

IMAIL: A WAP MAIL RETRIEVING SYSTEM

Herman Rao[†], Yuu-Heng Cheng^{*}, Kai -Hsiung Chang^{*}, Yi-Bing Lin^{*}

[†]AT&T Laboratories Research.
Email: herman@research.att.com

^{*}Department of Computer Science and Information Engineering,
National Chaio Tung University, Hsinchu Taiwan.
Email: {yhcheng, khchang, liny}@csie.nctu.edu.tw

ABSTRACT

This paper proposes iMail, a WAP mail retrieving system designed for wireless handheld devices to access Internet email service. In mobile data networks, the communication bandwidth is limited and the wireless handsets are constrained by small display, powerless CPU, limited memory, and unhandy input/output interface. To support mail retrieving in a mobile environment, we implement several features for iMail: mail filtering, two-phase fetch and predefined list reordering. Based on the user's filter options, mail filtering eliminates non-necessary mails sent to a wireless handheld device. Two-phase fetch is used to reduce transmission overhead of message delivery. We use predefined list reordering to speed up text typing in the wireless handheld devices. We also propose three mail notification mechanisms for iMail. When a new mail arrives, the mail notification mechanism informs iMail users immediately. Through the descriptions of the iMail features, this paper provides design and implementation guidelines for WAP applications.

1 Introduction

Mobile phone services have been widely used all over the world in the recently years. For example, by June 2000, more than half of the populations in Taiwan have subscribed to mobile phone service. With growing usage of Internet for all kinds of information, voice communication is insufficient for mobile phone users and wireless Internet services become a major trend in business. However, in the wireless environment, the terminal devices and the networks are different from that of the desktop or laptop Internet access environments. The wireless devices are restricted by limited CPU power, memory size, battery lifetime, small screen display, and unhandy input interface. Furthermore, the wireless network suffers from low bandwidth, long latency, and unstable connectivity.

To address the above limitations, the Wireless Application Protocol (WAP) [1][2] was proposed for the presentation and delivery of wireless information and telephony services. WAP facilitates the implementation of contents and applications on various types of wireless

terminals. The WAP architecture is shown in Figure 1. In this figure, a mobile customer uses a WAP handset to access Internet services through the WAP gateway. The term "WAP handsets" represents cellular phones, PDAs, or other wireless devices that have WAP capability. In WAP, a Remote Access Server (RAS) provides PPP connection between the wireless network and the IP network [4]. Note that if the WAP is supported by wireless bearers such as General Packet Radio Services (GPRS) [3], then RAS may have already been supported by the wireless network nodes such as Gateway GPRS Support Node, and the external RAS is not required. WAP contexts are written in Wireless Markup Language (WML) [5]. When a WAP handset requests information from an origin server, the request would be sent through a WAP gateway and then sent to the origin server. The origin server returns the requested contexts to the WAP gateway. The WAP gateway is responsible for encoding the WML contents into binary codes. This operation significantly reduces the amount of data delivered in the low bandwidth radio link. Finally, the WAP gateway transmits the encoded contexts to the WAP handset through wireless network, and the corresponding user accesses the information by a micro-browser on the WAP handset.

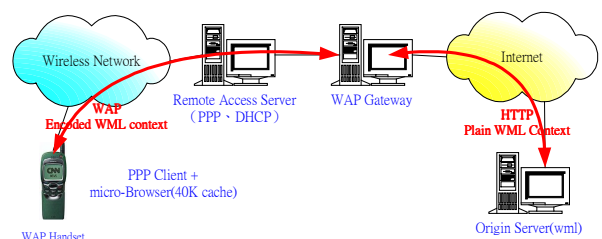


Figure 1: WAP Architecture

The WAP protocol stack is shown in Figure 2. The Wireless Application Environment (WAE) [6] is an application environment based on World Wide Web (WWW) and mobile telephony technologies. The primary objective of WAE is to establish an interoperable environment that allows service providers to build applications and services on top of different wireless platforms. WAE includes a micro-browser environment with functionalities such as Wireless Markup Language (WML) [5], WMLScript [7],

Wireless Telephony Application (WTA [8], WTAI [9]), and Content Formats [10]. The Wireless Session Protocol (WSP) [11] provides connection and connectionless session services for the application layer. The Wireless Transaction Protocol (WTP) [12] is a lightweight transaction protocol which provides unreliable one-way requests, reliable one-way requests, and reliable two-way request-reply transaction services. The Wireless Transport Layer Security (WTLS) [13] is a security protocol based on Transport Layer Security (TLS) protocol, formerly known as Secure Sockets Layer (SSL). The Wireless Datagram Protocol (WDP) [4] is the transport layer protocol of WAP. For bearer services supporting IP, the transport layer of WAP is UDP.

