휴대폰에서 Avata컨텐츠 기반의 E-mail Client/Server 모델 설계 및 구현

Design and implementation of the email client/server model using the avatar contents for the mobile phone

이경진, 이정윤, 김강석, 송왕철*
제주대학교 컴퓨터공학과*

Lee kyung-jin, Lee jung-yun, kim gang-seok,
Song wang-chul*
cheju Univ*

요약

현재 모바일 기술이 발달함에 따라 휴대폰을 이용한 인터넷 사용이 크게 증가 하고 있다. 대부분의 휴대폰 사용자의 인터넷 사용은 아바타 및 게임 컨텐츠가 주를 이루고 있다. 본 논문에서는 아바타 및 사용자 정보 관리를 위한 DB 모듈 과 메일 관리 모듈을 구현한 서버와 아바타 이미지를 애니메이션으로 재구성하여 메일 내용을 보여주는 클라이언트 모듈을 설계 하여 클라이언트의 부하를 줄이게 하였다. 휴대폰의 그래픽 및 CPU, 메모리 하드웨어 성능의 한계 때문에 본 논문에서 제시한 방식이 앞으로 많이 사용될 것으로 기대 되며 SK-VM 기반의 Application에 애니메이션 아바타와 E-Mail을 적용시킴으로써 기존에 아바타 서비스가 아닌 새로운 방향으로 아바타를 접목시킨 서비스를 할수 있다.

Abstract

Today, mobile technology is developing very fast and people use internet through mobile phones. Using mobile phones, people use games, email, avatar and other services. We designed client/server modules in order to join email and avatar together. First two major server modules were designed, one of them is DB management module for avatar user information and the other is email management module. One the client side, we designed avatar image animation module and a module for showing the email contents. The purpose of the design is to reduce the performance overhead on the client side as mobile phones cannot be used as a high performance device. We propose and implement a new avatar based service using the SK-VM environment, that is different from the current avatar services.

1. 서 론

오늘날 인터넷이 발전함에 따라 E-Mail서비스와 사이버 공간에서 자신을 표현하기 위한 아바타(Avatar)서비스가 활발하게 이뤄지고 있으며 아바타의 서비스 종류도 무수히 많아지고 있다. 그 중에는 현실 생활을 생생히 표현할 수 있는 3D 기술언어 VRML(Virual Reality Modeling Language)과 이를 제어할 수 있는 EAI(External Authoring Interface) 기술을 이용한 아바타 응용이 연구 되고 있다[1]. 이와 같이 인터넷의 발전과 더불어 모발일 기기도 기존에 단순한 기능을 뛰어 넘어 다양한 서비스의 개발이 진행되고 있다. 휴대폰 역시 기존에 단

순히 전화를 걸고 받는 기능만이 아니라 일정관리에서부터 다양한 기능들이 추가 되고 발전되고 있으며 무선통신을 이용하여 인터넷도 가능하게 되었다. 이처럼 휴대폰 기능이 향상 되면서 기존 PC에 서비스되던 기능들이 휴대폰에도 가능해지고 있지만 여전히 PC에 비해 디바이스의 제한된 속도와 메모리 등의 한계가 남아 있다.

현재 국내에서 휴대폰 아바타 서비스로 제공하고 있는 아바타 이미지는 sis 포맷이다. sis 포맷은 네오엠텔에서 개발한 휴대폰 다운로드용 2D애니메이션 포맷이다[2]. 2D 애니메이션 포맷은 휴대폰 WAP 브라우저를 통하거나 SMS(Short Message Service)를 이용하여 아바타 이미지를 다운로드 받아 휴대폰 초기화면에 쓰이는 정도로 기존 PC에서 자신에 아바타를 활용하는 면에 있어서 휴대폰에서의 아바타 서비스는 아주 미약하다고 할 수 있다.

본 논문에서는 휴대폰 환경에서의 아바타 서비스를 다양하게 적용하는 한 예로 휴대폰 사용자들이 많이 사용하는 E-Mail에 아바타를 결합한 아바타(avatar) 컨텐츠 기반의 E-Mail Client/Server 모델을 설계함으로써 휴대폰 환경에서 단순히 적용하던 아바타를 SK-VM Application 환경에서 메일 송신자의 애니메이션 아바타를 메일 내용과 함께 볼수 있게 구현하였다.

