國立交通大學

電機資訊學院

資訊科學系

博士論文

植基於電子郵件的行動代理人執行環境之研究

A Study on Email-based Mobile Agent Runtime Environment

研 究 生　：何敏煌

指導教授　：袁賢銘 博士

中華民國九十三年六月

植基於電子郵件的行動代理人執行環境之研究

A Study on
Email-based Mobile Agent Runtime Environment

研究生：何敏煌　　Student: Min-Huang Ho

指導教授：袁賢銘 博士　　Advisor: Dr. Shyan-Ming Yuan

國立交通大學
電機資訊學院
資訊科學系
博士論文

A Dissertation Submitted to
Department of Computer and Information Science
College of Electrical Engineering and Computer Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

in
Computer and Information Science
June 2004
Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

植基於電子郵件的行動代理人執行環境之研究

研究生：何敏煌　　　　　　指導教授：袁賢銘 博士

國立交通大學資訊科學研究所

摘要

現今，電腦硬體和軟體技術在網際網路的應用上其發展非常地快速。應用這些新的軟硬體技術在資訊擷取都已經獲得了相當大的成功。因為在網路上從事資訊擷取的應用非常地重要，在此方面的研究已經有相當多的成果。然而，目前這些資訊擷取方面的研究成果，主要大多是假設其使用的作業平台都具有穩定且高速的網路環境和強大的運算能力，對於那些網路連線速度慢且不穩定、運算能力薄弱且電源供應和相關資源都缺乏的行動運算設備和環境來說，要應用這些研究成果是非常困難且不容易的。

本論文的目的即在於廣泛地研究這一類資源缺乏的行動計算環境在資訊擷取及相關技術方面的問題，並提出一個可行的架構以解決這些問題。在論文中，我們考慮所有移動裝置使用者在從事資訊擷取時可能遇到的狀況，然後提出一個建構於有限資源計算能力和低品質網路移動式設備上的工作架構，讓使用者可以在此類網路上擷取出他所需要的有用資訊。在我們的設計中，使用者可以輕易地利用他能上網的移動設備在區域網路及網際網路上擷取他有興趣的資訊，而這些移動式設備可以包含行動電話、PDA、甚至是掌上型的筆記型電腦。由於這些裝置在操作行為模式及介面上非常地多樣化，所以我們使用標準的電子郵件通訊協定讓使用者可以用一致性的操作介面在區域網路及網際網路中擷取資訊。

電子郵件是所有可以上網的裝置中最受歡迎的網路服務之一，使得電子郵件的協定和其客戶端操作軟體在不同的可上網裝置中具有一致性。也因此，透過電子郵件來做為資訊擷取的媒介也將被一致化。也就是說，使用者可以利用一致性的介面和操作方式在我們設計的工作架構下設計、和操作一些在伺服器端的可移動式的行動代理程式，而這些行動代理程式即可在具有良好網路環境及高速計算能力的遠端伺服器中為使用者高速地擷取資訊及分析資料，不用去擔心他現在使用的是那一種行動裝置。

A Study on Email-based Mobile Agent Runtime Environment

Student: Min-Huang Ho Advisor: Dr. Shyan-Ming Yuan

Department of Computer and Information Science

National Chiao Tung University

ABSTRACT

Currently, the technologies of hardware and software about the Internet and information processing are developed in a very high pace. Applying these new technologies into the area of information retrieval is very successful. Because of the importance of information retrieval via the network, there are lots of prior studies dedicated in these related topics, and they have already got plenty of successful outcomes, but most of those researches always assume that the platforms by means of those technologies have good quality networks and high computing power CPU. For those poor quality networks, low computing power CPU, and resource-constrained mobile devices, obviously, most of the previous research outcomes usually can’t be easily adopted.

The objective of this dissertation is to study such resource-constrained mobile computing circumstances for information retrieval and related applications, and to propose a feasible framework to solve such situation. In this dissertation, we take care all situations that a mobile user may run into, and then propose a framework for the mobile devices with low-computing-power and poor-quality-networks to retrieval the information he is interested in from the network. In our design, a mobile user can retrieve the information from the network or the Internet via his/her networked mobile device. The mobile devices may include mobile phones, PDA, or mini-notebook PCs. Because the interface and operating model of those mobile devices are very diverse, to uniform the user interface among these devices, we utilize e-mail protocol and mobile agent technologies for the user to retrieve the information from the network or the Internet.

Because the e-mail is one of the most popular applications for all networked devices, and the e-mail protocol and e-mail client operation are common and uniform in those networked devices, the information retrieval operating interface will be unified. That is, the user may utilize the unified interface of our proposed framework to design, operate, and execute some mobile agents in the server-side, and those mobile agents will retrieve the information for the user in a very good computing environment. The user doesn’t need to worry about what kind of the mobile device he/she will use in the whole information retrieval process.

Keyword: Mobile Agent, Information Retrieval, Mobile Computing, E-mail

誌謝

能夠順利取得博士學位，要感謝的人實在太多了，首先是我的指導教授袁賢銘博士，在漫長的研究過程中，如果沒有您全力的指導和充份的信任，要完成博士論文及相關研究工作，對我來說幾乎是不可能的任務。我的良師益友，台北大學教授張玉山博士，由於你適時的指引和關心，對於我的論文研究方向及內容不斷地加以斧正，讓我的博士研究內容得以更加地充實且切合要旨。

當然，我也要感謝交大資科所各位教過我的教授們，在我研究的過程中給我很大的幫助，尤其是楊維邦館長和孫春在教授，一路從論文計畫審查、校內口試到學位考試，都能適時給我重要的指引，並給予鼓勵和建議，讓我的論文研究內容更加地實用，且不至於蕪蔓龐雜。施仁忠系主任在校內口試時給予的建議，適切地指正了論文部份的缺漏，讓我能夠及時地補正。中興大學資管所朱延平系主任和成功大學資工所鄭憲宗教授不辭辛勞地幫我審閱博士論文，並於學位考試時給我建議和鼓勵，在此一併向你們致上最高的敬意和謝意。

從台南每個星期通車到新竹求學研究的漫長過程中，多虧了分散式系統實驗室的各位學長學弟們的協助，才得以安然渡過研究過程的各個難關。焦信達、葉秉哲、許瑞愷、邱繼弘、蕭存喻、鄭明俊、高子漢、吳瑞祥、孫文駿、及其他的學弟們，謝謝你們。

