

# 서비스 엔진을 이용한 웹 기반 메일 에이전트 시스템의 설계 및 구현

임 양 원<sup>†</sup> · 권 기 훈<sup>††</sup> · 임 한 규<sup>†††</sup>

## 요 약

웹 기반 전자우편 시스템은 계정과 서비스를 제공하는 서버와 사용자외의 인터페이스 역할을 하는 클라이언트로 구성되며 웹 브라우저가 클라이언트의 역할을 담당한다. 즉, 일반 데일서버에 계정을 만들 수 없는 일반 사용자들이 웹 환경을 통하여 메일서비스를 제공받을 수 있도록 하는 것이나 최근 들어서는 양질의 서비스를 제공하고 있지만, 아직까지 만족할 만한 메일서비스가 제시되지 않고 있다.

본 논문에서는 웹을 기반으로 하는 메일 에이전트 시스템을 설계하였으며, 인터넷 익스플로러(Internet Explorer)를 기반으로 하는 시용지환경피, NT Server 4.0을 기반으로 하는 편리한 권리환경을 구현하였다. 메일 에이전트 시스템의 서비스 엔진을 이용하여, 부제중처리 기법과 메일 필터링(Mail Filtering), 수신거부 등을 통한 효율적인 사용자 편의성을 제공하고, 메일의 폭주와 사용자의 급증으로 인한 시스템 저하현상을 극복하는 효율적인 메일 서비스 엔진을 구현해서 시스템의 안정성을 제공하는 메일 에이전트 시스템을 개발한다.

# Design and Implementation of a Web based Mail Agent System using Service Engine

Yang-Won Lim<sup>†</sup> · Ki-Hoon Kwen<sup>††</sup> · Han-Kyu Lim<sup>†††</sup>

## ABSTRACT

A web based E-mail system consists of servers which supply mail accounts and service, and clients which take part of interfacing with users. The web browsers such as Netscape Navigator and Internet Explorer are the popular clients. Users who cannot afford to get an E-mail id in general mail servers are, therefore, able to reach an e-mailing service in the web environment by using client programs in connection with a mail server. Although the quality of service has been upgraded a lot in recent days, it doesn't provide with satisfactory services anymore.

In this paper, we intend to design a mail agent system based on the web environment. Through this system we could reach a convenient user interface based on Internet Explorer and could easily access administratory environment on the basis of NT Server 4.0. Using the service engine dominant in mail agent svstem, it could provide with convenient user interfacing tools like responding the absence of users, mail filtering and refusing messages, etc. And it would ultimately make possible a constant system security, by preventing system failure phenomena caused by extraordinary uses beyond its capability.

† 준 회 원 안동대학교 대학원 컴퓨터공학과  
†† 준 회 원 안동대학교 대학원 컴퓨터공학과  
††† 종신회원 안동대학교 전기정보산업학부 교수  
논문접수 : 1999년 12월 30일 심사완료 : 2000년 2월 3일

## 1. 서론

인터넷 기술이 발전함에 있어서 전자우편은 가장 활용성이 높은 인터넷 애플리케이션 중의 하나로서 TCP (Transmission Control Protocol)의 연결들 중의 약 반이 SMTP(Simple Mail Transfer Protocol)[1]를 위해 설정되어 있다.

그러나, 기존의 메일시스템은 메일 서비스를 받기 위해서는 적절하게 세팅된 클라이언트 프로그램이 필요하다. 이와 같은 기존의 클라이언트/서버 메일 시스템들은 독자적인 클라이언트 소프트웨어의 사용을 요구하기 때문에 제한적이고 폐쇄적이라 할 수 있다. 이러한 문제점들을 해결하기 위해 메일 서버에 계정을 만들 수 없는 일반 사용자들에게 웹 환경을 통하여 메일 서비스를 제공받을 수 있도록 하는 웹 기반의 메일 서비스가 현재 제공되고 있다.

웹 기반 메일 시스템은 어떤 클라이언트 소프트웨어보다도 보편적으로 이용되고 있는 웹 브라우저를 메일에 대한 클라이언트 소프트웨어로 이용함으로써 이디에서나 사용이 가능하며, 메일 사용자들은 웹 브라우저 이외에 어떤 새로운 클라이언트 소프트웨어도 설치할 필요가 없게 된다.

본 논문에서는 이러한 웹 기반 메일 시스템을 보다 효율적이고 다양한 기능을 제공하는 웹 기반 메일 에이전트 시스템을 SMTP를 이용해 설계하고 구현한다[12].

본 논문의 구성은 다음과 같다. 2절에서는 웹 기반 메일 시스템에 대한 기존의 연구와 개발환경에 대해 언급한다. 3절에서는 제안하는 웹 기반 메일 에이전트의 주요 기능에 대해서 설명을 하고, 4절에서는 이러한 기능의 설계 및 실제 구현에 대해 설명하며, 5절에서는 본 논문의 결론을 맺고, 본 시스템의 개선 방향과 향후 연구 계획에 대해 언급한다

## 2. 관련 연구

### 2.1 POP(Post Office Protocol)

POP는 RFC[1] 1460에 기술되어 있으며, 메일서버에서 메일을 회수하는 프로토콜이다. 기존의 메일서버는 메일을 회수하기 위해서는 메일 서버에 로그인 한 후 메일 프로그램을 사용해서 메일을 확인하였으나, 현재

모든 Mail Agent(Netscape, Outlook Express, Eudora 등)는 이 POP 및 SMTP를 사용하여 메일서버에 직접 로그인 하지 않고 메일을 확인할 수 있으며, 메일을 보낼 수 있다.

### 2.2 SMTP(Simple Mail Transfer Protocol)

인터넷 메일 시스템은 RFC 821 "Simple Mail Transfer Protocol"에 정의되어 있으며, 편지 배달 에이전트(IIS의 SMTP서버 같은 메일서버)가 다른 에이전트에 게 편지를 전달하는데 사용하는 프로토콜이다[2].

### 2.3 MIME(Multipurpose Internet Message Extensions)

이진파일 추가와 멀티미디어 지원을 추가하는 프로토콜로서 RFC 1521, 1522에 기술되어 있다. 물론, 기존의 RFC 822 메시지 포맷과 호환되고, 이진파일 전송, 메시지유형의 결정, 새로운 문자집합, 미래를 위한 성장 지원 등을 포함하고 있다[3].

