPC 호출 메시지 요소는 <그림 2-14>와 같이 SOAP body의 첫 번째 child로 전달된다. 실행하고자 하는 웹 서비스 연산의 이름을 요소 명으로 지정하며, 그 연산들의 파라미터 명들과 같은 이름의 하위 요소들을 만들어 파라미터들의 값을 전달한다. RPC 응답 메시지 요소 역시 SOAP body의 첫 번째 child로 전달되며 호출된 웹 서비스 연산 이름의 뒤에 "응답"을 붙인 이름의 요소에 결과 값들을 포함시켜 전달된다.

chargeReservation이라는 연산을 resrevation, creditCard라는 파라미터들을 전달하여 호출하는 RPC 호출 메시지를 전달하는 SOAP 메시지이다.

```
<xml version='1.0'>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/
soap-envelope" >
<env:Header>
...
</env:Header>
<env:Body>
<m:chargeReservationResponse
env:encodingStyle="http://www.w3.org/2003/05/soap-
encoding" xmlns:m="http://travelcompany.example.org/">
<m:code>FT35ZBQ</m:code>
</m:chargeReservationResponse>
</env:Body>
</env:Envelope>
```

<그림 2-15> SOAP RPC 응답의 예

chargeReservation 연산의 실행 결과 값이 “chargeReservation” + “응답” 을 요소 명으로 하는 chargeReservationResponse 요소에 포함되어 SOAP Body의 첫 번째 child로 응답되며 <그림 2-15>와 같다.

다. 문서 형식

문서 형식으로 정의된 웹 서비스는 요청과 응답 SOAP 메시지 모두 SOAP Body의 내용이 하나의 XML 문서로 구성된다. 그 내용은 어떤 XML이 오든 상관이 없다.

2.3.4 SOAP 에러

SOAP 메시지에 에러 정보를 전달하기 위해 사용한다. SOAP Body내에 유일한 요소로 폴트 요소를 포함시킨다. 본 논문에서는 SOAP 1.2의 SOAP 폴트의 경우에 대해 알아본다. SOAP 폴트는 폴트 요소로 표현하며 그 sub-element들은 <그림 2-16>과 같다.

필수 **Sub element : Code, Reason**
옵션 **Sub element : Node, Role, Detail**

<그림 2-16> sub-element

가. 코드

<그림 2-17> 코드 sub-element value 요소 값은 “VersionMismatch”, “MustUnderstand”, “DataEncodingUnknown”, “Sender”, “Receiver”로 제한되며 subcode 요소의 sub-element value 요소의 값은 SOAP 의 실행에 따라 정의해 넣을 수 있다.

```
<code>
< value>…< /value>
< subcode>
< value >…</ value >
< /subcode>
</code>
```

<그림 2-17> 엘리먼트 구조

<그림 2-18>은 Reason 엘리먼트의 구조이다.

<표 2-3> 폴트 코드들

VersionMismatch	엘리먼트가 스펙에서 지정한 로컬네임과 네임스페이스를 가진 요소가 아닐 때
MustUnderstand	mustUnderstand 속성 값이 “true”이면서 폴링 노드를 타겟으로 하는 헤더의 직속 요소를 이해하지 못할 때
DataEncodingUnknown	촐링 노드에서 지원하지 않는 데이터 인코딩 스타일이 적용 되었을 때
Sender	메시지가 올바른 구조를 가지지 않거나 충분한 정보를 전달하지 않을 때
Receiver	메시지 자체에는 문제가 없으면서 메시지를 처리하는데 문제가 생긴 경우

나. Reason

폴트에 대한 사람이 읽을 수 있는 설명을 전달한다.

```
<Reason>
      text
<Reason>
```

<그림 2-18> Reason 엘리먼트의 구조

다. 노드

SOAP 메시지 경로에서 어느 SOAP 노드에서 폴트를 발생시켰는지를 전달한다.

라. Role

폴트가 발생할 시점에 폴팅 노드가 수행하고 있던 규칙을 전달한다.

마. Detail

SOAP body 처리에 관련한 응용프로그램 특성 에러 정보를 전달한다. 웹 로직내부의 SOAP 엔진의 경우 스텍의 트레이스 정보를 전달한다.

2.3.5 프로토콜 바인딩

SOAP는 전달되는 메시지의 구조를 정의한 스펙으로, 실제로 SOAP 스펙에 따라 생성된 SOAP 메시지는 다양한 근원적인 전송 프로토콜들에 따라 전달 될 수 있다. SOAP 메시지의 전달에 사용되는 전송 프로토콜로는 HTTP, 메일 송수신 프로토콜(SMTP; Simple Mail Transfer Protocol), 파일 전송 프로토콜(FTP; File Transfer Protocol), 인터넷 메일 교환을 위한 멀티미디어 문서타입 정의(MIME; Multipurpose Internet

Mail Extension) 등 여러 가지가 있다. (주로 HTTP가 널리 쓰인다.) SOAP 메시지 스펙과 한 전송 프로토콜간의 연계를 바인딩이라 한다.

가. SOAP와 HTTP의 비교

SOAP 메시지는 다양한 전송 프로토콜을 사용할 수 있지만 웹의 범용성과 보편성에 따라 HTTP 프로토콜을 일반적으로 사용하고 있다.

(1) POST 방식 요구

<그림 2-19>의 POST 방식은 SOAP 요구를 그대로 HTTP Body에 넣어 전송한다.

```
POST /Reservations HTTP/1.1
Host: travelcompany.example.org
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: nnnn
<xml version='1.0'>
<env:Envelope xmlns:env=
"http://www.w3.org/2003/05/soap-envelope" >
<env:Header>
<env:Body>
</env:Body>
</env:Envelope>
```

<그림 2-19> SOAP HTTP POST 예

(2) GET 방식 요구

이때 /travelcom.example.org/reservations 는 SOAP 메시지를 응답하는 응용프로그램이다.

```
GET /travelcompany.example.org/reservations?code
=FT35ZBQ HTTP/1.1
Host: travelcompany.example.org
Accept: text/html;q=0.5, application/soap+xml
```

<그림 2-20> SOAP HTTP GET 예

Accept 헤더가 application/soap+xml 값을 가짐으로써 이 요구가 SOAP 응답을 바라고 있음을 알린다.

(3) 응답

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: nnnn
<xml version='1.0'>
<env:Envelope xmlns:env=
    "http://www.w3.org/2003/05/soap-envelope">
<env:Header>
...
</env:Header>
<env:Body>
...
</env:Body>
</env:Envelope>
```

<그림 2-21> SOAP HTTP 응답 예

SOAP 응답 메시지는 HTTP Response Body에 넣어 전송된다. HTTP "Content-Type" 헤더가 "application/soap+xml"의 값을 가진다.

나. SOAP 과 E-메일

<그림 2-22>는 SOAP 메시지를 SMTP를 사용하여 이메일 송수신하는 예를 보여주는 것이다.

