
SK-VM 기반의 Avatar E-mail 서비스 시스템 설계 및 구현

Design and Implementation of SK-VM based Avatar E-mail Service System

이정윤*, 이경진**, 송왕철***
제주대학교 컴퓨터공학과

Jung-Yun Lee(yuniobba@cheju.ac.kr)*, Kyung-Jin Lee(indigo@ncl.cheju.ac.kr)**,
Wang-Cheol Song(philos@cheju.ac.kr)***

요약

현재 모바일 기술은 빠른 속도로 발달하고 있으며 이러한 상황에 힘입어 휴대폰을 이용한 인터넷 사용이 증가하고 있다. 휴대폰 사용자의 인터넷 사용은 대부분 아바타 및 게임 콘텐츠가 그 주를 이루고 있다. 본 논문에서는 SK-VM을 기반으로 기존의 단순한 아바타 서비스가 아닌 이메일과 애니메이션 아바타를 접목시킨 클라이언트/서버 시스템을 설계, 구현했다. 서버는 아바타 및 사용자 정보 관리를 위한 DB 모듈과 메일 관리 모듈로 구현했고, 클라이언트는 아바타 이미지를 애니메이션으로 재구성하여 메일 내용과 함께 보여주도록 구현했다. 이 모델은 휴대폰의 제한적 하드웨어 성능에 맞춰 클라이언트의 부하를 줄이도록 설계했다. 모바일 기술이 빠르게 발전하고 있지만 휴대폰의 제한된 성능 때문에 본 논문에서 제시한 모델이 앞으로 많이 사용될 것으로 기대된다.

■ 중심어 : |모바일 |아바타 |이메일 |휴대폰 |SK-VM |

Abstract

These days, mobile technology is developing very fast. It has increased the usage of the Internet by mobile phones. Nowadays the avatar and game contents services are most popular in mobile phones using the Internet. In this paper, we propose a Client/Server system to combine the animation avatar and e-mail services on SK-VM which are different from simple avatar services. The server which consists of DB (database) module and mail management module is designed in order to manage the information about avatar and the user. The client is designed so as to show the email contents and avatar animation on the mobile phone. This model is intended to reduce the performance overhead on the client as much as possible. We expect that this module would be useful in the mobile phones with limited resources, although the technology is being developed very rapidly.

■ keyword : | Mobile | Avatar | e-mail | Mobile Phone | SK-VM |

I. Introduction

Since recently, content-based services in mobile devices have been a principal concern of many people [1][2]. Most popular contents in the mobile phone are MP3, game contents, Avatar and etc. Avatar on mobile phone is now evolving in services for an example, showing telephone caller's avatar, and in techniques such as his/her own avatar using pictures taken by built-in camera [3][4]. Also, mobile users usually desire to use the mobile contents synchronized with PC based avatar contents.

Email service as one of the most important killer applications in the Internet is also evolving through the development of several kinds of contents. However, email service in mobile phone is still used in simple form of text while email users are constantly increasing. In this paper, we try to graft avatar services into email service in mobile phones. A client/server model is designed as the Avatar based email service using SK-VM as a virtual machine on the cellular phone. With this model, we have implemented an email service system in which the animated Avatars for the email sender are reconstructed and sent.

The remainder of this paper is organized as follows. Section 2 describes SK-VM as the basis of this system. We explain the overall architecture of the Avatar based Email Client/Server model and its modules in section 3, and illustrate implementation of the model in section 4. Section 5 describes the conclusions.

II. SK-VM

SK-VM is a clean room implemented J2ME

(Java 2 Micro Edition) runtime environment developed by SK-Telecom, Korea. XVM is a J2ME platform which enables Java applications to be downloaded and executed on the mobile phones.

SK-VM is composed of XVM (eXtended Virtual Machine) which functions as KVM, M-Configuration, which is a clean room implementation of CLDC, clean room implemented MIDP M-Profile, and OEM-specific class to develop differentiated service.

Now the highly developed multimedia services are available on XVM shipping handsets such as network games, real time stock trading, music, instant messaging service and so on [5][6].

III. Avatar E-mail System Model

The mobile phone has limited resources, therefore, it is required that most of the work should be designed to be processed on server side and the client side is light-weighted.