我的妻子－美堯，從在大學的時候認識妳，我就一直是個只知道讀書的窮學生，如今，總算得以結束學生的身份，在家裡專心當個好老公了。回頭想想這段走過的路，如果沒有聰慧的妳從旁協助及打理，今天我可能還是一個留在台北獨自過生活的傻小子，謝謝妳。女兒小晴，兒子小暘，你們這兩個寶貝蛋，爸爸終於可以專心地陪你們了，從你們出生到現在，你們的爸爸都好忙好忙，現在，爸爸可是你們兩個人專用的喔！

我的母親張玉英女士，希望我這樣一點點的努力成果能夠稍稍彌補您在我身上的用心；媽媽，您辛苦了。

最後，謹將本論文獻給已故的父親，沒能讓您親自參加我的畢業典禮，是我這一生中最大的遺憾。

Contents

List of Figures

Chapter 1 Introduction

Chapter 2 Problem Description

Chapter 3 Background and Related Works

Chapter 4 Mobile Agent and Integrated Information Retrieval

Chapter 5 System Design and Implementation

Chapter 6 Experiments and Examples

Chapter 7 Applications

List of Tables

Chapter 1

Introduction

This chapter gives an introduction about the trend and the vision of the mobile computing environment, the idea and efforts in our researches will also be presented here. At the first, we point out the developing trend of the information retrieval technologies and mobile computing technologies nowadays, and then focus the encountered problems for the mobile users who are interested in retrieving the information on the network via their mobile devices. To solve the mentioned problems, we propose a novel and feasible solution. A brief introduction of our approach will be given in section 1.2. In section 1.3, we also list the current challenges in the related topics, and these issues will be solved in the following chapters of this dissertation. The final section of this chapter is a description about the organization of this dissertation.

1.1 Problem basics

The number of documents on the Internet is very tremendous nowadays, and the document numbers are also growing in a very fast speed. Thus, the Internet has become a very important information source for every information retriever or even every computer users. However, not only the volume of documents on the Internet is very huge, but also the formats of those documents are very diverse. Furthermore, because of lacking the proving process by editors, most of the documents on the Internet do not have a well structure, and those documents always lack a specified compact and precise description. Consequently, the technologies dealing with efficiently retrieving the information from the great amount diverse unstructured documents on the Internet are very urgent for the people who want to get more comprehensive and precise results from it.

For approximately 4000 years ago, people had already known to organize the information for later retrieval and usage, but today, facing to those almost countless documents, we need some useful and efficient technologies to help us. Information retrieval (IR) deals with the representation, storage, organization of, and access to information items. The representation and organization of the information items should provide the user with easy access to the information in which he is interested. In the computer centric view, the IR Problem consists mainly of building up efficient indexes, processing user queries with high performance, and developing ranking algorithms which improve the quality of the answer set.[1] Traditionally, libraries are the major information sources which the IR researches focused on. The digitized documents of digital libraries' management system have well defined structure and semantics. Thus, it is easy and efficient for an IR client to get the user's interested information from such well structured information sources. However, the documents of the libraries are not the only information source for the people to gather. As mentioned earlier, the information sources from the Internet or private local area network are more and more important nowadays. Thus, the related issues accompanied with retrieving the documents from the Internet have to be envisaged and be solved.

Currently, given the ever-increasing scale and diversity of information and applications on the Internet, improving the technology of information retrieval on the Internet is an urgent research objective. The documents retrieved from the Internet are either semi-structured or unstructured in format and its sources are extremely heterogeneous. In consequence, the task of efficiently gathering and extracting information from documents can be both difficult and tedious. Given this variety of sources and formats, many choose to use mediator/wrapper architecture [11], but applying these IR technologies also means that the information retrieval and gathering works need a high speed CPU and a good quality network environment.

Some reports predicate that there will be more than 2 billion mobile users all over the world between 2007 and 2010. That is, there is one mobile phone for every three living persons on the earth. Now, many people are familiar to use various kinds of portable terminal devices, such as PDAs or smart phones, to access the information on the Internet at any place in any time to assist their daily businesses. Some people also utilize some sort of agent technologies of desktop devices to help themselves taking cares of their daily affairs. Moreover, personal mobile agents can help them to search, to filter, to analyze, and to translate the large volume information of the Internet. In most cases, utilize portable devices to connect to the information source has to make use of wireless network. As the previous search [44] estimates, the trend of Internet connecting technologies is shown in Figure 1.1.

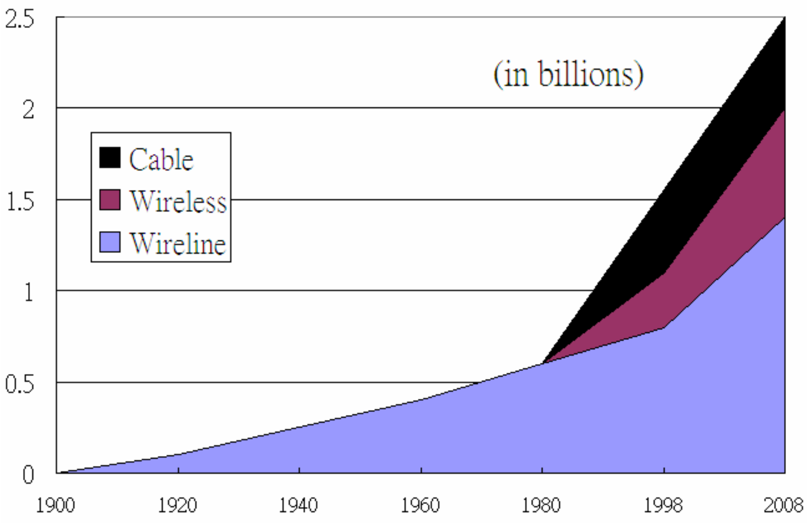


Figure 1.1: The ratio of the Internet among cable, wireless, and wireline.