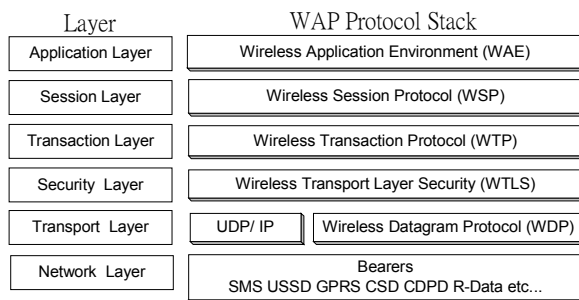


Figure 2: The WAP Protocol Stack

Base on the WAP architecture described above, we propose a mail retrieving system named iMail, which is designed to facilitate sending and receiving mails from WAP handsets. The iMail system provides filtering functionality for users to filter out undesirable messages to be received on the WAP handset. In the current iMail implementation, the wireless network is Global System for Mobile Communications (GSM) [14][15].

In this paper, we first introduce the iMail architecture over the GSM network. Then we describe the iMail system components and functionality. We propose several features in iMail: mail filtering, pre-defined list reordering, and two-phase fetch. These features are designed to support convenient mail access for WAP handsets. In addition, we show iMail deployment and address some difficulties we encountered in implementing iMail. Finally, we discuss some mail notification mechanisms that can be deployed in the WAP environment.

2 iMail Architecture

Based on the WAP architecture in Figure 1, the iMail system is illustrated in Figure 3. The iMail server (Figure 3 (A)) is an origin server. All end clients communicate with the iMail server through an interface implemented by Common Gateway Interface (CGI), which consists of two parts: iMail service part (Figure 3 (B)) and iMail profile configuration part (Figure 3 (C)).

The WAP handset (Figure 3 (D)) interacts with the iMail service part to perform operations such as retrieving, replying, and forwarding emails. These mail operations result in communication between the iMail server and the user's mailbox (Figure 3 (E)).

iMail profile configuration part allows users and administrators to manipulate the user profiles. By using WAP handset, a user may configure his/her profile through the WML interface (see the link between Figure 3 (G) and Figure 3 (C)). Same operations can be performed through the HTML interface for wireline Internet access (Figure 3 (F)). Through the HTML interface, an administrator can log in the administration page to perform create, update, delete, or search operations on the iMail user profiles.

Several simulation environments [17][18][19][20] are available to develop and test WAP services. In our development environment, the wireless network and the WAP handset Figure 3 (G) is replaced by a WAP simulator as shown in Figure 4 (D). The simulator simulates all data communication details and internal network behavior between WAP gateway and WAP handset (Figure 3 (G)). The WAP simulator can be connected either directly to an origin server or indirectly to that server through a WAP gateway. In this environment, the iMail system was developed and tested, and then actually runs on the GSM network.

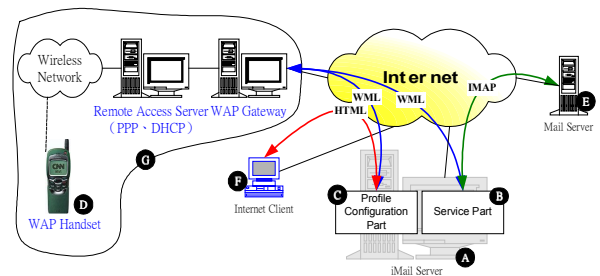


Figure 3: iMail Architecture

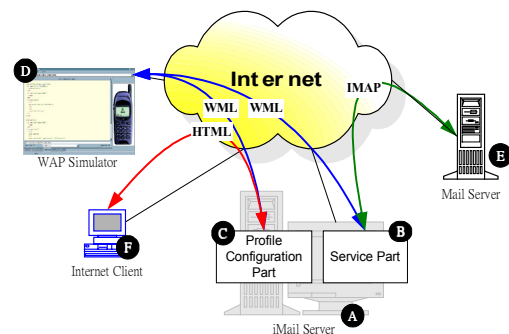


Figure 4: iMail System Simulation Environment

3 iMail Components

The components of the iMail server are shown in Figure 5, which include user profile database, service part, and profile configuration part.