[Fig. 1] shows the overall architecture of this C/S email service system. When the Mobile Service Server (MSS) receives the requests such as email-list, read, send and etc from the client, MSS retrieves the information about the user from database (DB) through correct user id and password and sends the request from the client to the mail server. After getting all data, MSS sends the avatar image data and email data to the client. MSS has the role to control overall operations, and DB restores user profile and his/her avatar image data. The Mail Server is simply a basic SMTP server.

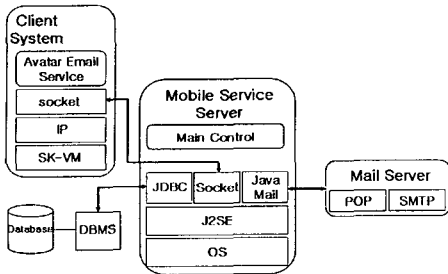


Fig.1. Avatar E-mail System

3.1 Client Module Design

The client module consists of three parts as shown in [Fig. 2]. First, the User Interface plays a role to show users the main menu and email contents, and receive the request from the user. Second, the Control module is responsible for transferring and receiving data through socket based connection to/from MSS. It processes user's requests using information from the connection module and the user interface. The role of the Main Control sub-module in the Control module is to transfer data from the Connection module and the User Interface to the Control module. The View Change sub-module modifies data from the Main Control according to the data type for user interfaces such as Log In, Menu List, Error Dialog, and etc, and sends it to the corresponding interface. The Error Control module performs error processing for all error messages that could occur in the client machine. When someone's avatar is sent from MSS with the email, the Save File saves the avatar to the phone memory as the EFS file system. The Load File can load the image file from the EFS file system. The Avatar Process shows several continuous Avatar images as one animated image. We intend to process all of contents for email and avatar images on the server side and assign the client system as simple tasks as possible. Basically the MSS sends the

client several avatar images as a JAR file and forwards the email without header data. The main functionalities of the client module are connection to the server through TCP/IP socket, reading/writing emails, and displaying animated avatar images of the email sender with email contents from MSS.

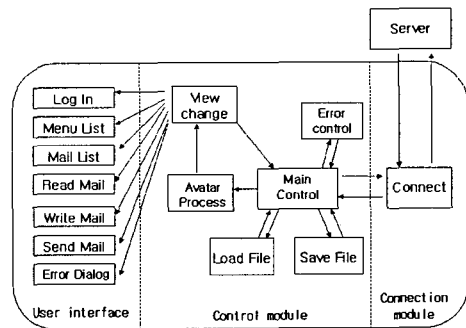


Fig. 2. Modules in the client system

3.2 Server Module Design

The main functionalities of the MSS modules are user authentication with user id and password, reading/writing emails from/to the email server, and sending email and the sender's avatar images received from DB to the client. The User Interface in the server module of [Fig. 3] is the socket based module to forward the email and its avatar images to the client and process requests from the client. The Client Manager manages separately each client connected to the Server by creating a thread per user. Also, it has functionalities for user authentication, request for user profile and his/her avatar data to the DB Manager, and request to the Mail Manager for handling emails. The Image Sender module forwards avatar images from the Client Manager to the User Interface. The DB Manager has a role to retrieve the user profiles and its avatar image data from the database. The Mail Manager gets email lists and

its contents from the Mail Server, and sends them to the Client Manager. The MSS parses and processes data from the Mail Server and the database, and forwards them to the client. It usually parses data, screens out control data and headers, and makes them serialized. Its purpose is that the client system can display the contents to the user without any additional processing in the mobile device.