As Figure 1.1 shows, in the near future, the wireless technologies for connecting to the Internet and private local area networks will be developed in a very great pace, and its percentage is large among all three main network technologies. Some reports even estimate that wireless connection will replace all the wired connection in near future. As well as the development of wireless networking technologies, people will also familiar with connecting to the Internet by multiple devices. Figure 1.2 shows the trend of Internet users. [44]

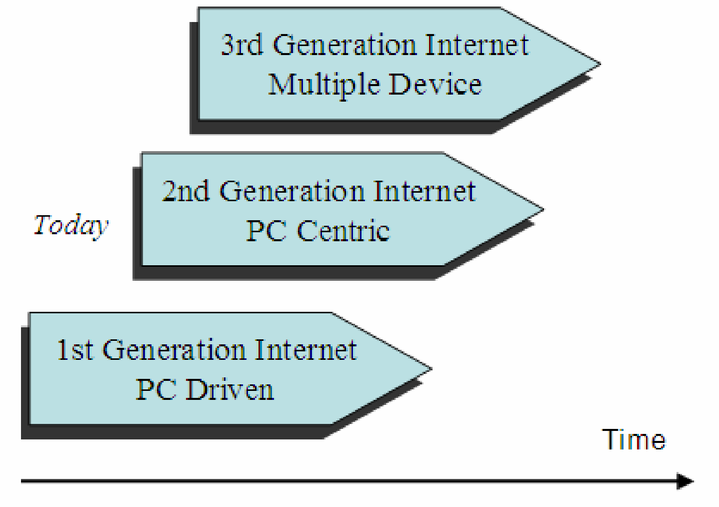


Figure 1.2: The trend of Internet users.

Today, most people mainly connect to the Internet by PC, and retrieve the information on the Internet to his mobile device via the application of the PC (such as the services from AvantGo). It is a PC centric Internet connection model. However, according to the fast development of wireless technologies, it is foreseeable that more and more mobile devices will be equipped the high power CPU and stable network connection. There will be more and more users are willing to utilize their mobile devices directly connecting to the Internet. And the services and applications on the mobile devices will be more and more fantastic and powerful. Needless to say, the software agent technologies and frameworks also can be easily applied into the mobile computing environment, but before the coming of the 3rd Generation Internet which is shown in Figure 1.2, there are some considerations have to be considered for providing more powerful applications on the mobile devices. The software agent technology is one of the savers.

In the viewpoint of our research, there are two major developing trends of software agent technologies. The one is the researches about the enhancement of the autonomy and intelligence, and the other is the enhancement of the mobility and accordance of the agents. The highly autonomous and highly intelligent agents can accept the delegation ordered by the user to execute his specified works intelligently and autonomously. The former kind of agent technology is focused on the intelligence and autonomous for process and analysis of collected information from the network. In such case, the agents always stay on the same machine or host, and retrieve the information via the network. The processed information is always returned to the user, and let him to make the decisions, but in many agent frameworks, to those more intelligent agent systems, they even can autonomously make some decisions for their delegates if they have the user's appropriate authorization.

The model of traditional static agent for doing the user's delegated works is depicted in Figure 1.3. As shown in Figure 1.3, the user or client may execute a buyer's program on the buyer's computer. Through the slower Internet communication (compared to the inner I/O speed of computer), the program always directly retrieve the great volume of information and documents from supplier's programs or services on the remote supplier's computer for further analysis and making decisions. If the retrieved documents are large enough, the process will consume lots of times.

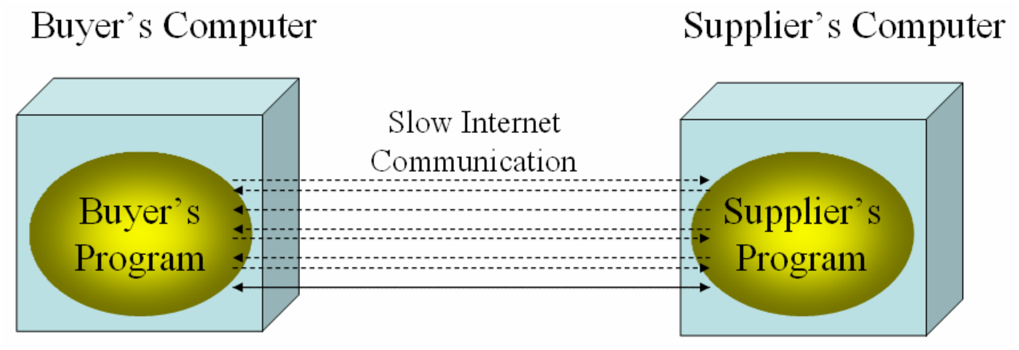


Figure 1.3: The model of traditional static agent for doing some user's delegated works.

The technologies of mobile agents are mainly devoted to solve the previous mentioned situation. Consider the picture shown in Figure 1.4. It is the same situation with Figure 1.3, but this time, the user delegates a mobile software agent, and sends it directly to the remote supplier's computer. In such case, all the works, includes retrieving, analyzing, and processing of documents, will be completed in mobile software agent on the remote supplier's computer. The data through the Internet communication are only the results and the mobile software agent itself. Lots of times for transferring the unneeded and temporary documents will be saved

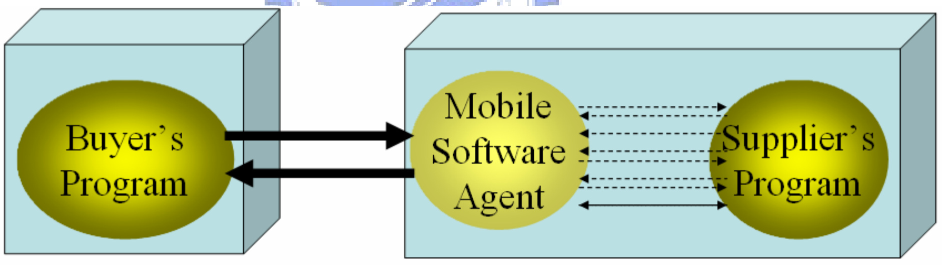


Figure 1.4: The model of mobile agent for doing the same works in Figure 1.3.

Applying the mobile agent technologies to the IR problem has many advantages. Consider the situation, if a user has to fetch all of the employee data from the remote database server, and then analyze and mining these data to find the some interested statistic trend. Using the traditional model has to fetch all the data items across the network into local machine. If the employee data is huge and is distributed across several database servers, the data fetching works will take very long time, and will waste lots of memory spaces of the local machine to store them. Of course, in the whole fetch process, the network connection has to be kept in a very high speed and good quality condition. If the database servers are distributed across many buildings or even several different cities, it will cost lots of money for temporary documents transmission.