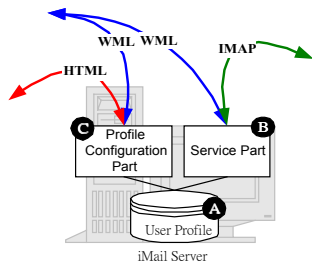


Figure 5: iMail Server Components

3.1 User Profile Database

User profile (Figure 5 (A)) is a database that contains the iMail user accounts. For each person, the corresponding profile contains four fields described as follows.

The *user information* field stores the user identity and authentication information such as mobile telephone number and iMail password. The *mail server* field stores the configuration information of the user's mail server including the server address, port number, login name, and password. The *filter options* are used to select delivered mails based on the mail header. To reduce the amount of information delivered to the WAP handsets, the iMail system selects mails that match one or more of the Sender, Receiver, or Subject options specified by the user. Details of the filter options will be described in Section 4.1. The *predefined lists* include a pre-defined message list and a pre-defined address list. These two lists store messages and e-mail addresses pre-typed by the user. To reply or forward a mail, the user may pick up pre-defined e-mail addresses and attach a pre-typed message with the mail rather than typing the texts through the WAP handset. Details of the pre-defined lists will be described in Section 4.2.

3.2 Service Part

The service part (Figure 5 (B)) provides a WML interface for a user to retrieve email from a specific mail server that supports IMAP4rev1 [16]. Both WML contents and IMAP operations will be described in Section 5. To access the service part, the user first inputs a name and a password for authentication. If authentication is successful, the iMail main menu is prompted (Figure 6 (a)). Then the user performs iMail operations. An example is described below.

The user may select “* Read Mail” to access his/her mailbox through the service part. Based on the user's mail server configuration, the service part fetches the mail

headers from the user mailbox by using the IMAP protocol. The service part then replies the subjects of mails that match user's filter options. These subjects are shown on the WAP handset (e.g., see Figure 6 (b)). When the user selects a specific mail to read, the service part fetches the mail body of the selected mail subject (Figure 6 (c)). The mail title and the mail body are delivered to the WAP handset through a procedure called “two-phase fetch”, which will be elaborated in Section 4.4.

After reading a mail, the user may decide to reply or forward the mail, or just go back to the main menu and read the next mail. To forward a mail (Figure 6 (d)), the following steps are executed.

1. The user selects the receiver address from the pre-defined address list (Figure 6 (e)).
2. The user may select a specific message from the pre-defined message list and attach it to the forwarded mail (Figure 6 (f)).
3. Finally, the user selects the option “* Send Mail” to forward the mail.

Replying a mail is similar to forwarding a mail, except that the user needs not to select the receiver address.

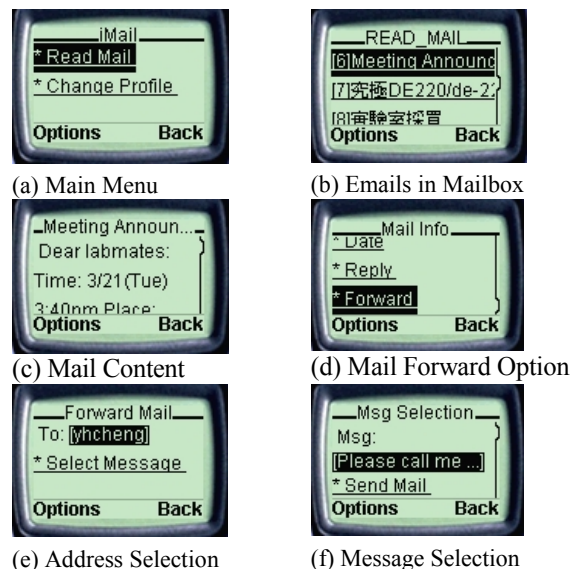


Figure 6: Read/Forward Mail Operations

3.3 Profile Configuration Part

A user can login from a WAP browser (through WML interface by using a WAP handset; see Figure 7) or web browser (through HTML interface by using a personal computer; see Figure 8) to configure his/her profile. WML interface provides a subset of configuration operations (such as filter setups) to manipulate the user's profile. Infrequently used operations (such as user and mail server information update) are not provided in the WML interface, and are only provided in the HTML interface. An administrator can only

login from a web browser to add, delete, or update certain user profiles (see Figure 9).