```
From: a.oyvind@mycompany.example.com
To: reservations@travelcompany.example.org
Subject: Travel to LA
Date: Thu, 29 Nov 2001 13:20:00 EST
Message-Id:
<EE492E16A090090276D20842490C0C@mycompany.example.com>
Content-Type: application/soap+xml
<xml version='1.0'>
<env:Envelope xmlns:env=
    "http://www.w3.org/2003/05/soap-envelope">
<env:Header>
….
</env:Header>
<env:Body>
…
</env:Body>
</env:Envelope>
```

<그림 2-22> SOAP 메시지의 SMTP 프로토콜 사용 예

2.3.6 SOAP 인코딩

SOAP의 기원은 웹 서비스의 개념이 유아기에 머물러 있을 때부터 시작한다. 당시에 사람들은 SOAP를 분산 구성요소 객체 모델(DCOM; Distributed Component Object Model), 분산 프로그램 객체 관리 규격(CORBA; Common Object Request Broker Architecture), 원격 객체 호출과 같은 분산된 객체 기술들을 XML, HTTP의 특정 목적으로만 사용되도록 고안된 인터넷 기술로 통합하는 유용한 수단으로 사용할 계획을 가지고 있었다. SOAP의 목표는 여러 기술들이 각기 선호하는 다양한 이진 메시지 시리얼 포맷들 대신에 XML 기반의 메시지들을 생산하고 소비하는 연결관을 구축하는 것 이었다

분산된 응용프로그램에서 메시지를 생산하고 소비하려면 이들 메시지들이 어떻게 보여 져야 하는지에 대해 알고 있어야 하며, 이를 위해 대부분의 분산 객체 시스템들은 메타데이터의 구성요소 객체 모델(COM; Component Object Model)타입 함수들, CORBA 인터페이스 저장들, 또는 자바 함수 파일들과 같은 바이너리 표현들을 이용한다. 그러나 이런 식으로는 분산 환경의 클라이언트에서 사용할 기술을 알 수 없어서 알맞은 기술의 바이너리 표현을 제공할 수 없는 등의 무수한 문제가 있다.

SOAP의 프로그래머들은 이 문제에 대해, SOAP에서 이들이 서로 메시지를 기술할 수 있는 공통의 새로운 방식을 정의하지 않을 것이라면 적어도, 공통의 객체지향 프로그래밍 구성체들을 XML로 맵핑(Mapping)하는 지침이라도 제공해야 한다고 결정했다.

그 결과 SOAP는 그래프에 기초한 데이터 모델을 정의하고, 그 SOAP 데이터 모델의 사례를 SOAP 메시지로 직렬화하는 방식을 설명한 SOAP 인코딩 규칙을 작성하였다. 그리고 각 기술별로 각각 고유의 형태에 따르

는 응용프로그램들이 정의한 데이터 구성들과 값들의 데이터들을 SOAP 데이터 모델로 맵핑하는 일은 SOAP 작성자들에게 과제로 남겨졌다.

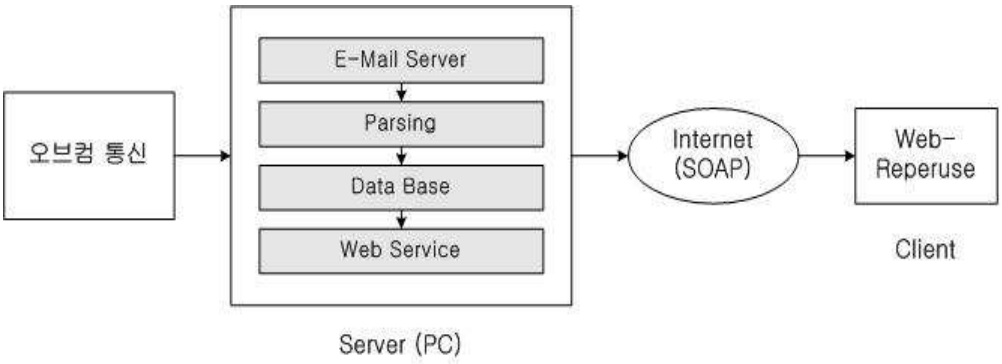
SOAP RPC Invocation, Response 는 각각 하나로 구성된 SOAP 데이터 모델로 다시 선언되어 SOAP 인코딩 규칙에 따라 SOAP 메시지를 구성하게 된다. 따라서 SOAP 인코딩을 더 이상 사용하지 않는 쪽으로 흐르고 있다. SOAP 인코딩 규칙들은 XML 스키마(Schema)에 대한 작업이 진행되고 있던 시점에 생성되었다. 지금은 XML 스키마가 완성되었기 때문에 개발자들은 간단히 XML 스키마 정의를 제공하여 요구/응답 메시지들이 어떻게 구성 되어야 하는지를 명시할 수 있으며 이 방법이 SOAP 인코딩을 사용하는 것에 비해 명시적이고 간단하여 많은 개발자들이 SOAP 인코딩에서 XML 스키마로 옮겨 가고 있는 실정이다.

SOAP 1.2 명시서에서조차 그 내용에 SOAP 데이터 모델, SOAP 인코딩, SOAP RPC 표현의 사용은 선택적이라고 언급 하고 있다. 즉, SOAP 툴킷류의 응용프로그램들은 SOAP 데이터 모델, SOAP 인코딩, SOAP RPC 표현을 꼭 지원할 필요가 없다는 것이며 사용자 또한 마찬가지이라는 것이다.

W3C 웹 서비스 기술 워킹 그룹에서는 WSDL 1.2 버전에 이르러서는 인코딩에 대한 지원이 없을 것이라 전망되고 있다. WS-I 베이직 프로파일 워킹 그룹의 정보 처리 상호 운용 가이드의 최근 초안된 버전에서는 SOAP 인코딩의 사용을 허락하지 않고 있다.

제 3 장 원격지 데이터 수집 시스템 설계

본 장에서는 오브컴 통신을 통한 원격지로부터 수집된 데이터를 원하는 유저들에게 제공하기 위한 웹 서비스 시스템 설계에 대해 설명하겠다. 원격지로부터 수집된 데이터를 가공하기 위한 E-메일 서버, 메일이 수신되었는지를 확인하기 위한 메일 확인 데몬, 그리고 들어온 메일을 저장하기 위한 DB, 웹 서비스를 제공하기 위한 서버와 클라이언트의 통신으로 구분하여 설계한다.



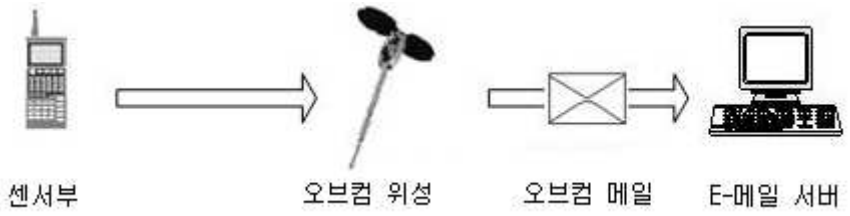
<그림 3-1> 원격지 데이터 수집 시스템 설계 구성도

3.1 E-메일 서버의 구성

만약 오브컴 망운용센터로부터 보내어지는 오브컴 메일을 우리가 일반적으로 사용하는 아웃룩 익스프레스(Outlook Express)나 한메일과 같은 웹 메일 서비스로 받아들인다면 메일을 받은 편지함에는 오브컴 메일과 여러 다른 곳으로부터 수신된 메일이 함께 섞여 있을 것이다. 또한 이 편지함에는 스팸메일까지도 있을 수 있다. 이것은 오브컴 메일로부터 데이

터를 추출하여 DB를 구성하는데 많은 문제점이 발생할 수 있는 가능성을 제공함으로 본 논문에서는 오브컴 망운용센터로부터 수신된 오브컴 메일을 위한 전용 E-메일 서버를 구성하여 내가 원하는 경로의 폴더에 저장하도록 하였다.

<그림 3-2>는 원격지로부터 수집된 데이터만을 전문적으로 취급하기 위한 E-메일 서버를 보여주고 있으며, 설치한 E-메일 서버는 EMWAC 서버이다.



<그림 3-2> 오브컴 메일을 수신하기 위한 E-메일 서버 구성도

EMWAC은 European Microsoft Windows NT Academic Centre에서 만든 Windows NT용 메일서버이다. 프리웨어 메일 서버 프로그램이며, Intel/Alpha/Power PC/MIP등에서도 메일서버구현이 가능한 프로그램이다. 이러한 EMWAC 중요한 기능들은 SMTP, POP3 기능, 한글 전송, 텍스트 및 바이너리 파일의 첨부가능, 메일링 리스트, 메일 포워딩, 자동 재전송 기능, 메일전송에 문제가 있을 경우 자동적으로 메일서버 관리자에게 메시지 전송 등이 있다. EMWAC 하나만으로 SMTP, POP3가 가능하다.

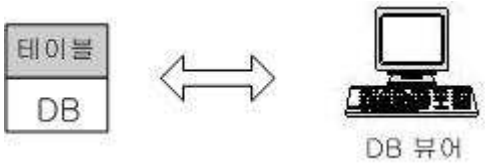
문제는 EMWAC 하나만으로 메일서버를 구축하면, 스팸메일 서버로 이용될 확률이 높다. 따라서 최근에는 IIS SMTP와 EMWAC POP3를 이용한 메일서버를 구축하고 있다.

3.2 메일 확인 데몬 및 데이터베이스 응용프로그램

메일 확인 데몬에서는 E-메일 서버로 수신된 오브컴 메일이 있는지를 계속 감시하는 부분이다. 이 데몬 프로그램에서는 E-메일 서버를 통하여 오브컴 메일이 저장되어 있는 폴더를 계속 검색하다가 새로운 메일이 수신될 경우 이 메일을 파서하여 메일안의 데이터를 DB에 갱신하기 위해 설계된 부분이라 정의할 수 있겠다.

<그림 3-3> 오브컴 메일을 수신여부를 판독하기 위한 메일 확인 데몬

원격지로부터 수신된 메일을 파서하여 원하는 데이터를 테이블에 저장하기 위하여 본 논문에서는 엑세스 2003을 이용한다.



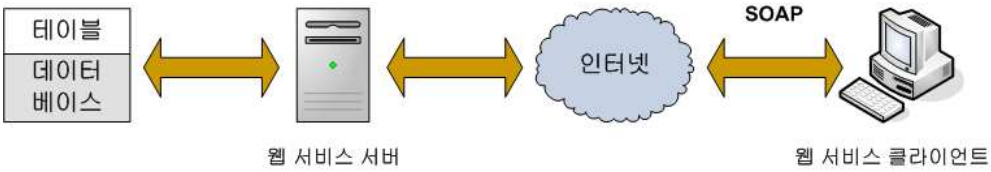
<그림 3-4> DB 테이블과 DB 뷰어

DB를 구성할 때 유의해야 할 점은 DB 테이블에 저장된 데이터 날짜와 센서로부터 센싱된 데이터의 날짜를 비교하여 새로 들어온 데이터인지 아니면 기존에 있었던 데이터인지를 비교하는 디폴트값으로 이용할 것이라

는 점을 유의 할 필요가 있다.

3.3 웹 서비스 서버와 클라이언트

XML 웹 서비스는 모바일 장치에서 원격함수나 데이터를 액세스할 수 있게 한다. 이들은 인터넷이나 인트라넷에서 HTTP를 통해 함수에 액세스하기 위해 웹 표준을 사용한다. XML 웹 서비스를 제공하는 서버에는 함수가 호출되는 동안과 응답이 단말 장치로 리턴되는 때에만 연결되어 있으면 된다.



<그림 3-5> 웹 서비스 서버 클라이언트

XML 웹 서비스로의 액세스는 윈도우즈 보안과 통합되고, 평범한 텍스트로 저장된 사용자 이름과 암호로 보호하거나, 익명 사용자를 구성될 수 있다. 또한 단말 장치에 응용프로그램은 추가로 소프트웨어를 설치하지 않아도 XML 웹 서비스를 사용할 수 있다.

3.3.1 XML 웹 서비스 구조

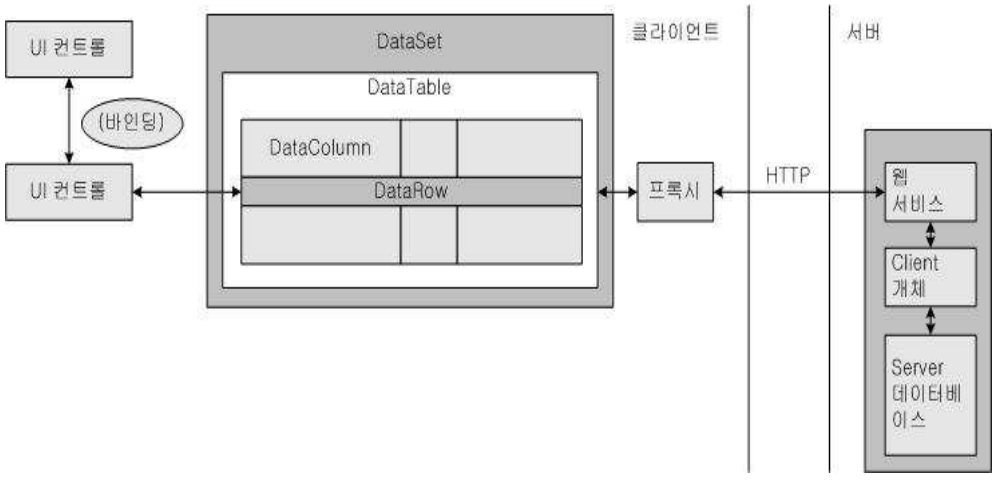
XML 웹 서비스는 네트워크를 통해 액세스할 수 있는 비즈니스 로직을 게시할 수 있게 한다. 데이터는 XML로 설명하고 표현한다. 그리고 표준

HTTP를 사용하여 전송되기 때문에 이진 데이터가 통과하지 못하도록 구성된 방화벽을 통과할 수 있게 된다. XML 웹 서비스에서 제공하는 연산은 표준 포맷으로 XML데이터를 사용하여 자기 자신을 설명한다.

웹 서비스가 두 응용프로그램을 중립적이면서 표준 방식으로 연결하기 때문에 레거시 시스템을 사용할 때, 인터넷에서 서비스를 처리하도록 제공할 때와 같은 경우에 적합하다.

예를 들어서 마이크로소프트의 맵 포인트 닷넷 웹 서비스를 비롯하여 인터넷에는 지도 서비스가 많아지고 있다. 맵 포인트 닷넷은 XML 웹 서비스로 공개되고, 인터넷을 통해 지리 정보를 액세스하고, 응용프로그램에 통합할 수 있다.

웹 서비스는 데이터베이스를 어떻게 연결하는지, 데이터베이스 객체를 클라이언트 응용프로그램에서 어떻게 사용하는지에 대해서는 모르는 채로 단말장치로 데이터를 가져온다. 닷넷 컴팩트 프레임워크를 사용하여 개발한다는 점에서 웹 서비스를 사용하는 것은 장점이 있다.



<그림 3-6> 웹 서비스와 서버 데이터

XML 웹 서비스는 닷넷 프레임워크와 대용량 기계의 모든 리소스를 가지고 서버에서 실행되도록 만들어진다. XML 웹 서비스는 ASP 닷넷을 사용하고, 마이크로소프트의 IIS에서 실행된다. 단말장치는 웹 서비스에 연결하여 메서드를 호출할 수 있는 프록시 객체를 만들기만 하면 된다.

<그림 3-6>은 서버 데이터베이스를 가지고 단말 장치를 연결하기 위해 사용되는 컨텍스트를 보여준다.

3.3.2 XML 웹 서비스 인프라

웹 서비스의 위치를 알아내거나, 서비스를 사용하는 방법 또는 XML 웹 서비스를 위한 통신을 포맷하는 표준을 제공해주는 인프라에 대해서 설명하도록 한다.

XML 웹 서비스 인프라는 XML 웹 서비스 디렉터리, XML 웹 서비스 검색, XML 웹 서비스 기술, XML 웹 서비스 포맷과 같은 서비스를 포함하고 있다.

3.3.3 XML 웹 서비스 디렉터리

외부에서 제공되는 웹 서비스를 응용프로그램에 추가하고자 한다면 그 서비스의 위치를 알아내기 힘들 것이다. XML 웹 서비스 디렉터리는 외부에서 제공하는 웹 서비스에 대한 설명의 위치를 제공한다.

마이크로소프트가 http://uddi.microsoft.com 웹 사이트에서 관리하는 UDDI(Universal Discovery, Description, Integration)가 인터넷 기반의 웹 서비스 디렉터리 예이다. UDDI 같은 웹 서비스 디렉터리는 XML 웹 서비스에 대한 정보를 게시하고 찾아내는 표준 방법을 정의한다. XML 웹

서비스 클라이언트가 레지스터리에 들어있는 정보를 질의할 수 있도록 프로그램에 검색 루틴을 넣을 수 있다.

XML 웹 서비스 디렉터리에 들어있는 정보는 비즈니스 정보, 서비스 정보, 바인딩 정보, 문서와 스펙과 같은 네 개의 카테고리로 나눌 수 있다.

이 정보는 검색 문서에 들어 있고, 이 문서는 XML로 작성되어 있다. 웹 참조를 추가하는 동안 비쥬얼 스튜디오 닷넷은 마이크로소프트에서 호스트(Host)하는 UDDI 사이트를 보여준다.

3.3.4 XML 웹 서비스 검색