### 2.4 메시지 구조(MIME)

RFC822 메시지 구조에 추가한 헤더 필드로 수신자

```
x-sender: shamaim@andong.ac.kr
x-receiver: eif2@wall.co.kr
Received: from mailx.andong.ac.kr [203.251.86.17] By
www.wall.co.kr with Microsoft SMTPSVC
(5.5.1775.675.6)
    Fri, 28 Jan 2000 11:43:04 +0900
Received: from www [1203.232.40.108] by mailx.andong.
ac.kr (8.9.1a-H/8.9.1) with SMTP id LAA24396
for <eif2@wall.co.kr>; Fri, 28 Jan 2000 11:42:39
+0900 (KST)
Message-ID:
<01401bf6939$647b2e70$6c28e8cb@wall.co.kr>
From: =?ks_c_5601-1987?B?wNO+574?=<shamaim
@andong.ac.kr>
To: <eif2@wall.co.kr>
Subject: Non-Deliveryable Mail TEST
Date: Fri, 28 Jan 2000 11:43:00 +0900
MIME-Version: 1.0
Content-Type: multipart/alternative;
    boundary="====_NextPart_000_0011_
    01BF6984.D4475F30"
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 5.00.2314.1300
X-MimeOLE: Produced By Microsoft MimeOLE V5.00.2314.
1300
Return-Path: shamaim@andong.ac.kr
This is a multi-part message in MIME format
====_NextPart_000_0011_01BF6984.D4475F30
Content-Type: text/plain;
    charset="ks_c_5601-1987"
Content-Transfer-Encoding: base64
```

(그림 1) MIME 메시지 구조

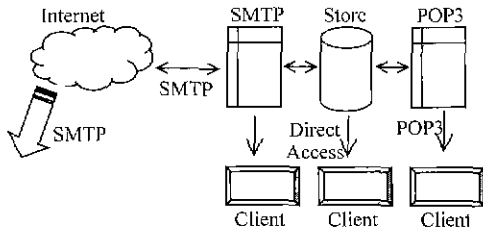
1) Request For Comment, Document 시리즈, 1969년에 게시된 Internet 프로토콜 체계에 대한 권위 실린을 기술한 것.

는 메시지가 MIME구조인지 확인할 수 있으며, MIME로 해석하게 된다

### 2.5 전자우편 시스템

전자우편 클라이언트 프로그램으로 SMTP서버에 편지를 보낸다. 전자우편 메시지 수신인의 도메인이 서버가 있는 도메인과 같으면 아스키 텍스트를 서버에 저장하고, 자기 도메인이 아닌 경우 해당 서버에 전달한다. 전자우편 메시지가 수신자의 SMTP서버에 도착하면 수신자에 할당된 사서함에 저장된다. 수신자는 아웃룩 익스프레스 같은 우편 클라이언트에서 POP3 프로토콜로 서버에 접속해서 자신에게 온 편지를 가져간다 이 과정에서 포트25를 사용하는 SMTP로 편지를 서버에게 전달하고, 포트 110을 사용하는 POP3프로토콜로 편지를 클라이언트까지 배달한다.

(그림 2)는 전자우편 배달과정을 보여준다



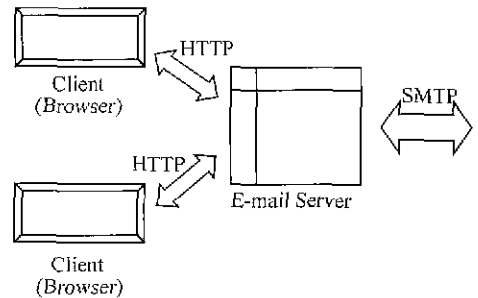
(그림 2) 전자우편의 전송

일부 시스템(주로 유닉스 플랫폼)에서는 SMTP 편지 저장 공간에 직접 접근할 수 있다. 즉, 사용자가 SMTP서버에 접속해서 SMTP와 사서함으로부터 전자우편을 직접 읽을 수 있는 것이다. 그러나, 아웃룩 익스프레스처럼 일단 사서함에서 클라이언트의 우편함으로 편지를 옮긴 후 읽는 편이 더 편리하다[4]

### 2.6 웹 기반 전자우편 시스템

웹 기반 전자우편 시스템은 계정과 서비스를 제공하는 서버와 사용자와의 인터페이스 역할을 담당한다 브라우저를 통해서 전자우편 서비스를 제공하는 웹 사이트에 접속한 사용자는 ID와 암호를 가지고 자신의 정보를 관리하게 된다 또한 사용자는 서버에 자신의 전자우편 주소를 가지게 되며 이를 이용하여 다른 사용자와 전자우편을 주고 받을 수 있다. 전자우편 클라이언트 역할을 하는 브라우저는 사용자로부터 데이터를 입력

받으며, 이는 HTTP(HyperText Transmsion Protocol) [5] 프로토콜을 이용하여 서버에 존재하는 CGI(Common Gateway Interface)프로그램에게 전달된다. CGI프로그램은 전달받은 사용자 데이터를 수정하여 실제로 전달 가능한 전자우편의 형태로 만들며 SMTP프로토콜을 이용하여 수신자에게 전달된다. (그림 3)은 웹 기반 전자우편 시스템의 동작을 보여준다



(그림 3) 웹기반 전자우편 시스템의 동작

이처럼 웹 기반 전자우편 시스템은 사용자가 특정 회사의 전자우편 클라이언트 프로그램을 구입하지 않아도 브라우저인 있는 환경이면 전자우편을 송수신할 수 있도록 함으로써 사용자에게 매우 편리한 환경을 제공한다 이러한 환경은 기업이나 학교 등의 인트라넷 환경에 적용될 수 있으며 공동의 사용자 인터페이스를 제공하므로 업무 수행 능력이나 소속감 등을 높이는 데에도 기여할 수 있다.

### 2.7 개발환경

서버의 개발환경은 기존의 서버/클라이언트 메일 시스템의 문제점을 해결하고, 사용과 관리에서의 사용자 인터페이스를 지원하기 위해 MS의 NT Server 4.0에서 구현한다. SMTP서버는 NT Server 4.0을 인스톨한 후, NT 옵션팩30의 IIS(Internet Information Server)에서[4] 지원하는 SMTP를 사용하고, 사용자 환경과, 관리환경을 구축하기 위해서 HTML2과[6] ASP3로[7] 구현하였으며, 서비스엔진의 서비스 모듈을 생성하기 위해 VB6.0을[8] 데이터베이스로는 MS-Access97을 사용한다.

2) HTML(HyperText Markup Language) HTTP를 통해 데이터 등 주고 받을 수 있도록 정의해 주는 언어  
3) ASP(Active Server Page) MS가 내놓은 서버베이스의 새로운 기술로서 디이티브한 웹페이지를 만드는 데 이용

2.8 성능비교

기존에 나와 있는 여러 가지 웹 기반 메일 시스템과 본 논문에서 제시하는 메일 시스템의 대표되는 기능들을 <표 1>과 같이 비교하여 보았다.