In such case, the quantity of data items is much larger than the size of processing program agent itself. Thus, we can transfer the agent into each database server respectively, and then execute the agent's program code to analyze the data in the remote data server. Consequently, it can save lots of network transmitting time and refine the utilization of local machine's memory space. This is the key benefit of mobile agent technologies. Utilizing the mobile agent technologies, all the remote-accessed data items will be acted as local-accessed data. Thus, not only the processing time will be reduced dramatically, but the cost for data transmitting will also be reduced too. Furthermore, the network connection is no more needed at most of the time during the data processing. Currently, the mobile agent technology is feasible for desktop computing environment, and it is more and more popular for many information retrieval projects.

Most of the mobile devices are the resource-constrained devices. It is not easy for such limited devices to execute too complex works. The battery duration also limits the execution time for any software on the mobile devices. Accordingly, the features of mobile agent technology are very suitable for the mobile devices to do some useful and long-term information retrieval projects. However, applying the mobile agent technologies to mobile devices also has some considerations.

The traditional mobile agent system is always an on-line system. That is, no matter what the processing node is at local or at remote, they have to keep the network connection available, especially for the host in the server-side. In the on-line mobile agent system, the host always has to be available on the network and wait the client's request. For the desktop PC and server, this is an acceptable solution, but for a mobile computing environment, considering the networking cost and networking stability, it is not easy to fulfill such requirements. As mentioned earlier, it is important to propose a feasible design to conquer the issues produced in mobile computing environment for retrieval information on the Internet.

For the technologies of mobile agent system, first of all, the key point we have to emphasize is that the off-line mobile agent model is very important than the on-line one in the mobile computing environment. The on-line mobile agent always stays at a device with well networking and high computing power. It is easy and fast for such on-line mobile agent to migrate itself from one host to another. The off-line mobile agent is different. It always stays on the mobile device with unstable networking and low computing power. The cost of wireless networking for mobile devices is also more expensive than the desktop's service.

The target of our approach is to design a complete framework. The solution has to minimize the utilization of expensive wireless networking for mobile devices, but keeps the flexibility of mobile agents. The framework not only provides on-line mobile agents on the host with well networking environment, but also provides off-line mobile agents on the mobile devices with poor networking environment and low-computing power. The user can utilize his mobile device (mobile phones, PDAs, or even pocket PCs and notebook PCs) to launch some mobile agents, and let those agents assist him to retrieve his interested information from the private database or the Internet by any e-mail client applications. For the purpose of information retrieval, those mobile agents will migrate from one host to another. It also can migrate itself to another mobile device as well.

1.2 Objectives and our contributions

According to the explanation in section 1.1, the objective of this dissertation is to propose a feasible solution for the user to comprehensively access the information on the Internet or the local area network by the resource-constrained mobile devices. Since the e-mail is almost the most popular service for any network connected devices, it will provide a unified operating interface to the user for accessing the Internet. In this dissertation, we propose a framework named Email-based Mobile Agent Runtime Environment (e-MARE). Utilizing the instructions provided by e-MARE framework, the user can easily operate the mobile information retrieval agent embedded in the main server to retrieve the interested information from the Internet or private databases via his mobile devices. Not just the embedded mobile agent of the server, the user can design his own mobile agent at their mobile device, and then launch that agent to the remote host for his own purpose.

The main contribution of this dissertation is the proposed email-based mobile agent and information retrieval application operating framework. We implement a prototype system named e-MARE. Through the framework, no matter what the device is, and no matter where the user is, the user can use the unified user interface to design, launch, and operate the mobile agent to achieve his own purpose. Because of the off-line and asynchronous characteristics of e-mail protocol, the mobile agent has more time to retrieve and analyze the documents fetched from the network and the Internet. The system may prepare some results with good quality as an e-mail, and then send it to user's mailbox. Under this framework, the user can do anything just as if he is using the desktop PC with good networking environment, even though he is just using a mobile device with poor networking environment and poor computing power.

Finally, to demonstrate the feasible of e-MARE, we provide many examples and useful applications in later chapters of this dissertation.

1.3 The challenges in our research

Although we use the simple e-mail protocol as the interface between mobile users and servers, due to the original difficulties of information retrieval for complex and versatile documents on the Internet, there are still some difficulties have to be considered in our proposed framework. The challenges for design and implementation of our framework are briefly shown as follows:

- Most of the documents of the information source on the Internet are semi-structured or even unstructured, they are hard to parse and analyze completely by application program.
- Different information source has different protocol for getting the documents. It needs some mature wrapper technologies to solve the diversity among several information sources.
- Each mobile device has its distinct user interface and applications. For client side, the proposed framework has to adapt all the client devices, let all devices can easily operate the mobile agent with our proposed framework. For server side, the result e-mail has to adapt itself to fulfill the different requirements and environment of that specified mobile device.
- For completely operating the local and remote mobile agent, the system also has to provide a complete instruction set or the programming language solutions for user. How to install, or compile and execute these commands and programming languages are also a great challenge of our design framework.
- Furthermore, for some mobile devices (such as mobile phones) with limited input interface, it is hard to input too many characters in a short time.
- The security issues are always the most important consideration for the

users and system developers. In our research, we applying the id/password mechanism to avoid some malicious users. Due to the advanced security issues are the future works of our approach, we just exclude these topics in the dissertation.

1.4 Organization of this dissertation

This dissertation is organized as follows. Chapter 1 gives a fundamental introduction including introductions, objectives, contributions, and challenges. Chapter 2 gives some discussions about the proposed problems and motivations more detail. Chapter 3 surveys the background and previous works related with the topic. In Chapter 3, we also survey the present and future technologies related to our research more detail. Chapter 4 mainly introduces some theories of mobile agent technologies, and their applications on the information retrieval. In Chapter 5, we propose the system design and discuss the issues derived from the design. We will evaluate some real-world examples in Chapter 6. Some more complex and useful applications which are applying the e-MARE framework are extensively introduced in Chapter 7. Finally, the concluding remarks are given in the Chapter 8.

Chapter 2

Problem Description

This chapter describes the related issues and technologies of our research. At the first, we review many obstacles about utilizing the mobile devices to retrieve the information on the Internet. Then, some related technologies, including agent technologies, mobile agent technologies and information retrieval technologies, for improving the whole process of information retrieval are briefly introduced in the following sections. Finally, we classify the type of information retrieval models, and state the advantages for applying the e-mail protocol and related technologies to those mobile devices information retrieval approaches.