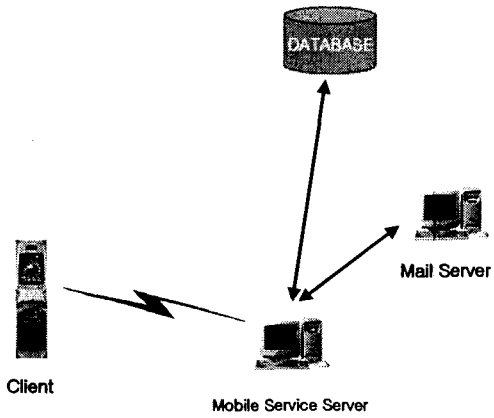
2장에서는 Avatar 기반 E-Mail Clinent/Server 모델의 전체 구조에 대한 설명과 클라이언트 / 서버 세부 모듈들에 대해 자세히 기술하고 3장에서는 2장에서 제시한 Client / Server 모델의 구현에 대해서 살펴본다. 마지막으로 4장에서는 본 논문의 결론 및 향후 연구 방향에 대해 설명한다.

2. Avatar 기반의 E-Mail Client /Server 모델 설계

2.1 클라이언트/서버 구조

그림 1.에서 클라이언트 / 서버 모델의 전체 구조이다. Client는 메일 리스트, 메일 읽기 및 보내기 요청을 Mobile Service Server한다. Mobile Service Server는 Client 요청을 처리하기 위해서DB(Database)에 사용자 정보 및 아바타 이미지 정보를 가지고 오며 Mail Server에서 메일리스트 및 메일 내용, 보내기를 요청하고 아바타 이미지 정보와 메일 정보를 Client로 전송한다. 데이터베이스는 사용자의 프로파일과 아바타 이미지 정보를 가지고 있으며 Mobile Service Server에게 필요한 정보를 제공한다.

Mail Server는 메일전송 및 수신을 담당하는 SMTP서버 이다.



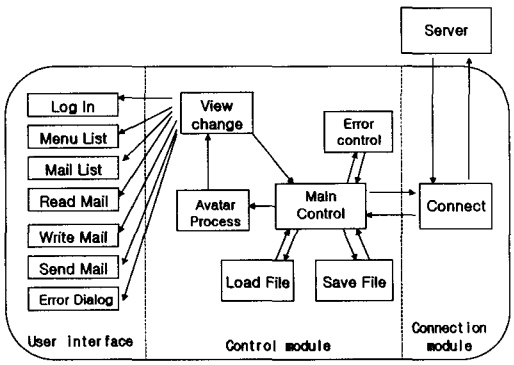
▶▶ 그림 1. Client /Server 전체 구조

2.2 클라이언트 세부 모듈

클라이언트 세부 모듈들의 주요 기능은 TCP/IP Socket을 이용한 서버 접속 기능, 메일읽기/쓰기기능 아바타 이미지를 받아 메일 내용과 같이 애니메이션 아바타 이미지로 재구성하여 보여주는 기능이 있다.

그림 2.는 클라이언트 세부 모듈들을 세부분으로 나누어 보여주고 있다. 우선 사용자인터페이스(User Interface)부분은 사용자와 클라이언트간에 상호작용한다. 커넥션 모듈(Connection moudule)은 TCP/IP 기반 Socket 으로 연결되어 서버와 데이터를 주고 받는 역할을 한다. 컨트롤 모듈(Control module)은 커넥션 모듈에서 얻어온 정보와 사용자 인터페이스에

서 가지고 온 정보를 컨트롤 한다. 컨트롤 모듈 내의 Main Control은 커넥션 모듈과 사용자인터페이스에서 정보를 각각 세부 컨트롤 모듈로 전달하는 역할을 한다. View change는 Main control에서 받은 데이터를 사용자인터페이스 안에 log In, Menu List. ..., Error Dialog 등 데이터 형태에 맞게 보내주는 역할을 한다. Error Control은 클라이언트 전체에 대한 에러 메시지 등과 같은 전체적인 에러 처리를 한다. Save File은 컨텍터(Connect)가 서버로부터 아바타 이미지를 받아 Main Control에 넘겨 주면 아바타 이미지를 받아 휴대폰 메모리에 저장한다. Load File은 휴대폰 메모리에 저장된 이미지를 가져온다. Avatar Process는 연속된 아바타 이미지를 애니메이션 아바타로 보일 수 있도록 처리해 준다.