<표 1> 성능비교 분석표

항목	A	B	C	W
OS	Unix	Linux	Linux	NT
메일검색	O	O	O	O
핀지힐징리	X	O	O	O
메일자동분류	O	O	O	O
수신확인기능	O	X	O	O
개인편지함	X	O	O	O
지동용답	O	X	X	O
서명파일포함	O	O	O	O
수신거부	O	X	X	O
잘못된사용자에 온 메일의 처리	X	X	X	O

<A~C 기존의 메일서버, W 제안된 메일서버>

현재 나와 있는 여러 가지 웹 기반 메일 시스템은 더 다양한 기능과 특징들을 제공하고 있으나, 본 논문에서 제시하는 웹 기반 전자우편 에이전트 시스템은 서비스엔진을 이용하여 더욱 빠르고 다양한 기능들을 제공한다. 특히, 잘못된 사용자에게 온 메일의 처리는 기존의 메일시스템에서는 상당한 시간이 지나야 잘못된 메일임을 확인할 수 있는데 반해 본 시스템에서는 수 초내지 수분내에 확인할 수 있다. 이 부분은 4장 구현 부분에서 더 자세히 설명하였다

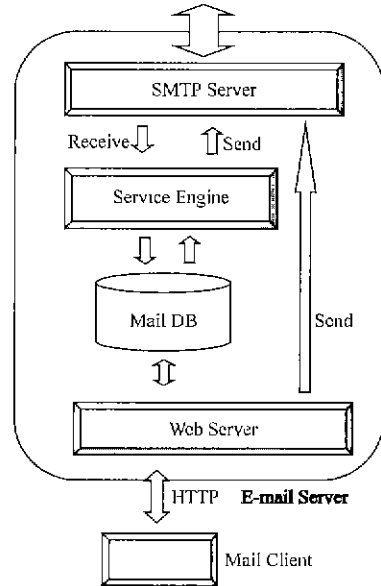
3. 주요 Agent 기능

메인 Agent의 기초적인 송신,수신을 기반으로 하여 사용자 부제시 부제증 관리, 메일필터링, 수신거부, 서명첨부, 메일쓰기의 데이터전이, 메일검색, 잘못된 사용자(등록되지 않은)의 처리 등의 기능을 구현하였다.

3.1 제안된 웹기반 전자우편 에이전트 시스템의 구조

본 논문에서 제시한 웹 기반 전자우편 에이전트 시스템의 구조는 기존의 전자우편 에이전트 시스템을 바탕으로 하였다. (그림 4)는 전자우편 서버에 계정을 가진 사용자가 메시지를 작성하여 전달하는 구조를 보여주고 있다. 사용자는 클라이언트 역할을 하는 브라우저를 통해 메일을 송신하거나 수신한 전자우편을

확인할 수 있다.



(그림 4) 제안된 웹기반 전자우편에이전트 시스템

본 논문에서 제시한 웹 기반 전자우편 에이전트 시스템의 동작원리는 다음과 같다. 사용자가 브라우저를 이용 웹기반 전자우편 서비스를 제공하는 사이트로 접속하여 서버에 도착한 메일을 확인하거나 사용자가 작성한 메일을 보낼 때, 서비스 엔진에서 이를 수행하게 된다.

사용자가 작성한 메일을 보낼 때, 메일 에이전트는 내부인지 외부인지를 확인한 후 내부이면 mail DB에 저장하고, 외부이면 SMTP를 이용해서 전달하도록 한다.

메일 에이전트는 일정한 시간이 경과하면 자신의 서버에 새롭게 배달되어온 메일이 있는지를 확인 후, 등록된 사용자에게 배달되어온 메일이면 mail DB에 새로운 메일을 저장하고 등록된 사용자가 아니라면, 메일을 발송한 발신인에게 등록된 사용자가 아님을 알리는 메일을 발송하게 된다.

수신된 메일을 확인할 경우 사용자는 서버의 자원(NT의 경우 mailroot/drop)에 직접 접근하는 것이 아니라, 전체메일이 보관되어있는 mail DB에서 읽어오게 된다.

결과적으로 등록되지 않은 사용자에게 메일이 수신되었을 경우, 그 즉시 발신인에게 알릴 수 있게 되고,

사용자는 메일을 확인할 경우 검증된 메일만을 수신할 수 있게 된다.

### 3.2 부재중 관리

사용자가 부재중 설정과 부재중 메모를 남기면, 메일 에이전트 시스템은 부재중인 사용자에게 메일이 도착했을 경우, 도착한 메일은 mail DB에 저장시키고, 발신인에게 사용자의 이름으로 부재중에 설정해 둔 메모를 전달하는 기능이다.

예를 들어 사용자가 장시간 메일을 확인하지 못할 때 부재중 설정을 할 수 있다. 이때 test@wail.co.kr로부터 메일이 배달되었다면 메일 에이전트는 사용자의 mail DB에 지장을 하고, test@wail.co.kr로 사용자의 부재중 메모를 보내게 된다.

### 3.3 메일필터링(mail filtering)

메일이 도착하면 메일 에이전트가 자동으로 지정된 작업을 수행하게 되는 기능이다.

어떤 특정인에게 오는 메일에 대해서 기록하거나 SPAM<sup>4)</sup>성 메일을 대응 할 수 있는 기능이다

### 3.4 수신거부

수신거부항목에 등록된 주소로부터 메일이 도착하면 메일 에이전트가 사용자의 mail DB에 저장하지 않고 즉시 삭제해 버리는 기능이다.

SPAM성 메일에 대해서는 처리는 비슷하지만, 메일 필터링은 사용자의 mail DB에 저장을 하고, 수신거부는 사용자의 확인을 거치지 않고 삭제해버리는 데에서 차이가 있다. 즉 사용자는 메일필터링에 등록한 SPAM 메일에 대해서 휴지통으로 비로 보내지게 할 수 있고 그 메일을 확인할 수 있지만, 수신거부로 등록한 SPAM 메일에 대해서는 사용자에게 전달되지 않는다.

### 3.5 메일분류

메일필터링은 메일이 도착하는 즉시 특정조건에 의해서 분류를 하지만, 메일분류는 현재 사용자에게 있는 메일들을 분류하는 기능이다 메일필터링처럼 지속적으로 분류를 하는 것이 아니라, 메일분류를 지정하는 순간에 메일들을 분류하게 된다

### 3.6 서명관리

사용자가 메일을 보낼 때, 메일의 마지막에 특정한

글이나 이미지를 삽입시킬 수 있도록 미리 정의할 수 있다 이러한 기능을 이용하면 메일을 보낸 사람이, 수신자가 알고 있는 사람인지를 확인 할 수 있어 실제계에서의 서명과 같은 효과를 낼 수 있다. 보안메일을 보낼 때의 방법중의 하나라고 할 수 있다.

### 3.7 메일검색

특정 편지함이나, 전체 편지함에 있는 메일을 제목이나 보낸 사람에 포함된 특정단어를 검색하는 기능이다.

### 3.8 메일쓰기의 확장기능

메일쓰기에서는 두 가지 기능을 제공한다. 받는 사람과, 제목과 내용만이 포함된 단순메일과 단순메일에 다중과일침부, HTML확장, 중요도, 전송후 저장소 선택 등을 포함하는 확장된 메일을 제공한다.

항상 단순메일의 쓰기상태가 되지만, 사용자가 확장된 기능들을 시용하려고 할 때, 편리한 사용자 환경을 제공하는 기능이다 이 기능은 단순메일과 확장된 메일간의 사용자가 쓴 메일의 내용전달이 가능하니

### 3.9 개인서함 관리기능

개인서함이란 메일시스템에서 집계놓은 편지함에 사용자의 정의하는 편지함을 말하는 것으로, 본 시스템에서는 사용자가 편지함의 이름과 설명을 정의할 수 있으며, 관리할 수 있는 사용자 환경을 지원한다.