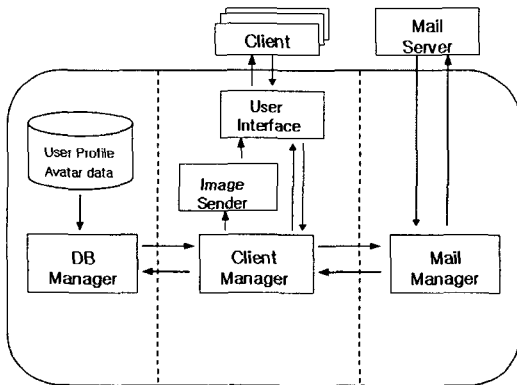


Fig. 3. Modules in the server system

IV. System Implementation

4.1 Implementation Environment

In this paper, we implement the client in SK-VM environment as the cellular phone platform. The implementation uses M-Profile package providing MIDP and SKT class package defining API necessary to the mobile internet services of SK Telecom. While developing it, we have used the phone_emul_64kc emulator based on SK-VM SDK 1.2.1_02 for test before the client can be deployed in SK-VM mobile phone. The client program can be downloaded as the application in the form of JAR file, depending on the information contained in the MSD file. The MSD file includes fields containing necessary

information about the application. In order to download the JAR file using the WAP browser, the download server is deployed using Apache 2.0.48, and is coded by the WML and the WML script[7][8].

The server system of Windows 2000 has J2SE SDK 1.4.2, JavaMail API, and MySQL Connector/J 3.0 supporting JDBC. The mail server is deployed using SquirrelMail version 1.4.2 on Gentoo Linux. The database is Mysqil 4.0.16 [9-12].

4.2 Client Implementation

The functionalities implemented in the client are socket based communication from/to MSS, storing avatar images as the png format of MIDP specifications to EFS file system of SK-VM, retrieving images stored in the EFS, and animating separate avatar images.

Table 1. Classes of the client

Class Name	Functionality
Logo	to display logotype
MobileQtel	to control flow of UI, ClientConnect class, and ShowContent class. (It is a MIDLET class playing the central role for client.)
ClientConnect	to transfer messages to/from the server using socket.
ShowContent	to show animated avatar and email contents
XTextFieldT	to support multiple lines in a page

The client has five classes in [Table 1]. Logo class is to display logotype as the initial page. MobileQtel class as a central MIDLET class controls flow of UI according to user's requests, that is, it acts as the user interface. It controls message flow between ClientConnect class and ShowContent class. ClientConnect class has the main function to connect to the server using socket. It forwards messages to MSS as the

user's request from MobileQtel class and transfers messages from MSS to the user through MobileQtel class. ShowContent class functions to show contents from MSS such as email contents and avatar images after processing the avatar images retrieved from EFS file system. XTextFieldT class displays one line text as the multiple lines on the mobile phone [5].

4.3. Protocol

We design a simple protocol to transfer data between MSS and the client as shown in [Table 2]. Five operations between the server and the client are defined i.e., user authentication, getting email lists, reading an email, writing an email and exiting. When parameters are needed in each operation, "#" is used as the delimiter. For example, the client can pass MSS a message of "Account", id and password using "#" like "Account#id#password" for user authentication. Then, MSS understands that this request is for authentication, so MSS processes it and responds to the client. If the user is successfully authenticated, MSS sends back Account_OK to the client, and if not, it responds with a message of Account_Fail.

Table 2. Protocol between the Client and the Server

Cases	Client request	Server response {success : (1) , failure : (2)}
authenticate user	Account#id#password	(1) Account_Ok (2) Account_Fail
get email list	MailList	(1) List_Ok email lists
read email	readMail #identification number	(1) readMail_Ok email contents according id No. Avata_Ready avatar images according id No. (2) readMail_Fail
write email	Msg:receiver#title #contents	(1) sendmail_Ok (2) sendmail_Fail
exit	EXIT	Null

4.4 MSS(Mobile Service Server) Implementation

MSS's functionalities are retrieval of the user profile and his/her avatar data from DB, request of read/write emails from/to the Mail Server, and responses to the client's requests while maintaining the session information. [Table 3] summarizes the functionalities and classes of MSS.

MailServer class processes the client's requests as threads while maintaining the connection to each client. After analyzing the client's requests, it requests information to MailMgr class and DBMgr class depending on user's request, and transfers their response to the client. The function of MailMgr class is to retrieve/send emails from/to the Mail Server according to requests from MailServer class. DBMgr class forwards the MailServer class user profiles and avatar images when MailServer class requests.

Table 3. Classes of Server

Class Name	Functions
MailServer	While maintaining connection to the client, it processes client's request by transferring data from MailMgr class and DBMgr class to the client.
MailMgr	It retrieve/send emails from/to Mail Server.
DBMgr	It retrieves user profiles and avatar images from DB (Database).