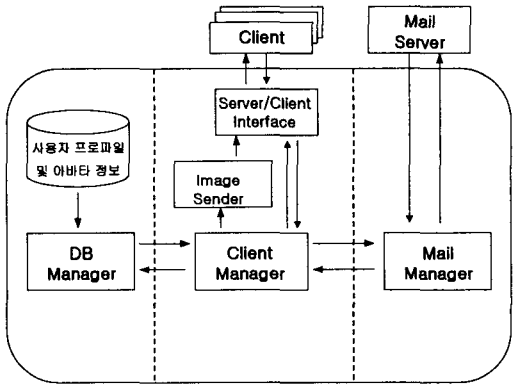


▶▶ 그림 2. 클라이언트 세부 모듈

2.3 Server 세부 모듈

Server 세부 모듈들의 주요 기능은 사용자 인증 기능 그리고 메일서버에게 메일읽기 요청을 하고 보낸 이의 메일 내용과 아바타 이미지를 클라이언트에게 보내준다. 메일 보내기 요청 기능도 가지고 있다.

서버 세부 모듈들을 보여주고 있는 그림 3.에서 Server/Client Interface는 클라이언트와 접속을 유지 하면서 클라이언트에 메일 내용, 아바타 이미지를 보내고 클라이언트로부터 요청을 처리하는 역할을 한다. Client Manager는 각각의 접속한 클라이언트들을 관리 하며 인증 처리 기능과 함께 DB Manager에 사용자 프로파일과 아바타 정보를 요청하며 Mail Manager에게 메일 내용 및 메일 보내기 요청을 한다. Image Sender는 Client Manager에서 보내준 연속된 아바타 이미지들을 Server/Client Inerface에 보내준다. DB Manager는 데이터베이스에 저장된 사용자 프로파일 및 아바타 이미지 정보를 가지고 오는 역할을 한다. Mail Manager는 Mail Server에 있는 메일 리스트 및 메일 내용을 읽어 와서 Client Manager에게 보내주고 메일을 보내는 역할을 한다.



▶▶ 그림 3. Server 세부 모듈

3. Avatar 기반의 E-Mail Client/Server 모델 구현

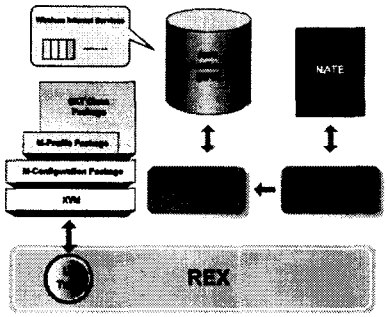
3.1 구현 환경

본 논문에서 설계한 Avatar 기반의 E-Mail Client/Server 모델에서 클라이언트(Client)는 SK-VM 기반 환경으로 구현하였다. SK-VM은 SK- Telecom에서 Clean room 기반으로 자체 개발한 J2ME(Java 2 Micro Eidtion)자바 실행 환경으로 가상머신 및 단말확장 UI, 네트워크,IO를 포함하는 클래스 라이브러리로 구성되어 있으며, 무선이동단말기에서 응용 프로그램을 다운로드 및 실행시킬 수 있는 환경을 제공한다[3].

그림 4.의 요소들이 SK-VM 플랫폼의 구성요소로 단말기에 추가되어 탑재되는 부분이다. Client 구현

은 MIDP를 구현한 M-Profile Package 그리고 SK Telecom에서 무선인터넷 서비스에 필요한 API를 정의한 SKT Class Package를 이용하였으며 Window 2000 에 SK-VM SDK 1.2.1_02를 설치하고 Phone_Emul_64kc 에뮬레이터에서 테스트 하였다.

실제 휴대폰에서 실행하기 위해서는 XVM Application에 대한 descriptor file로서 application의 attribute에 대한 filed들을 포함하고 있는 msd와 Content에 대한 class file, image, sound등을 모두 포함한 내용을 package화 시킨 jar file 쌍으로 구성되며, 또한 WAP Browser 와 연동하여 jar file을 다운로드 하기 위해서는 Apache 2.0.48을 사용하고 무선 인터넷용 Markup 언어인 WML과 WML Script 작성하여야 한다[4][5].