## 4. 웹 기반 메일 에이전트 시스템의 구현

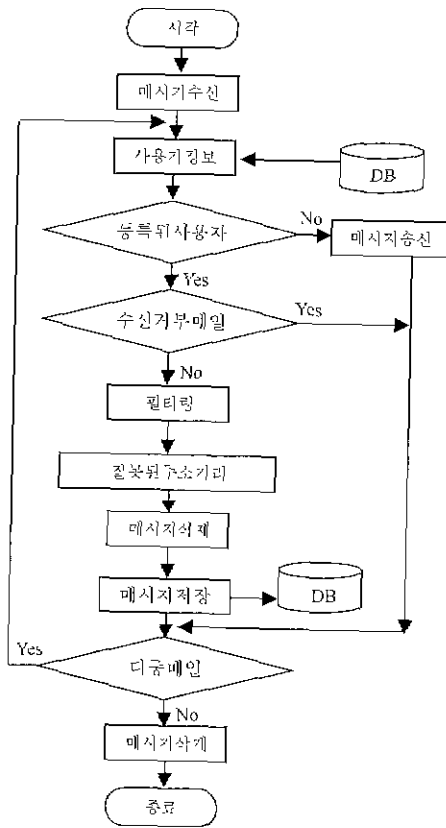
### 4.1 SMTP의 구현

본 논문에서 제안하는 웹 기반 메일 에이전트 시스템은 마이크로소프트(이하 MS)사의 NT4.0 Server의 SMTP를 기반으로 구현하였으며, 메일 에이전트의 서비스 엔진 구현을 위해 MS의 VC6.0으로 제작한 OCX<sup>5)</sup>를 이용하여 VB6.0으로 프로그래밍 하였다.

웹기반 메일 에이전트를 구현하기 위해서는 SMTP와 Mail Server사이에 서비스 엔진을 추가해서, 사용자가 직접 SMTP의 영역(NT의 경우 mailroot/drop)을 검색해서 사용자의 메일을 가지고 오는 것이 아니라, 서비스 엔진이 일정한 가격을 두고 사용자의 메일을 mail DB에 저장해야 한다. 서비스 엔진이 수신된 메일을 받을 때의 절차는 다음과 같다

4) SPAM 사용자가 원하지 않는 광고성 메일

5) OCX(OLE Control Extensions) DLL(Dynamic DataLink)구조의 함수 포인티 테이블



(그림 5) 서비스엔진의 처리 절차

1. 먼저, SMTP영역에서 파일로 된 메시지의 파일명을 읽어온다
2. 각각의 메시지에 대하여, 수신자별로 읽어온다  
`Set objSession = New CDONTS.Session`  
`objSession.LogonSMTP UID, EMail(UserCode)`
3. 만일, 등록되지 않는 사용자에게 온 메일이라면, 송신자에게 잘못된 메일임을 알리는 메시지를 보낸다.
4. 수신거부에 대한 처리, 메일필터링에 대한 처리, 잘못된 주소로부터 온 메시지의 처리, 부제중처리를 한다.
5. 수신된 메시지를 SMTP영역에서 메일 에이전트 시스템의 Mail DB에 저장한다.
6. 다중메일처리를 한 후 SMTP영역의 메시지를 삭제하고 서비스엔진은 종료한다

외부로부터 수신된 메시지는 SMTP의 특정영역에 다음과 같은 파일이름으로 저장된다

000d67020121ab9NT\_SERVER.eml

다중메일이란 송신자가 동일한 서버에 있는 여러 명의 사용자에게 보내는 메일을 말하는 것으로 메시지 구조에서, 수신인이 여러 명인 경우에 다음과 같은 메시지구조를 가지게 된다.

```
x-receiver joker@wail.co.kr
x-receiver' won@wail.co.kr
x-receiver' test@wail.co.kr
```

NT Server-10의 SMTP서버에서는 각각의 수신된 메시지를 사용자의 정보로 읽어온다 (그림 5)는 메시지가 도착한 후 서비스엔진에서 처리하는 과정을 나타낸 것이다

수신된 메시지를 수신되는 사용자에 대해서 처리한 후, 서비스 엔진은 SMTP의 메일이 저장되는 영역에서 수신된 메시지를 삭제해야 한다

#### 4.2 DB의 구현

본 논문에서 제안하는 웹 기반 메일 에이전트 시스템의 DB는 MS의 ACCESS 97로 구현하였으며, 보안처리 및 관리자 편의성을 위해 하나의 DB에 모든 정보를 포함시키고, DB의 접근은 VB6.0으로 구현한 서비스엔진과 DLL<sup>6)</sup>에서만 접근이 가능하도록 구현하여, 보안의 효율성을 한단계 증가시켰다

데이터베이스는 크게 사용자 테이블과 메일테이블로 나눌 수 있으며, 그 외 부가적인 테이블로 이루어져 있다.

모든 테이블은 동적으로 연결되어 있으며, DB접근은 사용자의 ID로 접근하는 방식이 아니라, 랜덤하게 만들어지는 사용자 고유 Code로 접근이 가능하게 하였다.

메일테이블에서는 기본적인 편지함에 사용자 편지함을 처리하기 위해 편지함이나 다른 테이블로 접근하는 방식이 아닌, 각각의 편지함을 Code화하여 하나의 테이블에서 관리하도록 구현하였다. 그 밖에 모든 테이블은 DB의 관계성을 극대화하여 관리의 편리함과 DB의 유용성을 제공하였다.

#### 4.3 메일읽기의 처리

사용자가 메일을 읽는 부분의 알고리즘은 같지만, 수행 성격상 두 가지로 구분지을 수 있다. NT 서비스 엔

6) DLL(Dynamic Linked Library) : 여러 개의 애플리케이션에서 동적으로 연결하여 사용될 수 있는 라이브러리.

진에서 처리하는 것과, DLL화 된 것으로 나눌 수 있다. 메일읽기는 NT에서 지원하는 CDONTS의 Session 객체를 써서 구현하였다.

먼저, 서비스엔진의 경우 처리는 다음과 같다.

1. NT 서버의 시작됨과 동시에 서비스 엔진도 시작된다.
2. 서비스엔진이 시작되는 시점에 SMTP영역에서 새로운 메시지를 확인한다.
3. 각각의 메시지에 대하여 수신되는 사용자별로 DB에 저장한다.
4. 미리셋팅된 일정한 시간이 지나면 다시, SMTP영역에서 메시지를 확인한다.
5. 위와 같은 과정을 계속해서 반복한다.

다음은 DLL로 처리되는 과정이다.

1. 사용자가 로그인 되는 순간 수행되거나,
2. 로그인된 사용자가 "메일읽기"를 클릭시, 수행하게 된다.