```
<xml version="1.0" encoding="utf-8">
<discovery xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xmlns="http://schemas.xmlsoap.org/disco/">
<contractRef
ref="http://192.168.0.6/ProjectWebService/TimeSheetService
.asmx.wsdl"
docRef="http://192.168.0.6/ProjectWebService/TimeSheetSer
vice.asmx
xmlns="http://schemas.xmlsoap.org/disco/scl/"/>
<soap address=
http://192.168.0.6/ProjectWebService/TimeSheetService.asmx
xmlns:q1="http://tempuri.org/binding=q1:TimeSheetService
Soap"
xmlns="http://schemas.xmlsoap.org/disco/soap/"/>
<discovery>
```

<그림 3-7> 웹 서비스 검색 문서 예제

XML 웹 서비스 검색은 특정 웹 서비스가 존재하는지 알아내고, 그 웹 서비스에 대한 설명이 들어있는 문서의 위치를 찾는 과정이다. XML로 작성되어 웹 서비스에 대한 설명의 위치가 들어있는 .disco 파일에 저장되어 있는 검색 문서를 읽어오면 검색과정이 쉬워진다. .disco는 마이크로소프트에서 지원하는 검색 메커니즘으로서, 인터넷을 사용하고 프로그램으로 작성할 수 있다. <그림 3-7>은 웹 서비스 중 하나에 대한 검색 문서 예이다.

3.3.5 XML 웹 서비스 기술

XML 웹 서비스의 공식적인 기술은 웹 서비스 기술 언어(WSDL; Web Services Description Language)을 사용하여 XML로 작성되고, WSDL 파일에 저장된다. 비쥬얼 스튜디오 닷넷을 사용하여 XML 웹 서비스를 작성하고 있다면 자동으로 WSDL 파일을 작성해준다. 이 파일은 XML 웹 서비스에서 제공되는 함수의 프로그래밍 인터페이스를 정의한다. 그와 함께 XML 웹 서비스에서 다루는 메시지의 포맷도 정의되어 있다. 인터넷 익스플로러에서 웹 서비스에 대한 WSDL을 보려면 앞에서 나열된 검색 파일의 <contractRef> 태그에 들어있는 URL을 클릭하면 된다.

3.3.6 XML 웹 서비스 포맷

이론적으로는 어느 프로토콜로도 XML 웹 서비스를 만들 수 있지만, 실제로는 공통적으로 사용하는 오픈 프로토콜(HTTP-GET, HTTP-POST, SOAP)에 맞추는 것이 더 쉽다. 비쥬얼 스튜디오 닷넷에서 XML 웹 서비스를 작성한다면 이들 프로토콜은 자동으로 지원된다. 따라

서 본 논문에서는 원격지 데이터의 웹 서비스를 위해 비쥬얼 스튜디오 닷넷을 사용하였다.

HTTP는 표준 웹 프로토콜 중 하나이고, GET과 POST는 요청의 헤더에 인코딩 파라미터와 전달 파라미터에 사용한다. GET은 파라미터를 웹 페이지 UR 뒤에 이름과 값을 쌍으로 파라미터를 전달한다. GET 파라미터는 요청의 구조와 데이터를 혼돈하지 않고, 텍스트가 아닌 문자를 전달할 수 있도록 URL로 인코딩된다. POST도 역시 이름과 값이 쌍으로 전달되지만 POST는 실제 HTTP 요청 메시지에 이 파라미터를 숨긴다.

비쥬얼 스튜디오 닷넷으로 작성된 XML 웹 서비스는 HTTP-GET과 HTTP-POST를 모두 지원한다.

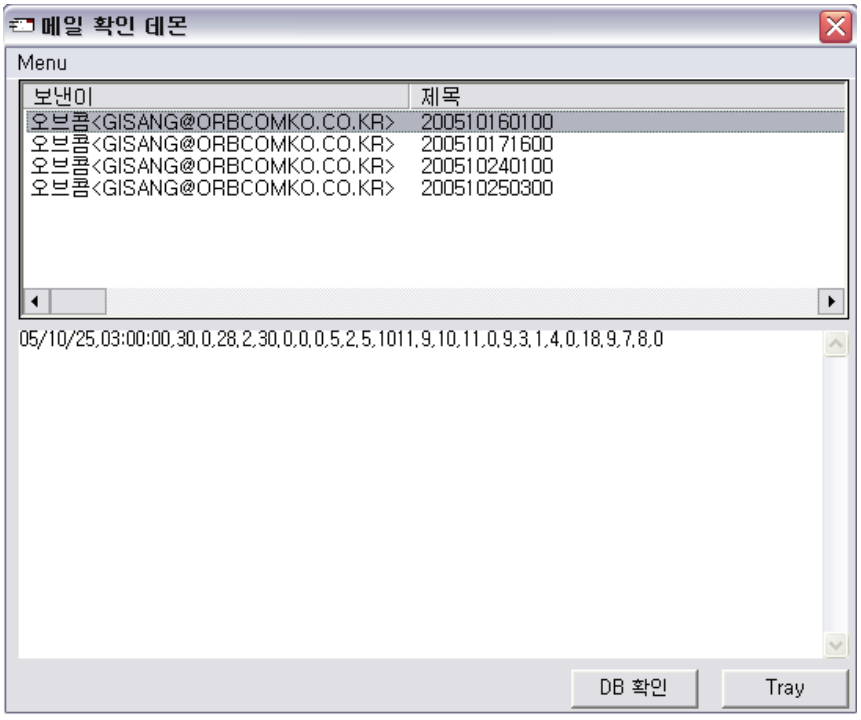
SOAP 스펙은 인코딩 규칙과 메시지 교환 패턴, 전송 프로토콜을 기술하기 위해 XML을 어떻게 사용하는지, 데이터를 표현하기 위해 XML을 어떻게 사용하는지 정의한다. 그 결과, 웹에서 HTTP를 통해 좀 더 복잡한 데이터를 전송할 수 있게 되었다. 그리고 클라이언트 응용프로그램에서 데이터 속성 및 메서드를 가진 객체로 표현할 수 있게 되었다. WSDL 파일 목록으로 다시 돌아가 살펴보면 SOAP 프로토콜을 위한 태그(<message>, <portype>, <binding>)들을 볼 수 있다.

SOAP 스펙 자체는 매우 유연하고 여러 가지 방식으로 구현될 수 있다. 언제라도 SOAP 메시지의 구체적인 내용은 비쥬얼 스튜디오 닷넷의 프록시 객체로 추상화 된다. 닷넷 컴팩트 프레임워크는 SOAP를 표현하는 클래스를 제공하고, 비쥬얼 스튜디오 닷넷은 HTTP를 통해 통신하기 위해 SOAP를 사용할 수 있도록 XML 웹 서비스를 켜두는 코드를 생성해주는 툴을 제공한다.

제 4 장 원격지 데이터 수집 시스템 구현

4.1 메일 확인 데몬

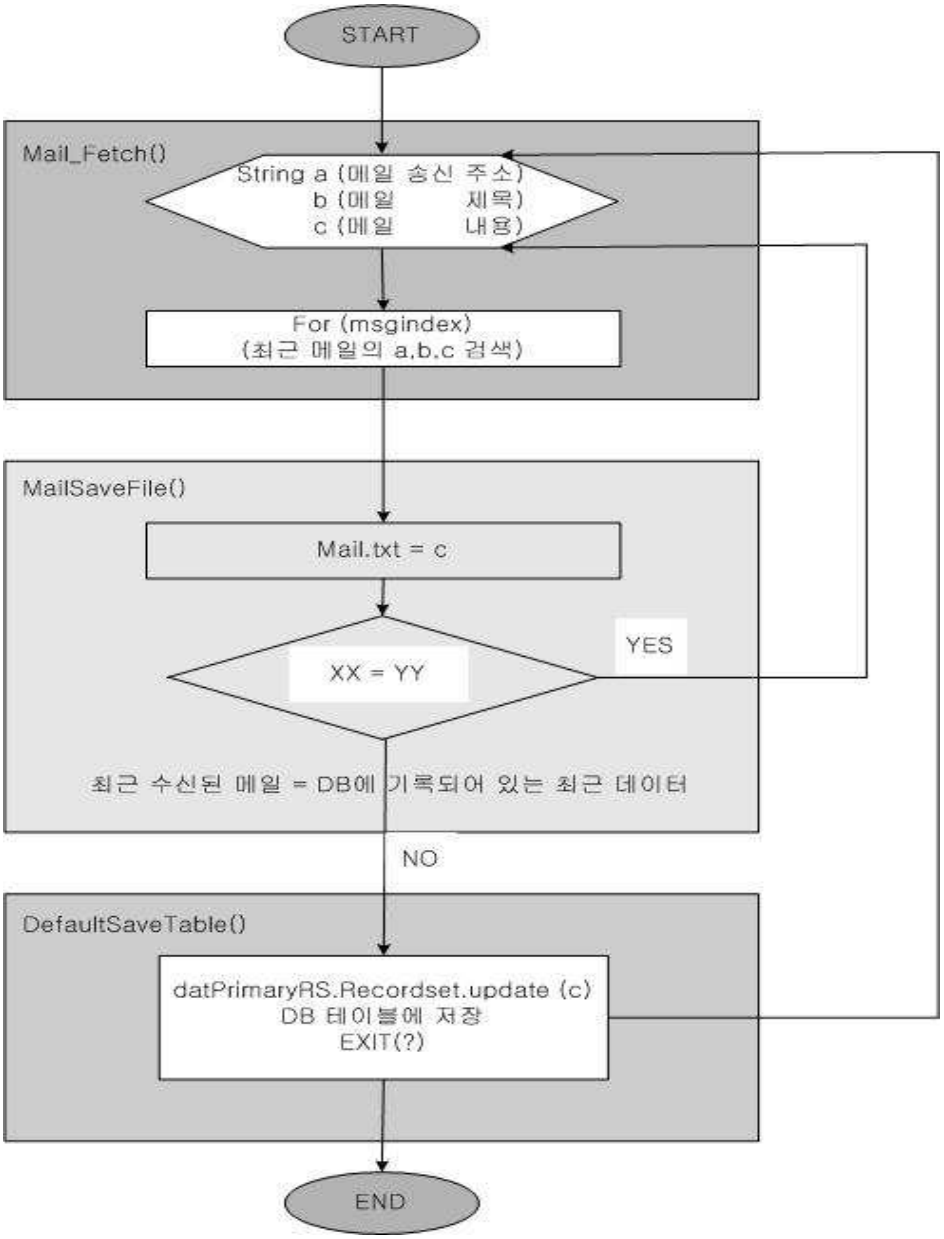
아래의 <그림 4-1>은 원격지로 부터 수집된 데이터를 오브컴 통신을 통해 수신한 메일이 있는지를 항상 확인하고, 새롭게 수신된 메일이 있다면 DB 테이블에 저장하도록 구동되는 메일 확인 데몬 응용프로그램을 구현한 것이다.



<그림 4-1> 메일 확인 데몬 구현화면

그러면 위의 <그림 4-1>에서 보여주고 있는 메일 확인 데몬의 구동방

법과 구성을 설명 하도록 하겠다.



<그림 4-2> 메일 확인 데몬 흐름도

<그림 4-2>는 메일 확인 데몬의 동작순서 흐름도로 나타내었다.

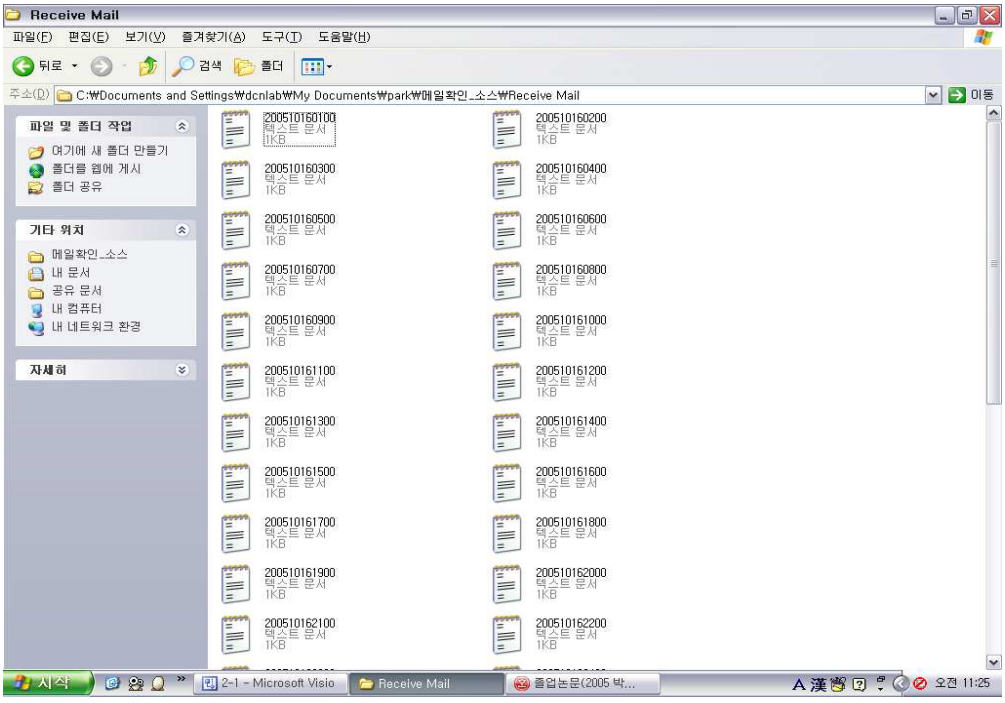
```
Sub Mail_Fetch()
Dim listv As ListItem
list.ListItems.Clear
Dim a As String
Dim b As String
Dim c As String
MAPIMessages1.Fetch
MAPISession1.DownLoadMail = True