▶▶ 그림 4. SK-VM 플랫폼의 구조

서버는 Window 2000, J2SE SDK 1.4.2와 SMTP를 이용하여 이메일을 보내거나 또는 POP3를 이용하여 메일을 읽어오는 JavaMail API 그리고 JDBC를 지원하는 MySQL Connector/J 3.0을 이용하였으며 MailSever는 SquirrelMail version 1.4.2을 Gentoo Linux에 인스톨하였다. DataBase 는 Mysql 4.0.16을 사용하였다[6][7][8][9].

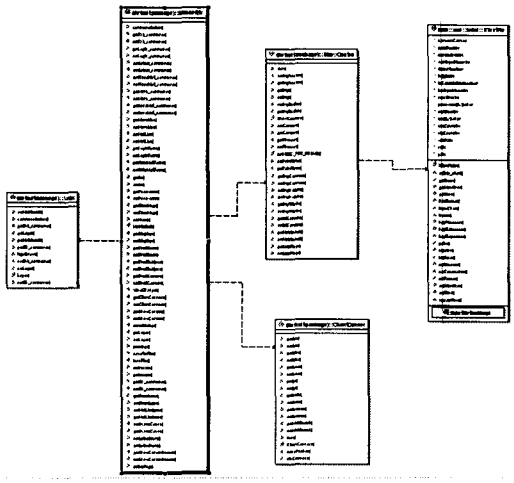
3.2 클라이언트(Client) 구현

클라이언트에서 구현하는 핵심 기능은 서버와 데이터 통신을 위한 TCP/IP socket 기능과 애니메이션 아바타를 구현을 위해 SK-VM에 제공하는 EFS 파일 시스템에 MIDP 규격에 맞춘 png 포맷형식의 아바타 이미지 저장 기능, EFS 파일 시스템에서 저장했던 이미지를 불러오는 기능, 각각의 아바타 이미지를 애니메이션화 하는 기능이다.

[표 1] 클라이언트 구현을 위한 클래스명과 간략한 설명

Class Name	기능
Logo	로고화면 표시
MobileQtel	메인 midlet 클래스로 UI 흐름을 컨트롤하고 ClientConnect 클래스와 ShowContent 클래스를 컨트롤한다.
ClientConnect	소켓을 사용해서 서버와 메시지를 주고 받는다.
ShowContent	애니메이션 아바타와 메일 내용을 보여준다.

표 1.에서 각각 기능들을 가지고 있는 클래스(Class)들을 설명하고 클래스(Class) 사이의 연관계들을 그림 5.처럼 UML을 이용하여 Class Diagram으로 표현 하였다.



▶▶ 그림 5. 클라이언트 클래스 다이어그램 (Class Diagram)

서버와 클라이언트 사이에 데이터를 주고 받기 위해서는 간단한 프로토콜(Protocol)을 설계하여 클라이언트의 요청과 서버의 응답을 정의 하여 에러 처리 및 데이터의 구분과 클라이언트 서버의 동작을 명확하게 할 수 있다. 표 2. 에서도 알 수 있듯이 클라이언트(Client)에서 유저 인증을 위한 요청을 할 때는 Account#id#password 형식으로 메시지를 보내고 서버(Server)는 요청을 받아 처리를 하고 그 결과를 클라이언트(Client)로 보내며 나머지도 위와 마찬가지 이루진다. 클라이언트에 가장 핵심은 송신자의 메일 내용과 송신자의애니메이션 아바타를 클라이언트에서 볼 수 있게 하는 것으로 EFS시스템에서 불러온 아바타 이미지와 송신자의 메일 내용을 인자로 받아 처리하는 ShowContent Class이다. 그림 6.은 아바타 이미지를 그려주는 paint() 함수와 paint()함수를 주기적으로 호출 하는 run()함수를 이용하여 아바타 이미지를 그려 줌으로써 애니메이션 효과를 내고 XTextField를 이용하여 메일 내용을 화면에 그려줌으로써 구현하였다.