#### 4.4 메일쓰기의 처리

메일쓰기는 CDONTS의 NewMail객체를 사용해서 다음과 같이 구현하였다(기본메일).

```

Set NewMail = CreateObject("CDONTS NewMail")
IF Request("HTMLMODE") = "Y" THEN
    NewMail.MailFormat = 0
    NewMail.BodyFormat = 0 'CdoBodyFormatHTML
Else
    NewMail.MailFormat = 1
    NewMail.BodyFormat = 1 'CdoBodyFormatText
end if
NewMail.To = Temp(UserIndex) & ""
NewMail.From = Email
NewMail.Importance = Request("Important")
NewMail.Subject = Request("Subject") & ""
NewMail.Body = Request("Content") & vbCrLf & strSign
NewMail.Send
Set NewMail = Nothing
    
```

(그림 6) 메일쓰기 알고리즘

1. 메일쓰기 폼에서 사용자가 입력한 내용을 넘겨받는다.
2. MailFormat과 BodyFormat의 값을 결정한다.
3. 수신자와 발신자의 주소를 지정한다.
4. 기타 사용자가 입력한 중요도와 제목, 내용등을 넘겨 받는다.
5. CDONTS의 NewMail을 이용해서 보낸다.

#### 4.5 잘못된 사용자에게 온 메일의 처리

잘못된(등록되지 않은) 사용자에게 온 메일을 처리하는 부분으로 처리하는 과정은 다음과 같다

1. 메일로드시 메시지에서 수신자정보를 얻어온다.
2. 사용자DB의 사용자 정보와 수신자정보를 비교한다.
3. 사용자DB에 없을 경우, 잘못된 사용자임을 알리는 메시지를 발신자에게 보낸다.

```

GetUserFound(ReceivedUser)
If GetUserFound = UserNotFound Then
    Mail.From = "Webmaster" & Domain
    Mail.To = SendID
    Mail.Subject = "사용자 [" & UID & Domain & "]을 찾을 수 없습니다."
    Mail.Body = "잘못된 사용자입니다."
    Mail.Send
End if
    
```

(그림 7) 잘못된 사용자에게 수신된 메일 처리 알고리즘

보내는 사람의 오타로 등록되지 않은 사용자에게 메일을 보냈을 경우, 기존의 메일시스템은 바로 돌려보내지 못하고 계속해서 수신되는 도메인으로 접속을 시도하다가, 일정한 시간이 흐르면, 다음과 같은 메시지를 발신자에게 전송한다.

제목 : Nondeliverable mail  
 내용 :  
 -----Transcript of session follows -----  
 feifej@andong.ac.kr  
 550 <feifej@andong.ac.kr>... User unknown

하지만, 본 시스템에서는 등록되지 않은 사용자에게 메일을 보냈을 경우, 수분내로 체크가 가능하기 때문에, 발신자는 자신이 보낸 메일이 잘못된 사용자에게 보내어졌음을 알게 된다.

제목 : 사용자 [dfdsjf@mail.co.kr]을 찾을 수 없습니다.  
 내용 :  
 잘못된 사용자입니다.

<표 2>는 기존의 Email시스템과 본 시스템의 잘못된 사용자에게 대한 처리를 시간별로 체크한 것이다.

질곡, 발신자는 짧게는 수초에서 길게는 서비스엔진의 일정한 시간의 최대시간(본 시스템은 12분으로 되어있다)안에 자신의 메일이 정확한 사용자에게 전달되

<표 2> Nondeliverable mail에 대한 비교

		보낸시간	되돌아온 시간	시간
1차	A	2000-01-28 05:36	2000-01-28 06:58	1시간 22분
	B	2000-01-28 11:43	2000-01-28 11:47	4분
2차	A	2000-1-28 11:41	2000-01-28 12:58	1시간 17분
	B	2000-01-28 11:47	2000-01-28 11:58	11분

<A : 기존의 메일서버, B : 제한된 내일서버>

있는지 확인할 수 있다는 장점이 있다.

4.6 부재중 관리기능의 구현

메일 에이전트의 서비스 엔진은 부재중 사용자에게 메일이 도착하면 사용자가 작성한 특정 메시지를 BODY에 포함해서 메시지를 발송하는 부분으로 처리과정은 다음과 같다

1. 서비스엔진이 메일을 SMTP영역에서 읽어온다.
2. 사용자확인을 한다.
3. 사용자의 정보를 사용자DB에서 확인한다.
4. 부재중 처리로 되어있을 경우, 부재중 메시지를 발신자에게 보낸다.
5. 수신된 메시지를 사용자의 메일영역 DB에 저장하고 처리를 끝낸다.

```

GetUserFound(ReceivedUser)
If GetUserFound = Yes Then
    Mail.From = UID & Domain
    Mail.To = SendID
    Mail.Subject = UID & "님은 현재 부재중입니다"
    Mail.Body = UserAbsentMemo
    Mail.Send
End If
    
```

(그림 8) 부재중 처리 알고리즘

4.7 메일필터링(Mail Filtering)의 구현

메일 에이전트는 사용자가 등록한 특정 메시지의 규칙을 이용해 Mail DB의 특정 영역으로 보내게 된다.

송신한 사람의 주소나 제목에 특정 문자가 포함된 메일을 검색해서 사용자가 등록한 DB의 특정 영역으로 보내게 되는 것으로 다음과 같은 처리과정을 수행한다.

사용자는 웹상에서 보낸사람, 제목, 또는 전체에서 특정 문자가 포함되면 특정편지함으로 보내는 설정을 했다고 가정한다.

1. 메일을 로드한다

2. 사용자 DB에서 메일필터링에 대한 설정이 되어 있는지를 확인 후, 필터링한 수를 가지고 온다
3. "전체"에 대한 설정에서, 보낸사람과 제목에 특정한 문자열이 포함되면, 지정된 편지함의 고유번호만을 해당 Mail DB에 저장하고...
4. 다음 메일필터링에 대해서 수행을 한다.
5. "전체"가 아닐 경우, "제목"과 "보낸사람"에 대해서 3,4의 처리와 같은 방법으로 처리를 한다.
6. 사용자가 설정한 개수만큼 실행 한 후, 필터링처리 과정을 마친다.

```

Room = 0
For J = 1 To AutoDivCount
    If AutoDiv(1, J) = AllSelect Then
        If InStr(Sender & "", AutoDiv(2, J)) > 0 Or
           InStr(Sender.Address & "", AutoDiv(2, J)) > 0 Or
           InStr(Subject & "", AutoDiv(2, J)) > 0 Or
           InStr(HTMLText & "", AutoDiv(2, J)) > 0
        Then
            Room = Cint(AutoDiv(3, J))
            Exit For
        End If
    ElseIf AutoDiv(1, J) = ReceivedSelect Then
        If InStr(Sender & "", AutoDiv(2, J)) > 0 Or
           InStr(Sender.Address & "", AutoDiv(2, J)) > 0
        Then
            Room = Cint(AutoDiv(3, J))
            Exit For
        End If
    ElseIf AutoDiv(1, J) = TitleSelect Then
        If InStr(Subject & "", AutoDiv(2, J)) > 0 Then
            Room = Cint(AutoDiv(3, J))
            Exit For
        End If
    End If
End If
Next J

Reading = NotReading
If Room = -2 Then
    Reading = Delete
    Room = 0
End If
    
```

(그림 9) 메일필터링의 알고리즘

4.8 수신거부의 처리

메일 에이전트는 수신거부를 등록해 놓을 경우 특정 발신인에게 오는 메일(예를 들어 SPAM성 메일일 것이다)을 받아보지 않고 삭제해 버린다. 이와 같은 수행을 하는 수신거부의 처리는 다음과 같다.