Dim i As Integer
msgindex = Int(MAPIMessages1.MsgCount) - 1
For i = 0 To msgindex
     MAPIMessages1.msgindex = i
   a = MAPIMessages1.MsgOrigDisplayName
 & "<" & MAPIMessages1.MsgOrigAddress & ">"
   b = MAPIMessages1.MsgSubject
   c = MAPIMessages1.MsgNoteText
'MAPIMessages1.Show
Set listv = list.ListItems.Add(, , a)
listv.SubItems(1) = b
listv.SubItems(2) = c
Next i
End Sub
```

<그림 4-3> Mail_Fetch 함수

흐름도를 보면 알 수 있듯이 크게 세 개의 함수로 구성 하였으며 Mail_Fetch, MailSaveFile, DefaultSaveTable라는 함수명을 가지고 있다.

처음의 Mail_Fetch에서는 우선 스트링형 변수 a,b,c를 두었고 이것에는 각각 메일을 보내온 메일 송신 주소, 메일 제목, 메일 내용을 저장하였다. 들어온 메시지의 개수를 파악하여 최신메일의 정보를 항상 확인하도록 하는 함수부이다. 최신 정보를 확인한 뒤 Mail_Fetch함수는 MailSaveFile함수를 호출하고 이 함수에서는 취득한 정보를 mail.txt라는 텍스트 파일에 기록해두도록 하였다. 하지만 실제로 텍스파일명은 원격지에서 센서가 센싱한 시간을 파일명으로 하도록 하였다. 이것은 <그림 4-4>에서 보여주는 바와 같다.



<그림 4-4> 수신된 메일을 텍스트 파일로 저장한 화면

텍스트 파일에 정보를 기록한 후, MailSaveFile함수는 수신된 메일이 새로 수신된 메일인지를 판별하고 만약 새로 수신된 메시지가 아니라면 이 함수는 다시 Mail_Fetch함수를 호출한다. 하지만 새로운 메일이 수신되었다면, MailSaveFile함수는 DefaultSaveTable함수를 호출하게 된다.