2.1 Information retrieval on the resource-constraint mobile devices

Information Retrieval (IR) is one of the most important applications of computer science nowadays. The traditional IR is mainly focused on the well structured digital library database and management system. However, today, the Internet provides the user a cheaper and easier way to access various data sources, and the advances in all kinds of digital communication provided greater access to networks. Moreover, the Internet also provides anyone an easier way to publish their own documents to the web as one of the data item of the information source. Therefore, the information on the Internet becomes a very important but unstructured and not easy manageable information source for information retriever. That is, the user is now facing the astronomical unstructured documents on the Internet.

There are many search engines on the Internet which indexing lots of documents, and providing the users to query their interested information, but there is no search engine can cover all documents on the Internet. Even the most popular search engine, Google (http://www.google.com), has been pointed out that its estimated coverage is about 45%-55% until year 2002 [53]. Hence, many researches develop meta-search technologies, trying to cover more and more Internet documents. The basic concept of meta-search technologies is shown in Figure 2.1.

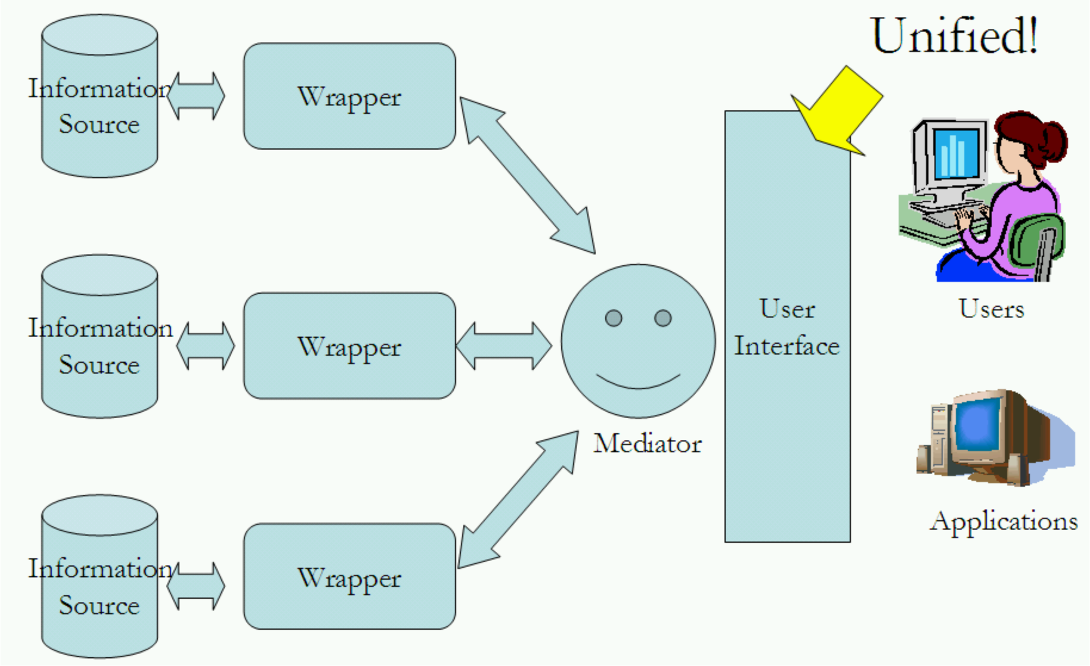


Figure 2.1: The basic concept of meta-search technologies.

As Figure 2.1 shows, a good organized meta-search system integrates several information sources to provide a unified user interface, letting the users or applications to access those information sources in a unified and integrated manner. The users or the application developers don't need to worry about the format of querying string at all. However, the diversities and differences of the terminal which is used by the client user or application are not considered in these approaches.

For more and more greatly increasing mobile device users, to provide a unified user interface for them to access those versatile information sources is urgent. Figure 2.2 shows the basic idea of such idea. As shown in Figures 2.2, one of the main objectives of our research is to contribute such a unified user interface mobile information retrieval framework. The detail of such framework will be explained in the following chapters of this dissertation.

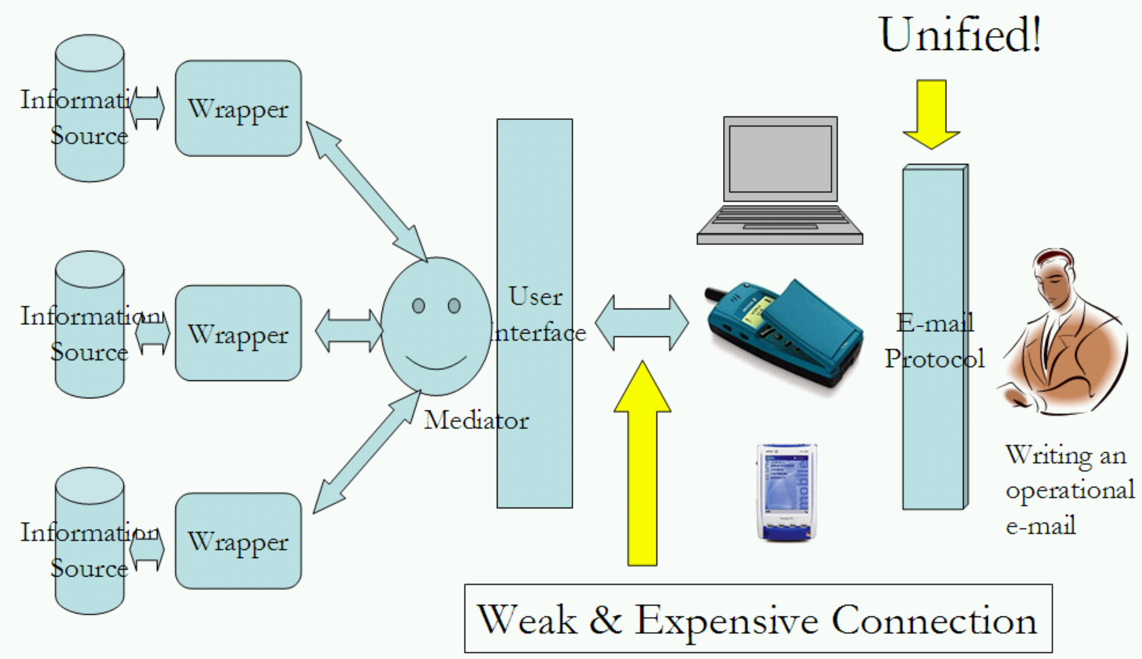


Figure 2.2: The unified user interface for the mobile device users.