1. 메일을 로드한다.
2. 사용자 DB에서, 수신거부로 설정된 주소를 얻어온다(GetProtectLast).



3. 메시지의 발신자의 주소와 등록된 수신기부 주소를 비교한다(IsProtectList).
4. 수신거부된 주소이면, Mail DB에 수신된 메시지를 저장하지 않고, 메일을 지워버린다(MailDeny).

```

Private IsProtectList
Function MailDeny()
    If IsProtectList(Sender.Address & "") Then
        For Each objMessage In colMessages
            objMessage.Delete
        Next
    End If
End Function

Private Sub GetProtectList(UCode)
    Dim I As Integer
    Get DenyAddress by MailDB
    GetProtectAddress(UCode)
    ProtectListCount = 0

    Do While Not GetUserAddress NotFound
        ProtectListCount = ProtectListCount + 1
        GetProtectAddress MoveNext
    Loop
    If ProtectListCount > 0 Then
        ReDim ProtectList(ProtectListCount) As String
        GetProtectAddress MoveStart
        For I = 1 To ProtectListCount
            ProtectList(I) = GetProtectAddress addr
            Result.MoveNext
        Next I
    End If
    Exit Sub
End Sub

Function IsProtectList(Addr)
    Dim I As Integer
    For I = 1 To ProtectListCount
        If ProtectList(I) = Addr Then
            IsProtectList = True
            Exit Function
        End If
    Next I
    IsProtectList = False
End Function
    
```

(그림 10) 수신거부처리 알고리즘

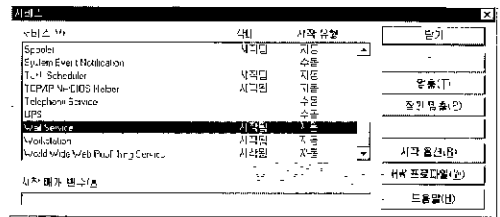
### 5. 구현결과 및 향후 연구과제

#### 5.1 구현결과

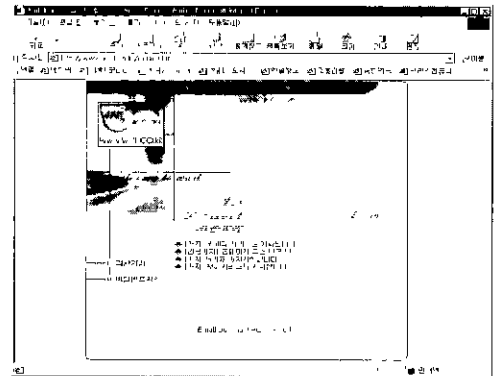
(그림 11)은 서비스엔진이 NT에서 서비스를 시작하고 있는 화면을 보여준다. NT서버가 구동되는 동안 서비스엔진은 항상 SMTP를 체크하게 된다.

(그림 12)는 메일시스템의 접속화면을 보여준다 등록된 사용자만이 사용자의 계정과 비밀번호를 입력 후 메일기능을 사용할 수 있다.

실제로, 사용자는 접속시 ID와 비밀번호로 사용자 인증을 확인하지만, 접속후 부터는 ID로 HTML로 이루어진 페이지와 DB를 접근하는 것이 아니라, 랜덤하게 만들어지는 사용자 고유Code로 페이지 이동과, DB 접속이 이루어지게 된다.



(그림 11) NT 서비스에서 구동되고 있는 서비스엔진

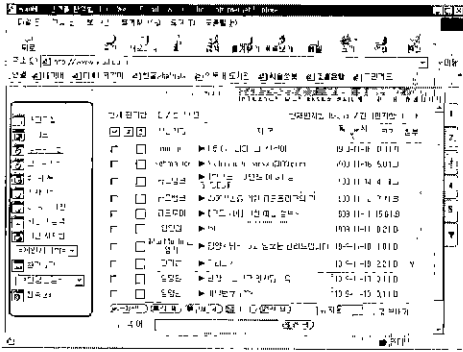


(그림 12) 웹기반 메일 서버의 접속화면

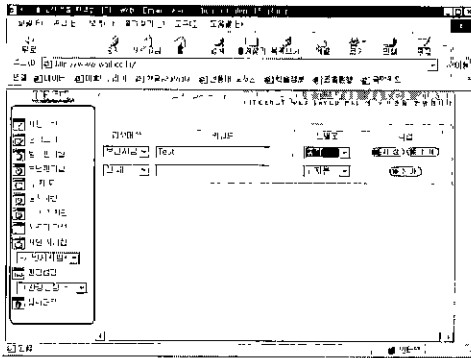
(그림 13)은 웹기반 메일서버의 기본적인 사용자 인터페이스이며, 사용자에게 도착한 메일은 채편지함으로 바로 검색될 수 있다. 채편지함에서 읽은 메일은 받은편지함으로 자동으로 옮겨지게 된다. 만일 사용자가 메일필터링 설정을 해 두었다면, 메일은 특정 편지함으로 보내지게 된다.

(그림 14)는 메일필터링을 설정하는 부분으로 보낸 사람, 제목, 전체에서 설정할 수 있으며, 특정 편지함으로 보내게 된다.

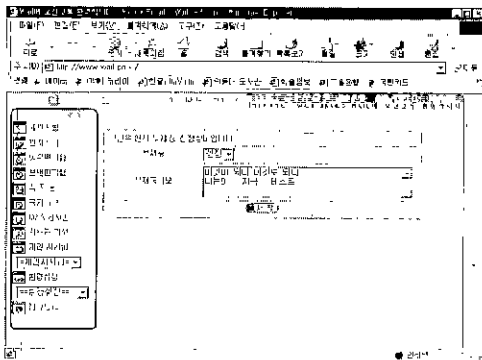
예로 사용자가 대학교수라면 리포트를 Email로 받을 경우에 리포트편지함을 만들어 놓고 제목이 "리포트"인 것만은 리포트편지함으로 저장할 것은 설정해 두면, 메일 에이전트는 SUBJECT에 "리포트"라는 단어가 포함된다면, 메일을 확인후 사용자가 만들어 둔 리포트



(그림 13) 웹기반 메일 서버의 접속후 편지함을 열었을때의 화면



(그림 14) 메일필터링 또는 메일자동분류



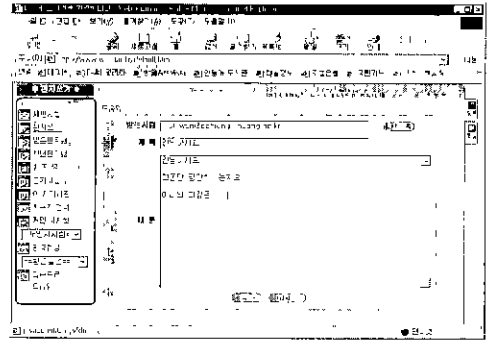
(그림 15) 부재중 관리

편지함으로 메일을 옮겨지게 된다.