```
Private Sub MailSaveFile()

  '원문 (파일로 저장하기)
  Form1.Text1.Text = list.ListItems(msgindex + 1).
  SubItems(2)
  Open App.Path + "\Receive Mail\" + MAPI
  Messages1.MsgSubject + ".txt" For Output As #1
     'Print #1, Format(Now, "yyyymmdd-hhmmss")
     Print #1, Form1.Text1.Text
   Close #1

   Call DeafultSaveTable

End Sub
```

<그림 4-5> MailSaveFile 함수

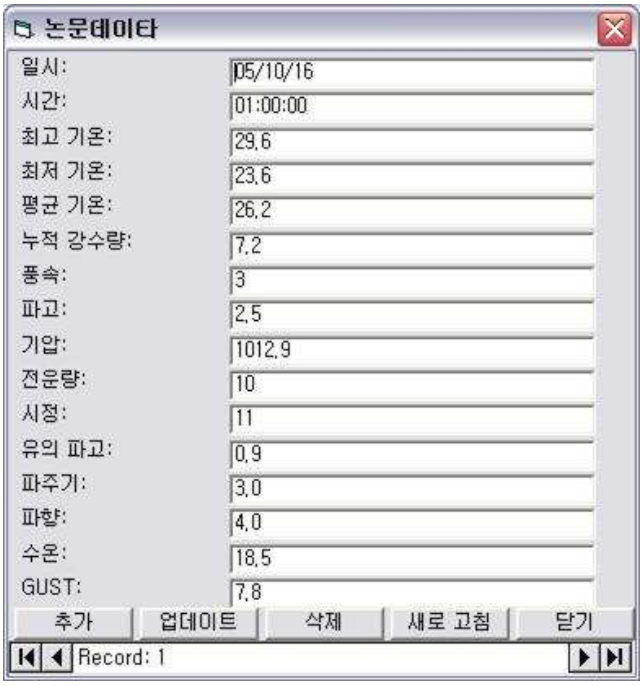
DefaultSaveTable함수에서는 새로 수신된 메일 내용을 DB 테이블에 추가 저장하고, 다시 Mail_Fetch함수를 호출하여 오브컴 메일을 계속 확인하게 된다.