Beside the consideration of the great number of documents on the Internet, how to efficiently fetch the useful information from the astronomical documents is the most important issue for many Internet users and information investigators. There are lots of former researches were dedicated on related topics, and they also have got lots of successes. There are also lots of products with these technologies and serves millions of users. However, most of the traditional approaches mentioned above assume that the client user is sitting at the front of a desktop PC, and the PC is also equipped some sort of well networking connection. All the information retrieval works need a powerful computing environment to achieve the information retrieving requirements. For more and more mobile users using mobile device with limited computing power, those traditional methods will fail. The major difference between the desktop PC and the mobile device are the limitation of battery power, computing ability, network speed and quality, and other computing resources. These differences will dominate the technologies it will use for retrieving information from the Internet.

Different to the desktop platform, most of mobile devices are equipped with low-speed CPU, fewer memory, smaller display size, and low-speed and unstable network connection. Most of the traditional information retrieval technologies are not suitable for such limited mobile devices. Table 2.1 compares many popular mobile devices with notebook and desktop PC.

Table 2.1: The comparison between mobile devices and PC.

Devices	CPU/OS	Memory/Networking	Display Resolution
SonyEricsson T68 	N/A	800KB RAM GPRS IrDA BlueTooth	101x80 256 colors
Nokia 3650 	ARM9 CPU 104 MHz Symbian OS 6.1 series 60	4MB MMC supported GPRS IrDA BlueTooth	176x208 4096 colors
Acer s10 	Motorola DragonBall VZ 33MHz Palm OS 4.1	16 MB SDRAM 8MB Flash ROM Memory Stick supported IrDA Serial MODEM	160x160 16 gray colors
Acer n10 	Intel PXA255 300MHz Microsoft Pocket PC 2003	64MB SDRAM 32MB Flash ROM SD/MMC supported CF Type I/II supported IrDA GPRS IEEE 802.11b	240x320 3.5“ TFT LCD 65536 colors
Acer TraveMate C300 Tablet PC 	Intel Pentium M 1.7GHz Microsoft Windows XP Tablet Edition	256MB (2GB) SDRAM 40GB Hard Disk CD-ROM/DVD-ROM IrDA 10/100/1000 Ethernet IEEE 802.11b 56K MODEM	1024x768 14.1” TFT LCD 16.7 million colors
IBM ThinkCentre M 	Intel Pentium 4 2.8 GHz Microsoft Windows XP Professional	256MB SDRAM 80GB Hard Disk DVD-ROM 10/100 Ethernet IEEE 802.11b 56K MODEM	1024x768 15” TFT LCD 16.7 million colors

Therefore, to reduce the gap between these two kind of devices, it may have two possible solutions for a mobile device to retrieve the information on the Internet, 1) downsizing the scale of these technologies and executing them on the resource-constraint devices, or 2) executing those on the server-end, and leaving the mobile devices as the simple front end for the user. In our survey and research, later choice is a better solution to adapt the most mobile devices for information retrieval.

2.2 Styles of information gathering

For the daily information gathering works of Internet users, we divide those works into two major styles: *On-line Information Gathering* and *Off-line Information Gathering*. We call an information gathering style *On-line* if the user is utilizing the WWW browser or any other WWW surfing applications to browse and fetch the documents on the Internet continuously. In general, the users of this type are always contiguously initiate some query commands or operations to some sort of information sources, such as search engines, on the web, sitting on the chair in front of computer, and waiting the results be displayed on the screen. Because the user is always waiting the results after he has issued his requests, the results have to be returned as soon as possible. To return the query results to the users faster, it means, there is no enough time for the information retrieval application (in this case, it is some sort of web browser or web surfing application) to provide the complete results which have been completely analyzed from the fetched documents on the Internet. In the meantime, the technologies for the time-restricted information retrieval processes are always utilize the simple keyword-based technology to reduce the response time for information retrieval, or the information provider has to completely analyze the documents in advance.

On the other hand, we call an information gathering styles *Off-line* if the user doesn't care the response time from the Internet or network after he has issued his queries. That is, in most cases, the most important consideration of those users is the completeness and correctness of the result sets. They would rather spend more time to wait the complete results from the information retrieval application. Thus, the information processing application has more time to parse and analyze the fetched documents, and to rank the display order for the user according to the specified preferences and other important arguments of the user. Compare the style to *On-line Information Gathering*, the user always can get more comprehensive and more precise results from the Internet, but the cost which the user has to pay is more time to wait.

In the traditional information retrieval working environments, spending more time to wait the results from the Internet is always not acceptable for most Internet users. However, in the unstable network with low-speed CPU mobile computing environment, such time waiting situation is common and acceptable for most mobile users. Furthermore, most of the mobile devices can not efficiently execute full function Internet browser to directly surf the web documents on the mobile devices for the user, it is hard to utilize *On-line Information Gathering Style* on the mobile computing environment. Consequently, utilizing the e-mail based solution, *Off-line Information Gathering Style*, on the information retrieval from the Internet for mobile computing environment is acceptable and feasible.

2.3 Classification of information retrieval models

As mentioned in the section 2.2, to retrieve the information from the Internet in mobile computing environment needs a doable and convenient method. It has to let the mobile user getting the satisfied information by his mobile device under acceptable operating environment. To further explain the feasible e-mail based solution for mobile computing environment, we first illustrate the current major models for getting information from the Internet. We divide the current information retrieval model from the Internet by retrieving style into three categories as follows:

- *Web-based Information Retrieval Model, WIRM*

Web-based Information Retrieval Model, which is shown in Figure 2.3, is a traditional model for the user to retrieve the documents on the Internet. It always also means the *On-line Information Gathering Style* which has been explained in section 2.2. Generally, the user in this model always executes a full functioned Internet WWW browser or information retrieval application to fetch the information and to operate the databases in real time. It always requires high computing power platform and always on-line networking connection. In many cases, the client machine should be a powerful desktop PC or notebook PC and to be able to run such complex and powerful applications. Also, the client application always provides the good-looking and fantastic user interface for the user to issue his query to the specified information source, and the user always stays in front of the PC during the whole retrieving process. Because the user always keeps concerning the whole progress of information retrieval, the response time from the information sources should be very critical.