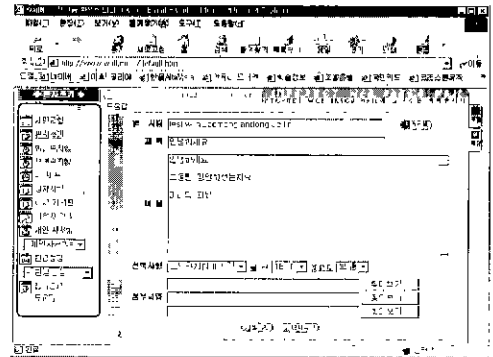
(그림 15)는 부재중 설정을 하는 부분으로 사용자가 얼마동안 메일을 확인하지 못할 경우 설정해 두면, 부

재중일 경우에 사용자에게 전달된 메일에 대해서 자동 응답기능을 갖게 된다. 메일 에이전트는 부재중 메모의 내용을 메일의 BODY에 포함하고, 사용자의 이름으로 메일을 발송하게 된다

(그림 16)과 (그림 17)은 메일쓰기부분으로 사용자가 기본상태에서 메일을 쓰다가 파일이 첨가된 메일을 보내기 위해 확장된 상태로 옮겨지는 것을 나타낸다. 사용자는 우측의 탭을 클릭함으로써 두 가지 상태의 빠른 전환을 할 수 있다. 물론, 이때, 사용자가 입력한 일부 또는 전부의 내용은 넘어가게 된다.



(그림 16) 기본상태의 메일쓰기 화면

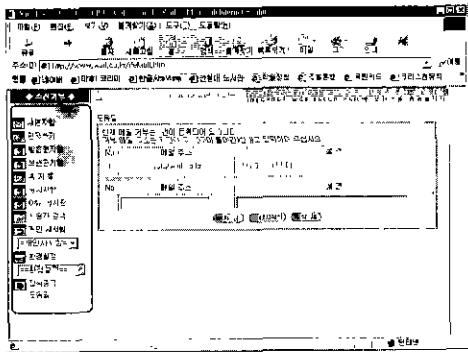


(그림 17) 확장된 메일쓰기의 화면

사용자가 확장된 기능에서 작성한 메일을 특정 편지함으로 저장하는 선택을 하지 않으면 기본적으로 보낸 메일은 보낸 편지함으로 저장된다.

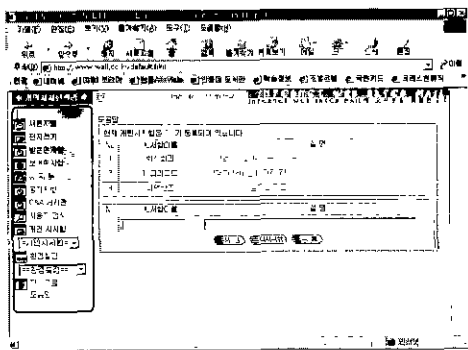
(그림 18)은 개인사서함 관리화면으로 사용자가 원하는 이름의 편지함을 등록할 수 있고, 등록된 편지함에 메일분류니, 메일필터링을 적용해서 사용자가 등록

한 개인사서함에 특정 메일을 저장할 수 있다. 개인사서함을 등록하면서 설정부분에도 사용자가 정의할 수 있기 때문에 편리한 사용자 환경을 지원한다.



(그림 18) 개인사서함 관리 화면

(그림 19)는 수신거부 관리화면으로 사용자는 수신거부할 주소를 등록하고, 등록된 주소로부터 오는 메일은 사용자의 편지함에 저장되지 않고, 서비스엔진에서 바로 삭제해버린다



(그림 19) 수신거부 관리 화면

사용자는 자주 쓰는 주소를 주소록에 등록할 수 있으며, 등록된 주소는 편지쓰기에서 선택할 수 있다. 편지쓰기는 여러 명에게 동시에 보낼 수 있는 다중사용자송신 기능을 지원한다.

모든 편지함에서는 특정편지함이나 전체메일의 날짜, 제목, 본문, 보낸사람의 조합 가능한 검색이 가능하며, 편지함 내의 보낸사람별, 제목별, 받은날짜별, 크기별 등 오름차순과 내림차순의 정렬을 제공한다.

메일보기에서는 보낸사람의 E-mail주소를 주소록에 추가할 수 있는 기능을 제공되고, 편지함의 목록페이지를 열지 않아도, 현재 메일의 다음메일과 이전메일을 검색할 수 있도록 하였으며, 현재 보고있는 메일을 특정 편지함으로 옮겨 놓을 수 있는 기능을 제공한다.

휴지통을 제외한 모든 편지함에서의 메일삭제는 사용자가 복원 가능하도록 휴지통으로 보내는 것만을 의미하며, 휴지통에서의 메일삭제는 복원할 수 없는 영구 삭제를 의미한다.

### 5.2 결론 및 향후 연구과제

본 연구에서는 기존의 메일서버에서는 한계가 있는 기능들을 웹기반으로 구현하였다. 서비스 엔진이라는 개념을 도입해서 사용자가 직접 SMTP서버로부터 메일을 가지고 오는 것이 아니라, 메일 에이전트에서 이를 수행하게 된다. 서비스 엔진은 사용자의 전반적인 기능들을 수행하게 되며, 메일 에이전트가 직접 사용자의 메일을 mail DB로 저장하기 때문에 사용자 급증시 시스템의 과부하를 줄일 수 있었다.

즉, 메일폭증이나, 사용자 급증시 서버의 부하를 줄이는 방법으로 NT Service를 이용해, 미리 설정되어진 일정한 시간마다 새로운 메시지를 체크해 Mail DB로 저장하는 처리와, 사용자 접속시 또는 접속된 사용자가 "새편지함"을 클릭할 때, 새로운 메시지를 체크해 Mail DB로 저장하는 처리를 병행한다. 이렇게 함으로써, 한꺼번에 많은 메일이 쌓이거나, 사용자 급증시 한꺼번에 SMTP서비스를 직접 제어해서 발생하는 서버의 부담감을 크게 줄일 수 있도록 하였다.

메일 에이전트는 일정한 시간이 지나면 SMTP의 메일 디렉토리에서 메일들을 받아온 후, 부제중처리, 잘못된 사용자의 처리, 메일필터링 등과 같은 기능들을 수행하고, 수행한 결과를 보내거나, Mail DB에 저장할 하게 된다.

본 논문에서 제시하는 웹기반 메일 에이전트 시스템은 인텔 프로세서에서 주로 동작하지만 Digital의 Alpha 프로세서에서도 동작한다. NT가 지원하는 맵스의 R4000/R6000, PowerPC등 다른 RISC 프로세서들도 지원하지 않지만, 이들 프로세서들은 NT를 위한 제어기 시스템을 제대로 공급하고 있지 않다. 따라서, 본 시스템이 정상적으로 수행되기 위해서는 Alpha프로세서와 펜티엄 프로세서에서 정상적으로 수행된다. 물론, 80486에서도 수행이 되지만, 현재 NT가 사용하는 많은 서비스를 소

화하기에는 무리가 따를 수 밖에 없다[21].