```
Sub DeafultSaveTable()
Dim ss As String '디비 파일명 변수
....
  Open App.Path + "\Receive Mail\" + MAPI
  Messages1.MsgSubject + ".txt" For Input As #1
         SrcStr = MAPIMessages1.MsgNoteText 'txt파일 내용
          구분자 = ","
          For i = LBound(Split(SrcStr, 구분자))
 To UBound(Split(SrcStr, 구분자))
          Word(i) = Split(SrcStr, 구분자)(i)
          Next i
  Close #1
   ....DB Open
...
If xx <> yy Then
frm논문데이타1.datPrimaryRS.Recordset.AddNew
frm논문데이타1.datPrimaryRS.Recordset.Fields("일시") =
Word(0)
 ........
frm논문데이타1.datPrimaryRS.Recordset.Update
frm논문데이타1.datPrimaryRS.Recordset.MoveLast
End If
End Sub
```

<그림 4-6> DefaultSaveTable 함수

4.2 DB 뷰어

아래의 <그림 4-7>은 DB테이블에 저장되어 있는 데이터를 불러와서 수정 및 추가 삭제를 하거나 데이터를 확인할 수 있는 DB 뷰어 이다. 데이터 컨트롤 속성을 사용하였으며, 데이터베이스가 데이터 컨트롤의 속성 중 데이터베이스와 연결되어 있고, 이 데이터베이스에 있는 테이블은 데이터 컨트롤의 속성 중 레코드 소서에 연결되어 있다.

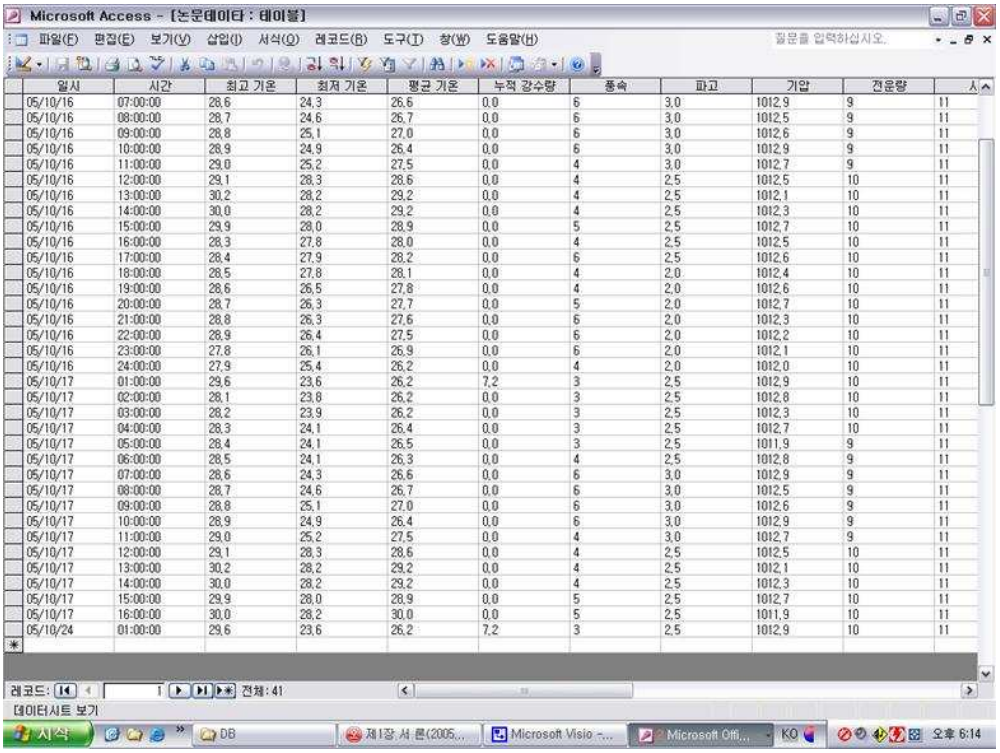


<그림 4-7> DB 뷰어

그리고 텍스트 박스의 속성 중 데이터 소스가 데이터 컨트롤에 연결되어 있으며, 텍스트 박스의 데이터 필드는 연결된 데이터 컨트롤의 레코드 소서를 통해 연결된 데이터베이스의 한 필드를 가리키게 된다. 즉 다시

말하면 데이터 컨트롤은 데이터베이스와 그 안의 테이블을 연결하고, 텍스트 박스의 속성 중 데이터 소스에 연결된다. 텍스트 박스의 속성 중 데이터 필드는 데이터 컨트롤의 정보를 가지고 와서 그 테이블 내에 있는 필드와 연결되는 것이다.

아래의 <그림 4-8>은 오브컴으로부터 수신된 메일의 데이터를 축출하여 DB 테이블에 저장되어 있는 화면이다.



<그림 4-8> DB 테이블

4.3 웹 서비스

인터넷 서비스 중에 Web Service는 Web으로 제공 되어지는 서비스이

다. 즉, 응용프로그램 개발의 최종단계로 어떠한 플랫폼에서도 개발자가 작업할 수 있도록 해주는 개념이다.

이 웹 서비스는 원격으로 프로그램을 사용할 수 있도록 하는 것이다. DCOM, CORBA, RMI등이 비슷한 기능을 수행한다. 하지만 웹 서비스는 그것과는 조금 차이가 있다.

HTTP와 XML을 이용해서 독립적인 표준을 사용한다. 그러니까, 이 웹 서비스를 사용하는 클라이언트에서는 정보를 제공해주는 서버가 어떤 운영체제와 언어를 사용하는지 알 필요가 없다. 하지만 기존의 DCOM은 JAVA에서 호출하지 못하고 RMI는 COM을 불러올 수 없다.

웹 서비스는 SOAP라는 것을 통해 XML로 전송된다. 이 프로토콜은 원거리 장비에 접속할 때 제공되는 표준이다. 이 SOAP의 장점은 HTTP로 보내진다는 것이다. 즉, HTTP에서 XML로 전송되어진다는 말이다.

하지만 웹 서비스는 최종 사용자를 위한 것은 아니다. 프로그래머를 위한 기술이라고도 할 수 있다. 어떤 하나의 프로젝트를 하면 그 프로젝트에 필요한 고수준의 알고리즘을 직접 개발할 필요가 없다. 공개된 웹 서비스에서 가져오면 된다.

공개하는 곳에서는 힘들게 여러 언어로 포팅(Porting)할 필요도 없다. 자기가 원하는 언어로 만들어서 웹 서비스의 표준만 따라서 배포하면 된다. 그리고 사용하는 곳에선 자신의 환경에 맞게 형변환을 할 필요가 없다. 그냥 SOAP에 맞게 요청해서 결과만 받아오면 되는 것이다.

4.3.1 클라이언트

XML 웹 서비스는 닷넷 프레임워크에서 ASP 닷넷으로 개발되고, 웹 서버에 있는 IIS 응용프로그램에서 실행된다. 비쥬얼 스튜디오 닷넷은 웹

서비스를 위한 WSDL 파일과 .disco 파일을 작성해준다. 웹 서비스에 대한 클라이언트 응용프로그램을 작성하려면 3장 시스템 설계 웹 서비스 구조의 인프라를 반영하는 각 단계를 통과해야 한다. 이들 단계를 설명 하자면,

가. 디렉터리

사용하고자 하는 웹 서비스를 찾아내야 할 것이다. 이 단계에서는 검색 문서로의 링크를 되돌려 줄 것이다. 이 과정은 프로그램으로 작성할 수 있다. 예를 들어서 디렉터리 웹 사이트에서 제공하는 툴을 사용한다. 하지만 직접 개발하는 웹 서비스를 찾으려면 기술 문서가 어디에서 찾을 수 있는지 이미 알고 있기 때문에 이 단계를 건너뛰어도 된다.

나. 검색

검색 문서를 사용하여 웹 서비스의 기술 문서 링크를 찾을 수 있다. 직접 개발하는 웹 서비스를 찾으려면 기술 문서가 어디에서 찾을 수 있는지 이미 알고 있기 때문에 이 단계를 건너뛰어도 된다. 이 단계에서는 WSDL로 작성된 기술 문서로의 링크를 되돌려 준다.

다. 기술

WSDL 문자 자체를 얻어오기 위해서 웹 참조라고 하는 WSDL 문서 링크를 사용한다.

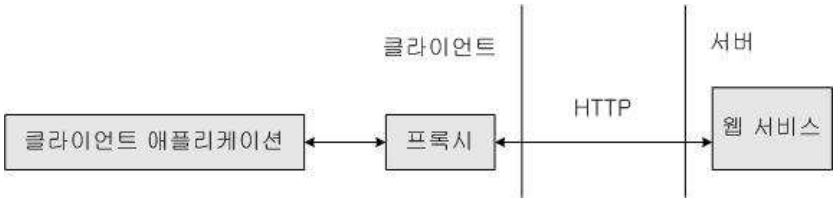
라. 개발

웹 서비스 프록시 클래스를 만들어주는 툴에게 WSDL 문서의 정보를 전달한다. 프록시 클래스는 클라이언트에서 웹 서비스를 나타낸다. 그리고

클라이언트 응용프로그램을 개발하여 만들 수 있게 한다. 이 단계에서는 XML 웹 서비스 클라이언트 응용프로그램을 되돌려 준다.

마. 런타임

클라이언트 응용프로그램은 XML 웹 서비스의 메서드를 호출하고, 서비스의 응답을 되돌려 준다.



<그림 4-9> 웹 서비스 프록시

4.3.2 서버

구현할 툴은 닷넷이고, 언어는 C#을 사용하였다. 물론 웹서버는 IIS이다.

앞서, 웹 서비스가 어떠한 것인지 확인하였다. 일단 필요한 것은, 개발툴이 닷넷이기 때문에 닷넷 프레임워크가 필요하다. 그리고 닷넷에서 웹서비스를 사용하기 위해서 닷넷 프레임워크 SDK가 필요하다. 먼저, 컴파일이 가능한 환경과, ASP 닷넷을 구동할 수 있는 환경으로 만든다. 닷넷 프레임워크 SDK 도구를 사용할 수 있게 환경변수 등록도 해주도록 한다. 웹 서비스는 SOAP를 통해서 통신이 이루어진다. 먼저, 노트패드를 열어준다. 그리고 전달 메시지 코딩을 한다. 웹 서비스를 만들기 위해서는 웹서비스임을 선언 해줘야 한다. 그리고 클래스를 선언하는데, 필수항목이

다. 적당한 사용자 정의 클래스 명을 선언한다.

다음으로, 웹 서비스의 네임스페이스를 설정하는데, [WebService (Namespace="http://localhost")] 의 형태로 설정한다. 네임스페이스는 웹 서비스들을 식별할 때 사용되는 고유한 이름이다. 이것을 통해서 클라이언트가 Web/Web Service를 구분할 수 있다.

다음은 System.WebServices.WebService 를 상속한 클래스를 만든다. 앞에서 선언부의 클래스명과 동일하다. 그 다음에 웹 서비스를 구현하면 된다.