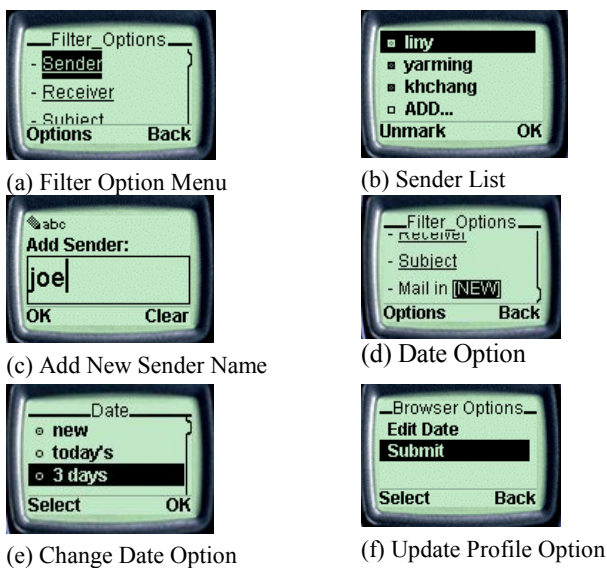


Figure 7: Profile Configuration in WML Interface

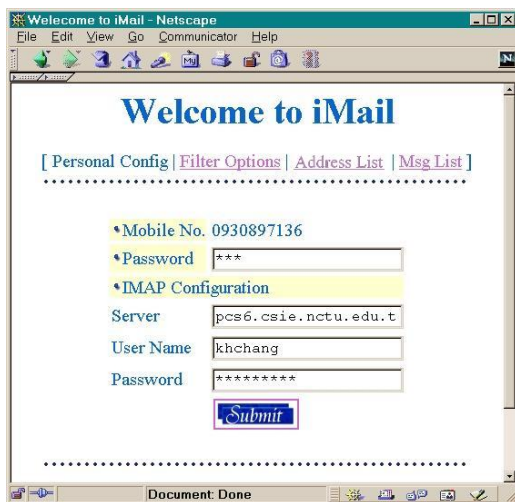


Figure 8: HTML Interface of User Profile Configuration for Users

To access the configuration part from a WAP handset, the user selects “* Change Profile” from the iMail main menu (Figure 6 (a)). The profile configuration part will prompt the filter options defined by the user, and return a menu as shown in Figure 7 (a). We use Sender option as an example to illustrate how a user profile is modified. After the Sender option is selected, the sender list is prompted as shown in Figure 7 (b). The user may delete a sender by unmarking it, or use “Add” to add new senders in the sender list (Figure 7 (c)). Similarly, Figure 7 (d) and (e) illustrate the Date operation. After the user completes modifications, he/she chooses “Submit” from the Options screen (Figure 7 (f)) and the profile configuration part will update the user's configuration. Then the WAP browser returns to the iMail main menu.

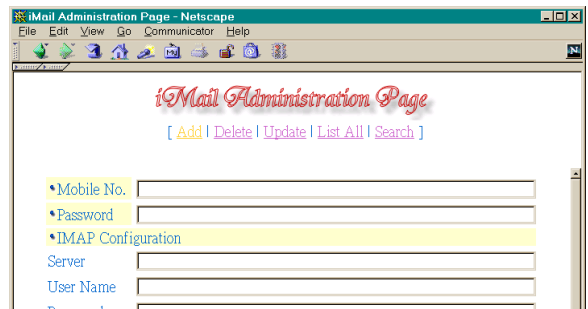


Figure 9: HTML Interface of User Profile Configuration for Administrators

4 iMail Features

Most email access applications are designed for desktop or laptop devices, which have large screens, keyboards, and flexible mouse inputs. On the other hand, accessing emails through a WAP handset is usually restricted by the handset's user interface capability. To provide efficient and user-friendly email access in WAP, we propose several techniques to reduce the wireless environment limitations, including mail filtering, pre-defined list reordering, and two-phase fetch.

4.1 Mail Filtering

The goal of mail filtering is to reduce the number of mails sent to a WAP handset. A user may tailor the filtering options so that only specified mails are delivered to the WAP handset. Five iMail filter options are defined.

Sender option specifies the mails to be sent to a WAP handset based on the addresses of senders. If no mail sender is specified in this option, the WAP user accepts mails from all senders.

Receiver option enables iMail to select mails that match the receiver's email address. For example, consider a user with mail address jenny@csie.nctu.tw. Suppose that the user is also included in mailing group news@imail.com.tw. If the user specifies news@imail.com.tw in the receiver option, then the WAP handset only receives mails sent to news@imail.com.tw, and no mails sent to mary@csie.nctu.tw will be forwarded to the WAP handset.

Subject option accepts mails that contain one or more key words in the subject. For example, if the user would like to receive mails with subjects containing words “HELP” or “PLEASE”, the Subject option is specified as “HELP, PLEASE”. The iMail system only forwards mails whose subjects match one of these words to the WAP handset.