4.5 Behavior of Model

Implemented system's operations are shown through example figures. [Fig. 4] shows several pages from initial page. It sequentially includes logotype, authentication, and email retrieval/sending pages. [Fig. 5] shows messages generated in MSS when a client is connected. It displays number of clients, client's id and so on.

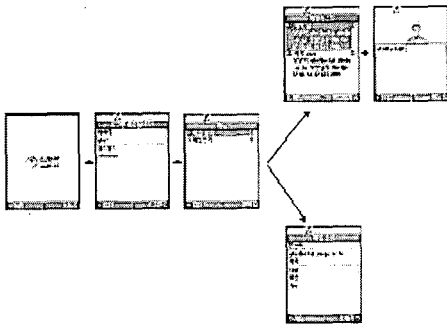


Fig. 4. Behavior of the Client

```

C:\WWW\Programs\20VM\VM.exe
[Server] Server Started. Accepts the client...
[Client] Number of client : 1
[Server]
[Client]
[Server] Mail List
[Client] Mail List read
[Server] send Message
[Client]
[Server]
[Client] read Mail_Granted:11
[Server] read message
[Client] read msg
[Server]
[Client] select Member of client : 1

```

Fig. 5. Behavior of the MSS Server

V. Summary

Since the use of mobile internet has increased rapidly, applications related to avatar have been increasingly developed. In most cases existing avatar content-based applications are simple, in which a favorite avatar is selected and is used on the initial screen of mobile phone. In this paper, we combined avatar contents and email service on the mobile phone. We designed light-weight client, MSS as the server, and a simple protocol between the server and the client. The implemented system is being operated well on real mobile phones with SK-VM.

참고 문헌

- [1] A. Carzniga and A. L. Wolf, "Content-based Network : A New Communication Infrastructure," NSF Workshop on an Infrastructure for Mobile and Wireless Systems, (Scottsdale, AZ), 2001.
- [2] H. A. Jacobsen "Middleware Service for Selective and Location-based Information Dissemination in Mobile Wireless Networks," Advanced Topic Workshop on Middleware for Mobile Computing, (Heidelberg, Germany), 2001.
- [3] K. H. Kim, "A Transmission Service Method for Processing Visual Recognition of Sender Information," Korea Information Science Society, Vol.10, 2004.
- [4] M. Y. Lee, S. J. Lee, and K. S. Ahn, "A Study On The Avatar Generation System Using Camera Embedded Phone.," Korea Information Science Society, 2003.
- [5] http://developer.xce.co.kr/korea_n
- [6] <http://www.sktelecom.com>
- [7] <http://httpd.apache.org/>
- [8] <http://www.wapforum.org>
- [9] http://java.sun.com/j2se/index.js_p
- [10] http://java.sun.com/products/java_mail/
- [11] <http://www.mysql.com>
- [12] <http://squirrelmail.org>

저 자 소 개

이 정 윤(Jung-Yun Lee)

준회원



- 2004년 2월 : 제주대학교 컴퓨터 공학과(공학사)
- 2004년 3월~현재 : 제주대학교 컴퓨터공학과(석사과정)
- <관심분야> : USN

이 경 진(Kyung-Jin Lee)

준회원



- 004년 2월 : 제주대학교 컴퓨터공학과(공학사)
- 2004년 3월~현재 : 제주대학교 컴퓨터공학과(석사과정)
- <관심분야> : 정책기반 네트워크

송 왕 철(Wang-Cheol Song)

정회원



- 1982년 2월~1986년 2월 : 연세대학교 식품공학과(공학사)
- 1987년 3월~1989년 2월 : 연세대학교 전자공학과(공학사)
- 1989년 3월~1991년 2월 : 연세대학교 전자공학과(공학석사)
- 1991년 9월~1995년 8월 : 연세대학교 전자공학과 (공학박사)
- 2001년 2월~2002년 1월 : University of Western Ontario, Postdoctoral Fellow
- 1996년 3월~현재 : 제주대학교 통신컴퓨터공학부 (부교수)
- <관심분야> : 망관리, 인터넷 QoS, 정책기반 네트워크, USN