```
public class Hello : WebService
{
..
}
```

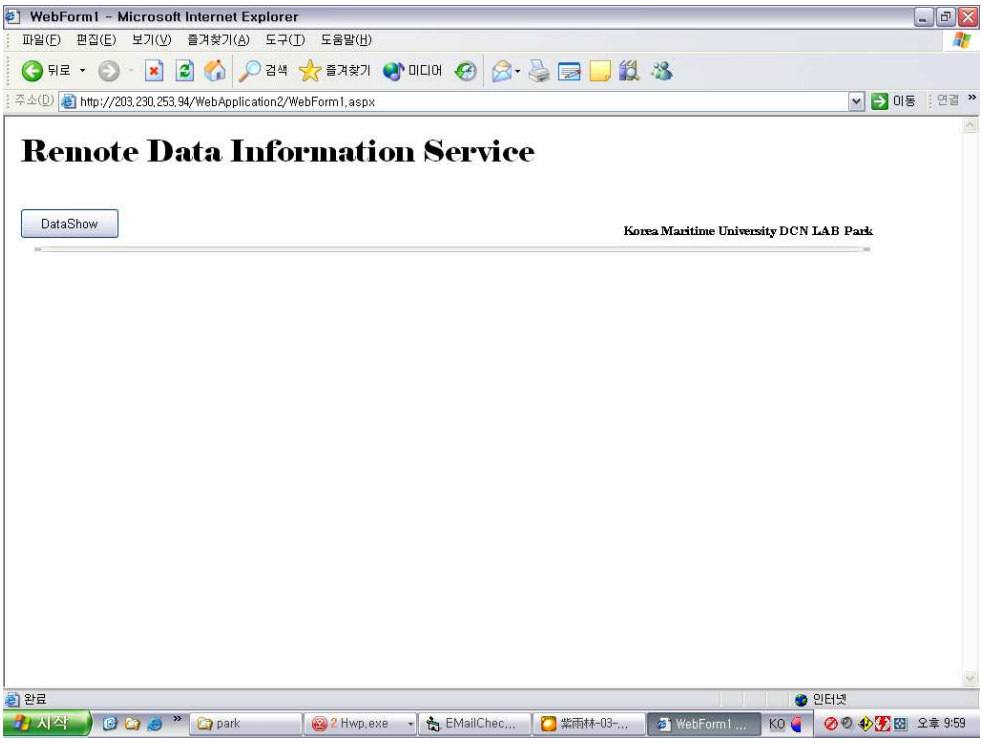
<그림 4-10> 웹 서비스의 네임스페이스 설정

그리고 클래스 내에 메소드 선언시 웹 서비스임을 알리는 프로버티 값을 준다.

[WebService]

그리고 메소드를 만들어주고 웹 서버에서 불러본다.

구축된 웹 서비스를 응용프로그램에서 어떻게 사용하는지에 관하여 설명 하겠다. 닷넷에서 사용법은 간단하다. 우리는 웹 서비스를 제공해주는 사이트를 알고 있다. 이것을 UDDI에서 찾았건 홈페이지에서 찾았건 상관없다.



<그림 4-11> 웹 서버에서 호출한 메소드 화면

http://203.230.253.94/WebApplication2/WebForm1.aspx는 앞에서 만든 웹 서비스의 주소이다. 웹서비스를 닷넷에서 사용할 문법파일로 만든다. 그리고 만든 문법파일을 컨트롤로 만든다.

마지막으로 컨트롤을 사용한다. 먼저, 콘솔을 실행하고, 웹 서비스를 문법파일로 만들기 위해서는 WSDL을 사용한다. 사용법은 WSDL [웹 서비스 URI] /language:[cs|vb|js] /out:[파일명]의 형식으로 D:>wsdl http://203.230.253.94/WebApplication2/WebForm1.aspx과 같이 작성한다.

Microsoft(R) Web Services Description Language 유틸티리
[Microsoft (R) .NET Framework]
Copyright (C) Microsoft Corporation 1998-2001. All rights reserved.

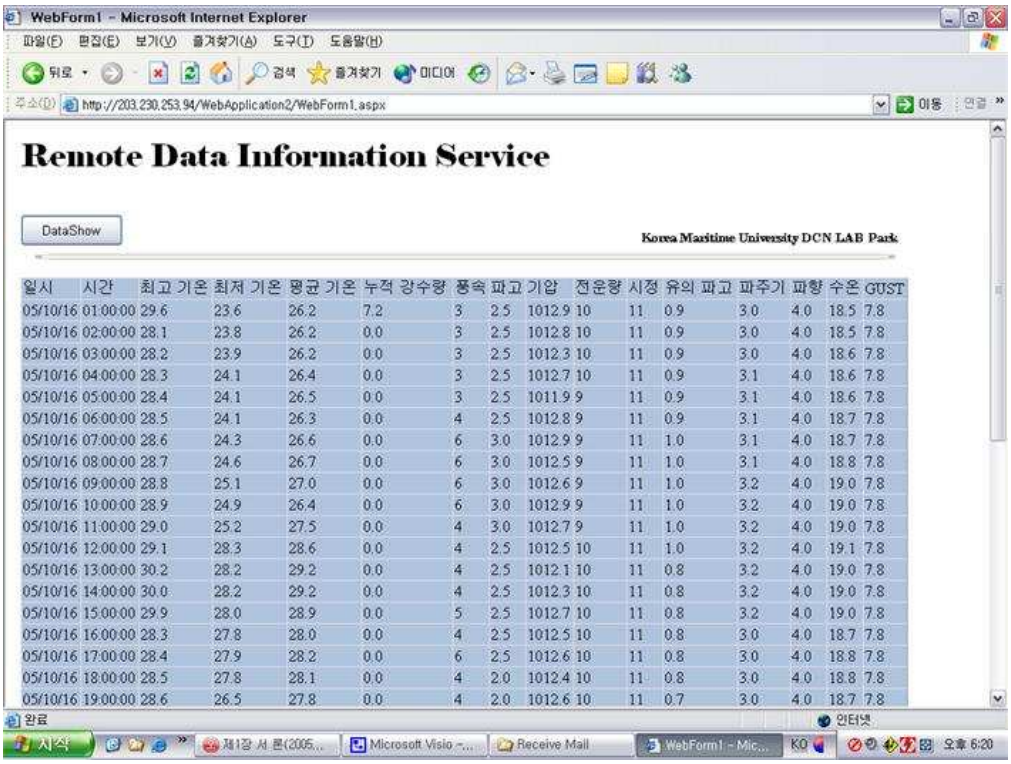
아래의 <그림 4-12>는 웹 서비스를 닷넷에서 사용한 문법 파일이다.