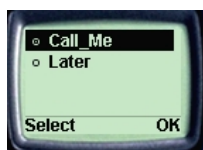
Size option specifies the maximum size of the mails sent to the WAP handset. Mails with larger sizes are

truncated before they are delivered through the wireless network. The default maximum mail size is 500 bytes.

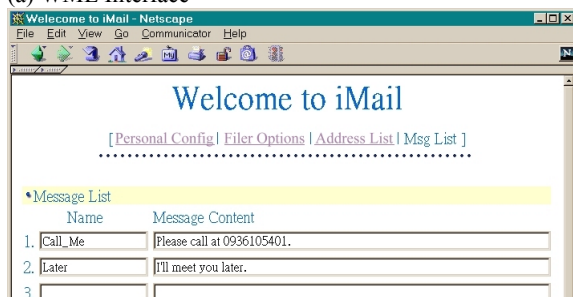
Date option specifies the latest date of mails to be retrieved. There are four alternatives: new, today, three days, and one week. The default value is “new”, which means that all unread mails will be sent to the WAP handset. The iMail date option screen is shown in Figure 7 (e).

4.2 Pre-defined Lists

iMail provides a web interface for users to pre-type frequently used messages (Figure 10 (b)) and e-mail addresses.



(a) WML Interface



(b) HTML Interface

Figure 10: Pre-defined Message List

Pre-defined message list is a collection of messages pre-typed by the user. Every pre-defined message content is identified by a “name”. When a WAP user receives an important message, this user may wish the sender to call his/her handset. In this case the WAP user selects, for example, the message named “Call_Me” in Figure 10 (a), which results in attachment of the content “Please call me at 0936105401” to the mail to be replied to the sender.

Pre-defined address list contains pre-typed destination e-mail addresses. From this list, a user can select email addresses when forwarding or sending a mail.

4.3 Pre-defined List Reordering

To provide convenient menu selection, iMail service part reorders the items in the pre-defined message list and the pre-defined address list based on the access frequencies of the items. In our approach, the latest-accessed or frequently used email address becomes the first item of the menu. The

message list is reordered when the user forwards or replies a mail, while the address list is reordered only when the user forwards a mail. The pre-defined list reordering algorithm is a combination of the least recently use (LRU) algorithm and second-chance algorithm [24]. The LRU algorithm keeps the most recently used entries in the front of the message list. On the other hand, the second chance algorithm gives frequently accessed entries a second chance to be kept in the list. Our reordering algorithm combines LRU and second chance algorithm so that the most recently and frequently used item can be kept in the front of the list. This algorithm is used to reorder both the pre-defined address list and the pre-defined message list. We use pre-defined message list to describe the reordering algorithm as follows.

The access patterns of the pre-defined messages are recorded in a history table. Each entry of the history table contains two fields: `message_name` and `reference_count`. The `message_name` is the name of the message content that the user selected to attach with the mail. The `reference_count` is a flag that indicates if the entry has been accessed more than once. For the first access, the `reference_count` is set to zero. If the entry is accessed again, the `reference_count` will be set to one. The following steps are executed to reorder the history table when a message name is selected.

1. If the `message_name` already exists in the history table, set the `reference_count` to one. Move the entry to the top of the table, and exit the procedure.
2. If the `message_name` does not exist and the history table is not full, then create an entry for the message with a zero `reference_count`. The procedure exits after the new entry is inserted as the first entry in the history table.
3. If the history table is full, then select the oldest entry with a zero `reference_count`. If no such entry exists or if the entry is in the first half of the table, the last entry is selected. Then delete the selected entry. The storage of the deleted entry is reclaimed to accommodate the new entry that is inserted at the top of the table. Set the `reference_count` of the oldest entry to zero and exit the procedure.

4.4 Two-Phase Fetch

In the early stage of iMail development, mail subjects and contents are grouped into one document. If there are too many mails in the user mailbox, users will experience long transmission delay and unhandy navigation. This issue can be resolved by two-phase fetch. As its name suggests, this approach fetches data in two phases. In the first phase, only the subjects of the mails in the user mailbox are sent to the WAP handset. If the user decides to access the mail body, then the requested content is actually delivered to the WAP handsets in the second phase. Our experience indicates that

two-phase fetch effectively eliminates transmission of non-necessary mail bodies.

5 iMail Implementation

This section discusses implementation issues for iMail user interface, iMail and mail server interaction, and so on.