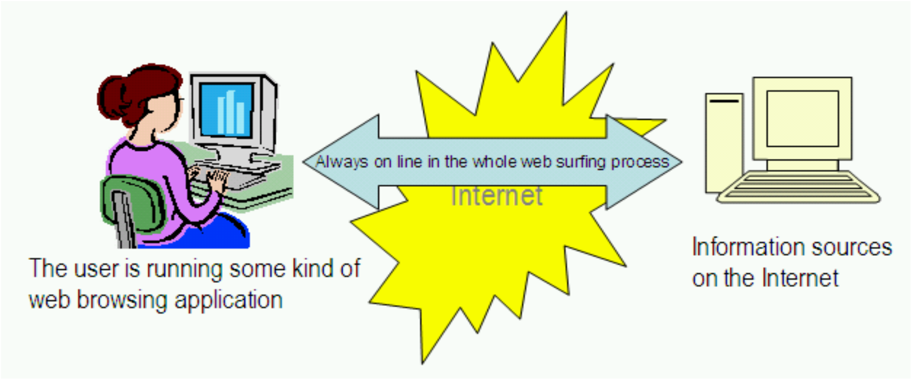

Always on line in the whole web surfing process
Internet
The user is running some kind of web browsing application
Information sources on the Internet

Figure 2.3: The model of Traditional Web-based Information Retrieval Service.

Although most information sources of the Internet provide some kind of good-looking user interface for the user to fetch their documents, there still are few Internet search engines (such as Google, http://www.google.com) provide the concise but straightforward user interface for mobile devices' (such as cellular phones or PDA) WAP browser or simplified mini-browser. However, although the user interface is simple, those information sources still need to equip some sort of WAP-HTML gateway technology. Thus, when the user browses Google's web pages via mobile device, Google will first transform the original HTML format into WAP format. The mobile user then may use built-in or plug-in WAP browser to browse the Google's web pages. The WAP–HTML gateway technology is partly solving the information retrieval issues of mobile devices, but due to the translation technology between WAP and HTML is not mature (especially for Chinese web pages), and it can't optimize the results in a very short time, moreover, the mobile phone's wireless networking is very expensive, this is not a good solution for mobile devices to retrieve the information from the Internet.

■ *E-Paper Style Information Retrieval Model, EPSIRM*

The E-Paper Style Information Retrieval Service, which is shown in Figure 2.4, is a more flexible method for a user to fetch information from the Internet. A user can subscribe some kinds of E-Paper on the E-Paper provider web site in advance by their subscription interface which is provided by those famous information sources. After the user has been subscribed and verified, the E-Paper provider web site will actively send some e-mail with the content of E-Paper specified by the user in advance to the user's mailbox periodically. All the contents of E-Paper from the web site are actively sent into user's mailbox, the user is not required to take care of any other chores except subscription. Of course, in most cases, there is no any customization mechanism for the user to adjust the contents of those e-papers.

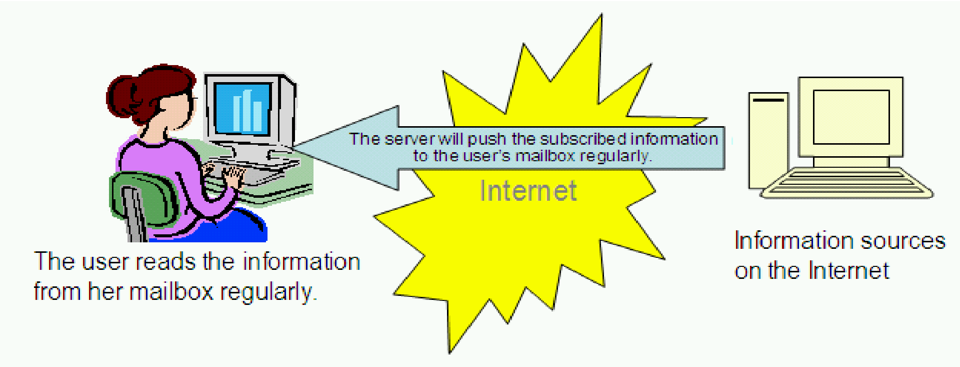

The server will push the subscribed information to the user's mailbox regularly.
Internet
The user reads the information from her mailbox regularly.
Information sources on the Internet

Figure 2.4: The model of E-Paper Style Information Retrieval Service.

As mentioned earlier, this method is not required an always on-line networking connection. In addition, lots of networked mobile devices can read the e-mail directly. Thus, it is a feasible solution for most information retriever of traditional client platform, but one of the defects of the method is that the types of contents of E-Paper are limited, and in most cases, it has no way to customize the content to fulfill the user's personal requirements. Moreover, the web site always assumes the client's machine is desktop PC or notebook PC, so it lacks some acceptable technologies to adapt the contents into different applications and platforms. Again, the most important defeat of EPSIRS is lacking of interaction mechanism between the user and the information providers. In most current E-Paper providers, there is no way for a user to operate any specified E-paper, and to further customize the contents of those E-Papers. Consequently, EPSIRM is not suitable for a mobile user who wants to actively and comprehensively retrieve the information on the Internet.

■ *E-Mail based Information Retrieval Model, EMIRM*

Using e-mail as the tool of information retrieval is not a new idea.

There were lots of researches and products using the e-mail protocol as the front end for other services (includes WWW, FTP, NNTP, Gopher, etc.) on the Internet at the era of poor infrastructure of network. Today, the quality of Internet networking for desktop computing environment is fast and stable. The role of e-mail for information retrieval and other Internet services is replaced by multimedia-enabled web browser. However, for a mobile user with limited resources mobile device, most of the mobile devices have not enough computing power to execute the complex multimedia-enabled web browser and information analyzing works. The primary basic problem about limitation of resources for information retrieval is rising again.

In such a case, the best solution is letting the mobile device be treated as the front end client for the whole information retrieval process, and letting a powerful server as the back end server to fetch and analyze the documents from the Internet. The role of mobile device can be treated as the front end of information retrieval and only to provide a unified operating interface for the user to program, operate, and launch his remote applications, programs, or even mobile agents, and then wait the results returned from the programs or agents in the back end server. Due to most of the mobile devices has not the ability to execute the powerful web browsing, information retrieving applications, and the information analyzing applications, the interactions between the front end and back end are all through the e-mail protocol. That is, if the user wants to retrieve some information from the Internet, he may have to compose an operational e-mail (which is a mail in a proper format for operating the remote applications or information retrieval process) to specified the interested information, and send it to the specified e-mail server. Then, the e-mail server will launch the information retrieval process for the user, and prepare the results into an e-mail. Few minutes later, the results will be sent back to the user’s mailbox. The user may browse the retrieved results from the Internet by opening the e-mail client application again. The mechanism of EMIRM is depicted in the Figure 2.5.