그림 7.은 Client를 실제로 실행한 것으로써 클라이언트가 TCP Socket을 이용하여 서버에 접속하여 로그인을 하면 서버로부터 로그인 Account_Ok 메시지를 받아 메뉴화면이 나타난다. 메뉴는 메일보기와 메일쓰기가 있으며 메뉴 보기를 선택하면 Client는 Mail_List 메시지를 서버에게 보내 주고 서버는 List_Ok 메시지와 메일 리스트를 보내주면 화면에 메일 리스트를 보여준다. 그중에 읽고 싶은 메일을 선택하면 readMail#식별번호 메시지를 서버에게 보내면 서버는 Mail Server에서 식별번호에 해당하는 메일 내용을 가지고 오고 데이터베이스에서 애니메이션에 필요한 아바타 이 표 2. Client/Server 프로토콜(Protocol)

의미	Client	Server
유저인증	Account#id#password	성공:Account_Ok 실패:Account_Fail
메일리스트보기	MailList	성공:List_Ok 보낸 후 메일리스트를 보낸다.
메일 읽기	readMail#식별번호	성공:readMail_Ok 식별번호에 맞는 메일 내용을 보낸다. Avata_Ready를 보낸 후 식별번호에 맞는 아바타 이미지 전송 실패:readMail_Fail
메일 쓰기	Msg:받는이#제목#내용	성공:sendmail_Ok 실패:sendmail_Fail
종료	EXIT	Null

미지를 가지고 와서 클라이언트에게 보내주면 받은 메일 내용과 아바타 이미지를 애니메이션 이미지로 재구성하여 보여준다.

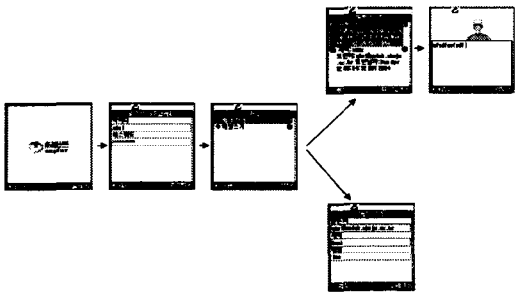
```
protected void paint(Graphics arg0)
{
    int widthSize = getWidth();
    int heightSize = getHeight();

    if(imgBuffer == null){
        imgBuffer = Image.createImage(widthSize,heightSize);
    }
    Graphics g2 = imgBuffer.getGraphics();
    g2.setColor(95,103,255);
    img = (Image)imgVector.elementAt(imgCounter);
    imgWidth = img.getWidth();
    imgHeight = img.getHeight();
    g2.drawImage(img,(widthSize/2)-50, 10, Graphics.LEFT | Graphics.TOP);
    uiContent.setBounds(2,60, widthSize-5,heightSize-10);     //XTextField 그리기위해서 화면 위치를 설정한다
    uiContent.paint(g2);
    arg0.drawImage(imgBuffer,0,0,Graphics.LEFT | Graphics.TOP);
    imgBuffer = null;
}

public void run()     // thread 실행 부분
{
    /*
    long nextTime = System.currentTimeMillis();
    long curTime;
    paintTime = nextTime;
    */
    while(stopFlag){
        synchronized(this){
            try{
                Thread.sleep(100);
                repaint();
                imgCounter++;

                if(imgCounter == 5)
                    imgCounter = 0;
            }catch(InterruptedException ex){
                ex.printStackTrace();
            }catch(Exception e){
                e.printStackTrace();
            }
        }
    }
}
```

▶▶ 그림 6. ShowContent Class 내에 paint()와 run() 함수

다시 메뉴에서 메일 쓰기를 하면 상대방 받는 이에 이 메일 어드레스와 제목 내용을 적어 보내기를 하면 Msg:받는이#제목#내용 메시지 형식으로 서버에게 보내면 서버는 Mail Server에게 메일을 보낼 것을 요청 한다.

3.3 서버(Server) 구현

서버 구현에 핵심 기능은 DB(Database)에서 사용자 프로 파일 및 아바타 정보를 읽어 오는 기능과 Mail Server에서 메일을 가지고 오거나 메일 보내는 것을 요청하는 기능 마지막으로 각각의 클라이언트들에 접속을 유지하며 클라이언트에 요청에 응답하는 기능이다.



▶▶ 그림 7. 휴대폰에서 실행

표 2는 각각 기능들을 포함한 클래스들을 정리하고 클래스의 기능을 간략하게 설명하고 있다.

그림 8은 MailServer Class 내에 아바타 이미지를 클라이언트(Client)로 보내주는 sendImage()함수에 대한 소스이다.