5.1 Handset Capabilities

Due to the length limitation of WSP service data unit, the page size of an encoded WML document is restricted to 1400 bytes [11]. In the case of Nokia 7110, the maximum encoded WML page size is 1397 bytes [27]. Over-sized pages may result in errors and cannot be displayed. Other WAP handsets, such as Ericsson R320/R380, can receive encoded WML pages more than 3000 bytes [29]. Most WAP handsets can view around 4-5 lines of ASCII characters or 2-3 lines of Chinese BIG-5 characters at a time. In the current iMail implementation, the text of a message is bounded to 20 lines (or 4-5 displays). Over-sized messages are truncated.

Various WAP handsets may have different micro-browser capabilities. The WML tags supported by each micro-browser may differ and the same WML page may not appear or behave in the same way. For example, Ericsson handsets support various fonts such as ****, **<big>**, and **** while Nokia 7110 does not. [30] compares the micro-browser for various wireless data-enabled devices. [30] shows the WML tags supported for various WAP handsets.

```
1 <card id="READ" title="Sample Mail">
2   <do type="options" label="From">
3     <go href="#mFrom"/>
4   </do>
5   <do type="options" label="To">
6     <go href="#mTo"/>
7   </do>
8   <do type="options" label="Subject">
9     <go href="#mSub"/>
10  </do>
11  <p> This is a sample mail. </p>
12 </card>
```

Figure 11: A Card Contains Multiple `<do>` Elements with Type Options

Consider another example in Figure 11, where a WML card with title “Sample Mail” contains three `<do>` elements of type `options`. When a user presses the `Options` key, Nokia WAP 1.1 browser displays all the `<do>` elements. On the other hand, in Ericsson R320’s micro-browser, only the first `<do>` element appears on the screen and the remaining `<do>` elements are ignored. To resolve this issue in iMail, multiple `<do>` elements in lines 2-10 of Figure 11 are implemented in another card “Mail Info” (see lines 7-13 in

Figure 12). The original “Sample Mail” card is modified (see lines 2-4 in Figure 12), which only contains a hyperlink to the “Mail Info” card. When the user presses the `Options` key, micro-browser will link to the “Mail Info” card.

```
1 <card id="" READ" title="Sample Mail">
2   <do type="options" label="Mail info">
3     <go href="#mOption"/>
4   </do>
5   <p> This is a sample mail </p>
6 </card>
7 <card id="mOption" title="Mail Info">
8   <p>
9     <a href="#mFrom">* From</a><br/>
10    <a href="#mTo">* To</a><br/>
11    <a href="#mSub">* Subject</a><br/>
12  </p>
13 </card>
```

Figure 12: A Card Contains One `<do>` Element with Type Options

Based on the above discussion, one potential research direction is to build a WAP development environment that automatically tailors WML documents to fit various WAP handset models. Then the WAP application server (e.g. iMail) can utilize the HTTP/1.1 environment variable `HTTP_USER_AGENT` to identify the type of a micro-browser, so that appropriate WML pages can be delivered to specific WAP handsets.

5.2 WML Character Set

Two issues of WML character set are identified in the iMail implementation: international language support and special character recognition.

WAP utilizes Unicode to provide multi-language ability. Some character sets can be directly accepted if they are subsets of the Unicode. In iMail, a multi-language WML document may consist of BIG-5 and ASCII characters. The encoding process is required before the BIG-5 characters can be included in the WML document. Specifically, the BIG-5 characters in the iMail message are of the format “&#xUNICODE;”. The prefix token “&#x” tells the micro-browser that the following string is a Unicode in hexadecimal format. The postfix token “;” specifies the end of UNICOD. In run time, iMail utilizes a Perl module `CPAN:Unicode-Map` [28] to convert BIG-5 characters into the Unicode format, and combines ASCII characters with encoded BIG-5 Unicode into a multi-language page.

During the process of translating different character sets into Unicode, special characters such as some characters in MIME (Multipurpose Internet Mail Extension) [23] mails, might cause errors. To resolve this problem, iMail needs to check the characters carefully to avoid run-time errors. Some special characters, such as `@`, `$`, and `&`, may be conflict with Perl’s semantics. Other characters, such as `<`, `>`, and `/`, if appears as normal text characters, may be

considered as special symbols of WML by the micro-browsers, which cause parsing errors.

6 iMail Notification Mechanisms

When a new mail arrives, a notification mechanism is required in iMail so that the user can be immediately notified of the mail arrival through the WAP handset. The iMail users can receive notification message either by GSM short message service (SMS) or by WAP push facility. We consider three approaches to build iMail notification mechanism: polling, forwarding, and relay.