운영서버로는 NT Server4.0의 IIS4.0으로 구현해서 시스템 관리의 편리와 서비스 엔진과의 호환성을 높였다. 또한 메일서비의 구축을 위해 별도의 소프트웨어를 설치하는 것이 아니라 NT OptionPack의 SMTP만 설치하면, 기초적인 메일서버를 운영 할 수 있다. 본 시스템의 서비스엔진과 HTTP를 지원하는 HTML문서만을 설치하면 누구나 쉽게 메일서비를 구축할 수 있다. 서비스엔진은 6600k정도의 메모리를 필요하며, 전체 시스템이 설치되면, 약 13M정도의 하드디스크공간이 필요하게 된다.

서비스 엔진은 메일서버의 사용자 급증으로 인한 시스템 성능문제를 해결함과 동시에 사용자에게 다양하고 편리한 기능들을 제공한다.

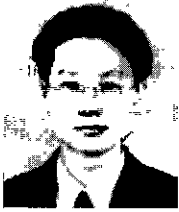
형후, 웹기반 메일 에이전트 시스템의 서비스 엔진을 확장하여 사용자가 시간을 정해서 보낼 수 있는 예약 메일기능과 보낸 메일에 대한 수신확인기능, 특정사용자로부터 도착한 메일에 대한 확장된 처리기능, 메일 도착시 사용자의 이동전화로 메시지를 보내주는 SMS<sup>7)</sup> 연동기능, 보낸 메일이 잘못 보내어졌을 경우의 삭제기능, 메일작성시 한글자동집중 기능 등의 기법을 추가하는 연구를 수행할 계획이다. 현재 이를 통해, 서비스 엔진을 이용하여 시스템의 가용성을 높일 수 있는 연구를 진행중이다

### 참 고 문 헌

[1] W. Richard Stevens, "TCP/IP Illustrated, Vol 1," Addison-Wesley Publishing Company, Inc, 1994  
 [2] 천승현, 김미영, 이귀상, "Implementation of Mail Agent for Group-Messenger sayU"  
 [3] RFC 821, 1460, 1521, 1522.  
 [4] Leon Braginski and Matt Powell, "Running Microsoft Internet Information Server," Microsoft Press, 1998.  
 [5] T Berners-Lee, R.Fielding and H.Nielsen, "Hypertext Transfer Protocol -- HTTP/1.0," RFC1945, 1996.  
 [6] Alex Homer and Chris Ullman, "INSTANT IE4 Dynamic HTML," Wrox Press Ltd, 1997.

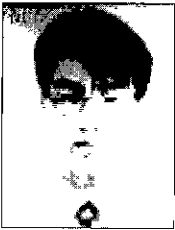
[7] Alex Homer, et al, "Professional Active Server Pages." Wrox Press Ltd, 1997.  
 [8] Richard Mansfield and Evangelos Petroustos, "Visual BASIC Power Toolkit," Ventana Press, 1995.  
 [9] 최대영, "웹기반 보안 전자우편 시스템의 설계 및 구현", 경북대학교 석사학위논문, 1998.  
 [10] "New And Emerging E-mail Applications," IEEE communications Magazine, November 1999.  
 [11] David G.Schwartz, "When email meets organizational memories. Addressing threats to Communication in a learning organization " International Journal of Human-Computer Studies, 51. Article No. ijhc.1999.0276. pp.599-614, 1999.  
 [12] 정진호, "SMTP를 이용한 하이퍼 메일 시스템의 설계 및 구현", 한국과학기술원 석사학위논문, 1993.  
 [13] 독고세준, 이택균, 이형우, 윤성현, 이성환, 김창현, 김태운 "이미지와 텍스트 메시지의 통합 사용자 뷰를 제공하는 전자우편 시스템", 한국정보처리학회 논문지, 제4권, 2호, pp.563-572. 1997.2.  
 [14] 이봉환, 박문호, 이하욱, 주기호, 이찬도, 이남준, 심영진, "IMAP 프로토콜을 이용한 멀티미디어 메일 시스템," 한국정보처리학회 논문지, 제4권, 5호, pp.1297-1307, 1997.5  
 [15] 이정현, 임성락, "인터넷에서 한글 메시지를 위한 전자 메일 브라우저". 한국정보처리학회 논문지, 제5권 제1호. pp.172-180. 1998.1  
 [16] 우준, 하영국, 임신영, 이재광, "자바기반의 배달증명이 가능한 전자메일 시스템 구현", 한국정보처리학회 논문지, 제6권, 제11호, pp.3289-3298, 1999.11.  
 [17] 조진희, 정재열, 한기준, "WWW 환경을 위한 멀티미디어 전자 우편 시스템의 구현". 정보과학회 논문지(C), 제2권, 제3호, pp 296-307, 1996.9  
 [18] 이준연, 권번선, 김영찬, "MIME을 지원하는 OLE 기반 멀티미디어 전자우편 시스템", 정보과학회 논문지(C). 제3권, 제6호, pp 735-742, 1997.12  
 [19] Robert J Hall, "How to Avoid Unwanted Email," Communications of the ACM, Vol 41, Num 3, pp.88-95, March.1998.  
 [20] Rohit Khare, "Seventh Heaven: The Spcc's in The Mail," IEEE Internat Computing, Vol.2, Num 5, pp.82-86, September/October 1998.  
 [21] "Windows NT 4.0 Server," <http://windows.designweb.org/lib/nt4/basic/sec.asp?number=19>

7) SMS(Short Message Service)는 이동전화 가입자들 간에 짧은 메시지를 주고받을 수 있는 단문메시지서비스를 말한다



임 양 원

e-mail : shamaim@andong.ac.kr  
1999년 충주대학교 컴퓨터공학과 졸업(공학사)  
1999년~현재 안동대학교 컴퓨터 공학과 석사과정  
관심분야 : 전자상거래, 에이전트, 웹프로그래밍



권 기 훈

e-mail : kkh@andong.ac.kr  
1996년 안동대학교 컴퓨터공학과 졸업(공학사)  
1995년~1996년 아남 반도체 기술 MIS팀  
1996년~1999년 안동대학교 전자 계산소 조교

1997년~현재 안동대학교 컴퓨터공학과 석사과정  
관심분야 : WebDB, 웨이전트, 전자상거래, 프로그래밍



임 한 규

e-mail : hklm@andong.ac.kr  
1981년 경북대학교 전자계산기공학 전공(박사)  
1984년 연세대학교 산업대학원 전 산 전공(석사)  
1997년 청원관대학교 대학원 전 자계산공학 전공(박사)

1981년~1982년 대한주대공사 전산실  
1982년~1986년 한국전자통신연구소 위성통신연구실 연구원  
1986년~1994년 한국IBM 소프트웨어연구소 선임연구원  
1994년~1998년 한시대학교 전산정보학과 조교수  
1998년~현재 안동대학교 전자정보산업학부 조교수  
관심분야 : 자인언어처리, 영상처리, 멀티미디어