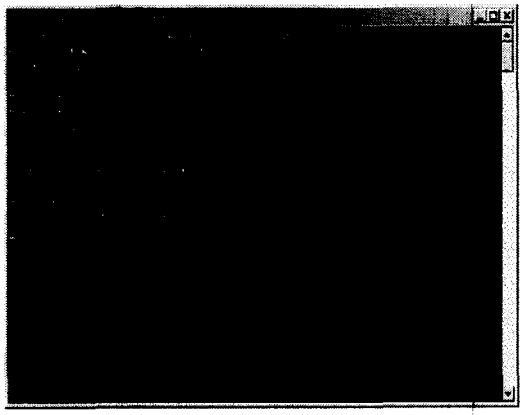
그림 9는 Server를 실행, 클라이언트와 서버사이에 동작상태들을 보여준다.

[표 2] 서버 구현을 위한 클래스명과 기능

Class Name	기능
MailServer	클라이언들에 접속을 유지하며 클라이언트에 요청을 처리하기 위해 MailMgr와 DBMgr에서 정보를 받아 클라이언트에 응답한다.
MailMgr	Mail Server에서 메일 내용 가지고 오거나 메일 보낸다.
DBMgr	DB(Database)에서 사용자 프로파일 및 아바타 정보를 가지고 온다.

```
private void sendImage(String image)
{
  try
  {
    File file = null;
    FileInputStream inputFile = null;
    byte[] buf = null;
    for(int i = 1; i <= 5; i++)
    {
      file = new File("c:/MProject/image/" + image + "/"  +  i  +  ".png");
      inputFile = new FileInputStream(file);

      buf = new byte[inputFile.available()];
      inputFile.read(buf);

      sendMessage("Avata_Ready");
      System.out.println(identifier + " avata_ready" + i);
      dos.writeInt(buf.length);
      dos.write(buf);
      dos.flush();

      inputFile.close();
      file = null;
      buf = null;
    }
  }
  catch(IOException e)
  {
    System.out.println("Image send fail...");
  }
}
```

▶▶ 그림 8. MailServer Class 내에 sendImage() 함수



▶▶ 그림 9. Server의 동작 상태

4. 결론 및 향후 연구 방향

본 논문에서는 메일을 읽고, 보내는 것만이 아니라, 메일을 읽을 때는 보낸 이의 움직이는 아바타를 볼 수도 있으며, 자신의 아바타를 상대방에게 메일을 보내면서 같이 보낼 수 있게 하는 Avatar 컨텐츠 기반의 E-Mail Client / Server 모델을 설계 하고 구현하였다. 이는 기존에 휴대폰 초기화면에 사용하던 아바타 서비스가 아닌 SK-VM 기반의 Application에 애니메이션 아바타와 E-Mail을 적용시킴으로써 기존에 서비스 되던 아바타 서비스가 아닌 새로운 방향으로 아바타를 접목시킬 수 있다는 것을 보였다. 그

리고 서버에 사용자 정보와 아바타 정보를 관리함으로써 휴대폰에 부담을 줄일 수 있게 하였다.

구현 당시 우리나라에서 표준화 신행중인 위피(WIPI)에 대한 언급되고 있었으나, WIPI 지원 휴대폰은 출시가 이뤄지지 않은 상태였기 때문에, SK-VM 환경에 휴대폰을 이용하였다.

앞으로 SK-VM 기반한 프로그램을 WIPI 기반으로의 변환과 관련된 연구와 단순히 애니메이션 되는 아바타가 아니라 3D 기반의 아바타를 표현할 수 있는 3D엔진 연구가 진행될 필요가 있다.

▌참고문헌▐

[1] "EAI를 이용한 인터넷 상에서의 아바타 동작에 관한 설계 및 구현", 배재대학교, 정회경, 안성욱, 정재길
[2] 네오엠텔 홈페이지, http://neomtel.co.kr
[3] XTD-1 Tutorial, http://developer.xce.co.kr/korean/
[4] Apache 2.0.48, http://httpd.apache.org/
[5] WML(Wireless Markup Language) Wap Forum, http://www.wapforum.org
[6] J2SE SDK 1.4.2, http://java.sun.com/j2se/index.jsp
[7] JavaMail API, http://java.sun.com/products/javamail/
[8] MySQL, http://www.mysql.com/
[9] SquirrelMail version 1.4.2, http://www.squirrelmail.org/