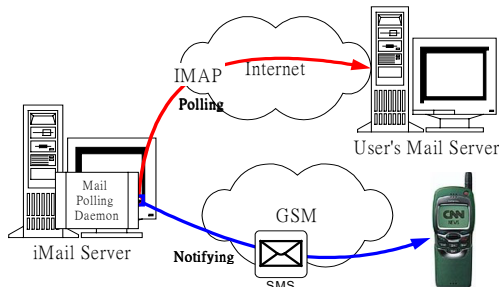


Figure 13: iMail Notification Mechanism: Polling Model

In the polling model (Figure 13), a daemon residing in iMail periodically polls the mail server to check whether the user has new mails in the mailbox. If the new mails meet the constraints of user-specified filters, iMail may inform user by sending a notification message through the GSM network.

In the forwarding model (Figure 14), when the iMail daemon receives a forwarded mail from the mail server of an iMail user, it triggers the mail notification process. Each iMail user needs to set up mail forwarding link from his/her mail system to the iMail system. When a mail arrives at the user's mail system, this mail is immediately forwarded to the iMail daemon. If the mail satisfies the user-specific filter constraints, then the daemon sends a notification to the WAP handset. When the notification procedure is completed, the forwarded mail is either dropped or cached in iMail for further processing.

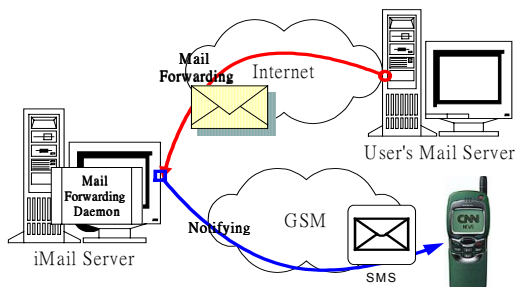


Figure 14: iMail Notification Mechanism: Forwarding Model

In the relay model (Figure 15), a relay mechanism is built in the iMail server (Figure 15 (A)), which has a unique domain name just like *ieee.org* for IEEE. The actual

mailboxes of the users are located at their mail servers (Figure 15 (B)). Assume that the iMail domain name is *imail.com*. When a customer Jenny subscribes to the iMail service, she also obtains an account *jenny@imail.com*, and a forwarded destination, say, *jenny@realmail.com* is specified. Any mail sent to *jenny@imail.com* is delivered to the iMail server (Figure 15 (A)). The iMail server forwards the original mail to the user's mail server (Figure 15 (B)), and the iMail filter extracts some user-specific information from the content of the received mail (Figure 15 (B)). Based on the extracted information, the iMail demon determines if it should send a notification message to Jenny's handset. Upon receipt the notification message Jenny then decides whether she wants to take further actions. This extracted information in iMail is either cached in a database (Figure 15 (C)) or simply dropped after the notification is performed.

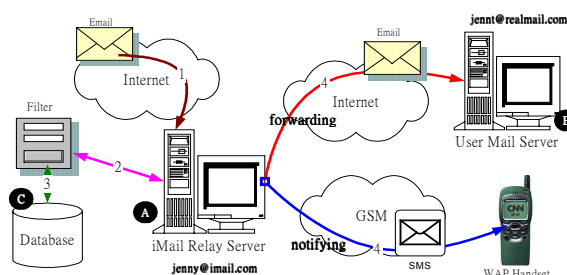


Figure 15: iMail Notification Mechanism: Relay Model

It is obvious that the polling model is easy to implement. However, the polling daemon needs to poll each user's mail server periodically, and this action might not be efficient. When the user population is large, the polling model will experience heavy communication load, which results in poor performance. Furthermore, the notification cannot be done in real time.

In the forwarding model, communications between the user's mail server and iMail server occur when new messages actually arrive and notification can be done in real time. Similar to the forwarding model, the relay model supports real time notification. It also provides users with forwarding email account just as *ieee.org* does to IEEE members. This "permanent" email account helps service providers to attract their customers. In terms of business opportunities, iMail service provider can collaborate with free-email service providers, such as Yahoo and Hotmail, to expand its market opportunities. Furthermore, the forwarding model builds user-specified filters into mail system. In other words, the filter and mail access protocol, such as IMAP4rev1 and POP3, can be merged to support mail-retrieving devices with limited capabilities (such as wireless handheld device).