```
using System.Diagnostics;
using System.Xml.Serialization;
using System;
using System.Web.Services.Protocols;
using System.ComponentModel;
using System.Web.Services;
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.ComponentModel.DesignerCategoryAttribute
("code")] ....
```

<그림 4-12> 웹 서비스를 닷넷에서 사용한 문법 파일

웹 서비스는 함수로 사용할 수 있는 형태로 만들어져 있으며 동기식과 비동기식 모두를 사용할 수 있다. 그리고 이 파일을 함수로 컴파일을 하고, 응용프로그램에서 사용하기 위해 DB 테이블과 연결하여 볼 수 있게 해주는 부분을 코딩한다. 그러면 아래의 <그림 4-13>과 같은 결과 화면을 볼 수 있다.

본 논문에서는 오브컴 데이터 웹 서비스를 할 때 SOAP를 사용하였다. SOAP는 인터넷에서 원격 객체를 접속하기 위해 고안된 프로토콜이며, SOAP는 HTTP와 XML을 사용하기 때문에 개발도구나 플랫폼에 구애를 받지 않고서 SOAP 서버 혹은 클라이언트를 보다 쉽게 개발할 수 있다. 이처럼 COM 컴포넌트, 유틸리티 등으로 구성된 SOAP 툴킷도 제공되고 있다.



<그림 4-13> 웹 서비스를 결과 화면

또한, HTTP와 XML이 갖는 장점을 모두 포함하면서 컴포넌트의 상호 운용성을 높일 수 있다.

SOAP는 컴포넌트를 활성화하는 방법이나 호출하는 방법에 대해 전혀 관여하지 않으며, 이에 대한 상세한 사항은 HTTP 요구를 수신하는 수신자에게 위임하고 있다.

제 5 장 결 론

오브컴의 가장 큰 장점은 35개 저궤도 위성이 지구의 모든 지역을 커버하기 때문에 오지 및 바다 한 가운데서도 자신의 위치 파악뿐만 아니라 제 3자에게 자신의 위치 전송, 데이터 송수신이 가능하다는 데 있다.

또한 오브컴만의 VHF 대역을 사용하고 있어 전파 손실이 없고 전송이 빠르며 데이터를 패킷단위로 압축해서 전송서비스하고 있어 더욱 빠른 통신 서비스를 받을 수 있다. 이러한 통신으로 고객들의 통신이용료를 절약하는 부가효과와 다양한 기능의 단말기를 저가로 보급하는 등 고객의 사용 목적 및 빈도에 따른 다양한 형태의 가격패키지 적용으로 타 위성통신 수단 대비 저렴한 통신사용료로 오브컴의 다양한 서비스를 제공받을 수 있다.

일반적으로 메일 서비스가 이루어질 때 사용하는 다른 응용 계층 프로토콜의 경우 그들만의 약정된 TCP 또는 UDP 포트를 사용하기 때문에 인터넷상에 설치되어있는 방화벽에 많은 제약을 받게 된다.

본 논문에서는 여러 장점을 지닌 오브컴 웹 서비스 제공 시 일반적으로 사용하는 응용 계층 프로토콜이 아닌 SOAP 프로토콜을 사용하였다. 웹 서비스 시 HTTP와 같은 80번 포트를 사용하는 SOAP를 사용하므로 방화벽의 제약 없이 클라이언트 측에서 서버 응용프로그램을 실행할 수 있다.

본 논문에서 구현한 웹 서비스 응용프로그램은 특정 분산 컴포넌트 기술에 의존하지 않는 객체 액세스 프로토콜로서 XML과 HTTP를 사용하여 플랫폼에 독립적으로 서비스하는 방법 혹은 분산 객체를 액세스할 수 있다.

또한 분산 환경 하에서 정보교환을 하기 위한 하나의 프로토콜인

SOAP를 전송매체로서 HTTP를 사용하기 때문에 인터넷에서 널리 사용할 수 있다. 또한 객체지향기술이나 컴포넌트 기술을 사용하지 않는 응용프로그램일지라도 SOAP를 통해 객체서비스를 제공하거나 제공 받을 수가 있다.

참 고 문 헌

[1] KOREA ORBCOMM, http://www.orbcomm.co.kr/k_orbcomm/orbcomm.htm

[2] SOAP, http://www.w3.org/TR/soap

[3] 앤디 위즐리, 스티븐 월라이트, Microsoft .NET COMPACT FRAMEWORK, 정보문화사, pp.579 ~ 606, 2003.

[4] 윤석현, 김동준, 한상용, “웹 서비스 품질 기반 검색을 위한 UDDI 개선 모델”, 한국정보과학회 논문지, Vol. 31, No. 05, pp.511 ~ 518, 2003.

[5] EMWAC, http://cs.epnetworks.co.kr/letslearn/nt/4.html

[6] 김혜정, 한동일, 이상수, 김문규, “웹서비스 Intermediary에 관한 연구”, 한국정보처리학회, Vol. 10, No. 01, pp.1461 ~ 1464, 2003.

[7] 주경민, 박성완, 김민호, Visual Basic Programming Bible Ver. 6.x, (주)영진출판사, pp.1023 ~ 1075, 1999.

[8] 이수정, 용환승, “웹 서비스를 이용한 바이오 서열 정보 데이터베이스 및 통합 검색 시스템 개발”, 한국정보처리학회 논문지 D, Vol. 11, No. 04, pp.755 ~ 764, 2004.

[9] SOAP Version 1.2 Part 0: Primer, http://www.w3.org/TR/2003/REC-soap12-part0-20030624

[10] Latest version, http://www.w3.org/TR/soap12-part0

[11] Previous version, http://www.w3.org/TR/2003/PR-soap12-part0-20030507

[12] 홍의경, 데이터베이스 배움터, 생능출판사, pp.273 ~ 303, 2003.

[13] 김신강, 임효상, 이정훈, 한욱신, 황규영, “SOAP 기반 응용프로그램 디스패쳐의 설계 및 구현”, 한국정보과학회, Vol. 30, No. 01, pp.459 ~ 465, 2003.

[14] Eric Brierley, Paul Sanna, Anthony Prince, Visual Basic 5 HOW-TO, 영진출판사, pp.621 ~ 733, 1998.

[15] 권오경, 한태숙, “nML 프로그래밍 환경에서 SOAP 클라이언트 구현”, 한국정보과학회 논문지 B, Vol. 30, No. 02, pp.164 ~ 172, 2003.

[16] 윤종호, TCP/IP 및 윈도우 네트워킹 프로토콜, (주)영진출판사, pp.335 ~ 361, 1999.