7 Conclusion

This paper proposed iMail, a WAP mail retrieving system. In iMail, we introduced several features such as two-phase fetch, mail filtering, and predefined list reordering. Our experience indicated that these features indeed improved the

efficiency of iMail system. Moreover, when a new mail arrives, a mail notification mechanism is needed to inform the iMail user immediately. We proposed three approaches to build iMail notification mechanism. This paper also pointed out some design and implementation guidelines for WAP applications.

A potential research extension of our work is to integrate mail and voice translating functionality into the iMail system. Users may utilize voice as the user input/output interface of the iMail system through the WAP handsets.

8 Reference

- [1] "Wireless Application Protocol White Paper", WAP Forum, June-1999. URL: <http://www.wapforum.org/>
- [2] "Wireless Application Protocol Architecture Specification", WAP Forum, 30-April-1998. URL: <http://www.wapforum.org/>
- [3] ETSI/TC. Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Service description; Sate 2 (GSM 03.60 version 7.0.0 Release 1999). Technical Report Recommendation GSM 03.60, ETSI, 1999.
- [4] "Wireless Datagram Protocol Specification", WAP Forum, 30-April-1998. URL: <http://www.wapforum.org/>
- [5] "Wireless Markup Language Specification", WAP Forum, 16-June-1999. URL: <http://www.wapforum.org/>
- [6] "Wireless Application Environment Overview", WAP Forum, 04-November-1999. URL: <http://www.wapforum.org/>
- [7] "Wireless Markup Language Script", WAP Forum, 04-November-1999. URL: <http://www.wapforum.org/>
- [8] "Wireless Telephony Application Specification", WAP Forum, 08-November-1999. URL: <http://www.wapforum.org/>
- [9] "Wireless Telephony Application Interface", WAP Forum, 08-November-1999. URL: <http://www.wapforum.org/>
- [10] "Wireless Application Environment Specification", WAP Forum, 04-November-1999. URL: <http://www.wapforum.org/>
- [11] "Wireless Session Protocol", WAP Forum, 05-November-1999. URL: <http://www.wapforum.org/>
- [12] "Wireless Transaction Protocol Specification", WAP Forum, 11-June-1999. URL: <http://www.wapforum.org/>
- [13] "Wireless Transport Layer Security Protocol", WAP Forum, 05-November-1999. URL: <http://www.wapforum.org/>
- [14] ETSI/TC. Mobile Application Part (MAP) Specification, Version 4.8.0. Technical Report Recommendation GSM 09.02, ETSI, 1994.
- [15] Lin, Y.-B. and Chlamtac, I. Wireless and Mobile Network Architectures. John Wiley & Sons, 2000.
- [16] RFC2060, Internet Message Access Protocol – Version 4rev1; M. Crispin, December 1996.
- [17] Phone.com Software Developer's Kit (UP.SDK) 4.0, Phone.com, 1999. URL: <http://developer.phone.com/>
- [18] SmartPhone Emulator version 1.0 beta, YOSPACE, 2000. URL: <http://www.yospace.com/>
- [19] Nokia WAP Toolkit URL: <http://www.forum.nokia.com/>
- [20] Ericsson WapIDE Service Development Kit URL: <http://www.ericsson.com/WAP/>
- [21] "Developer's Guide" Nokia WAP Toolkit version 1.1. Nokia, JUN-1999. URL: <http://www.forum.nokia.com/>
- [22] "WML Reference" Nokia WAP Toolkit version 1.1. Nokia, JUN-1999. URL: <http://www.forum.nokia.com/>
- [23] "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies" IETF RFC 1521
- [24] Operating System Concepts, Fifth Edition, Abraham Silberschatz and Peter Baer Galvin, 1999.
- [25] Extensible Markup Language (XML) URL: <http://www.w3c.org/XML/>
- [26] Malcolm Beattie, Perl Mail::Cclient module, Version 0.5, University of Washington imapd source distribution, 1999. URL: <ftp://ftp.cac.washington.edu/mail/imap-4.5.tar.Z>
- [27] Nokia WAP developer Forum, FAQ: Nokia 7110 related questions, URL: http://www.forum.nokia.com/wapforum/main/1,6668,1_1_30_3_2,00.html
- [28] Martin Schwartz, Perl Unicode::Map module, Version 0.105, 1998. URL: <http://www.wbs.cs.tu-berlin.de/~schwartz/perl/>
- [29] WAP One discuss forum, URL: <http://www.wapone.com.tw/Layout.asp?Page=Forum&Forum=discussion>
- [30] Wireless data-enabled devices: phones, AnyWhereYouGo.com, URL: <http://www.anywhereyougo.com/ayg/ayg/Content.po?name=wap/Devices>