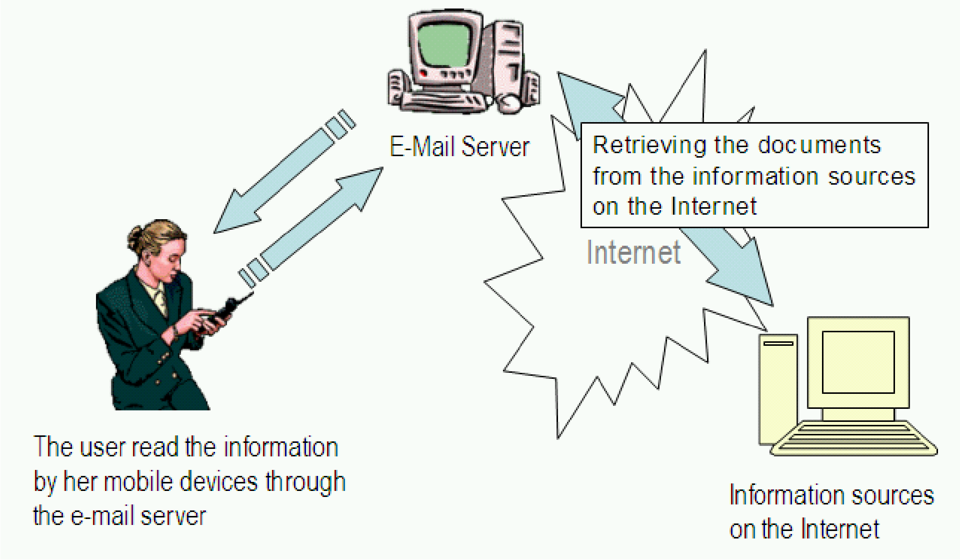

E-Mail Server
Retrieving the documents from the information sources on the Internet
Internet
The user read the information by her mobile devices through the e-mail server
Information sources on the Internet

Figure 2.5: E-mail based information retrieval model.

As Figure 2.5 shows, the user will not directly access the information from the information sources on the Internet. On the contrary, the user directly access to the e-mail server. The works for actually retrieving and analyzing the documents are all done by e-mail server. The major advantage of this model is the efforts of the mobile devices will be dramatically reduced. All the chores can be finished on the powerful and high speed desktop PC or workstation with broadband networking connection.

In summary, the listed three models are suitable for each different hardware and software environment respectively. For the high speed computing environment with broadband networking, the first one is the most popular information retrieval model, but if the user has no time to sit at the front of desktop computer, or the user wants to get regular news and stuffs from some information providers, the second model is the best candidate for him. However, if the user is on the road and grasps some kind of resource-constraint mobile device in his hand, to fetch documents or information from the Internet in such mobile device, the third model is the only suitable solution.

2.4 Basic concept of e-mail based information retrieval

At the client side point of view of e-mail based information retrieval model, when the user wants to do something on the Internet or private database in remote location, he may use any front end device (Desktop PC, notebook PC, PDA, or mobile phone) to launch the e-mail client application, and then writing the commands provided by proposed framework or mechanism as the e-mail subject and body, sending this operational e-mail to a specified e-mail account on the back end server which is registered in advance, and finally, waiting the reply from the back end server. Few minutes later, he may execute the e-mail client application again to check and see his mailbox. In most cases, he will get a response from the back end server, and the contents of the e-mail always are the executed results of his delegated remote application or mobile agent in the remote server.

At the server side point of view of the e-mail based model, the back end server always checks the registered e-mail account to see any incoming messages. When a message has received, it first parses the subject of the e-mail to determine the type of the e-mail and to verify the id and related information according to the information of this e-mail. If the e-mail is with some simple operating commands (such as '&&SS;' to show the status of the server), the server will operate these commands right now, and will pack the results into an e-mail, and then replies the e-mail to the user's e-mail account as soon as possible. In other case, if the user's operation sheets are complex, the server will reply a status report e-mail to the user, and then packs these operations as a program for executing. Meanwhile, the server will send the current progressive statistics to the user every 10 minutes (the default value, it can be changed by the user).

The key mechanism of this model is the role of the e-mail server. It is responsible for accepting the user's request, and will act as a coordinator to manage the information retrieval application or mobile agents on the server. All of the interactions between the user and the programs on the server should be through the e-mail protocol. Every result e-mail will also be rearranged in a specific manageable format, which can be further operated by user to fetch more information from the Internet by specifying the identification number in the e-mail.

Using e-mail protocol as the gateway between the front end device of information retrieval and other Internet services has many advantages which are shown as follows:

- ***Real Cross-Platform***: No mater what the devices the user will use, includes desktop PC, notebook PC, PDA, or even mobile phone, most of these devices with networking ability are all equipped the e-mail client application. So, the e-mail based IR mechanism provides the real cross-platform service.

- ***User Simplicity***: Due to the simplicity of e-mail protocol, there is no any extra works has to be done for the mobile device of the users. All he needs is to look up some information retrieval commands, and composes those commands into a proper operational e-mail. The other operations are all the same as the normal e-mail application.

- ***Compatibility and Flexibility***: The e-mail protocol is a very simple but complete protocol. The contents of e-mail can be represented in standard ASCII encoding. Every program can easily read and process the subject and contents of any standard e-mail messages. Thus, the mobile device with limited resources can also easily process the contents of e-mail too. Nevertheless, for the more powerful platform, such as desktop PC or notebook PC, of course they can process more complex data by attaching those data files into the operational e-mail.

- ***No client application installation***: Because of the e-mail is the common application for many networked devices, most of these mobile devices don’t need to install any related program for adapting this model. This is very convenient for system developer to propose their new services. All the new services are only to be installed into server. There is no need to modify any application at client side. The cost of deployment will be greatly reduced.

- ***No user training***: Almost every mobile device users have enough knowledge and abilities to make use of e-mail application to process their own messages. So, this e-mail based model can save lots of extra training works.

- ***Low cost, and low and short bandwidth required***: Because the e-mail based model doesn't need to educate the user, and it also doesn't need to install any program into client side, the cost and time will be greatly reduced. Moreover, because the e-mail protocol basically is some kind of off-line network protocol, it needs the network connections only while the client application is being transferred the e-mail. Thus, the networking cost will be dramatically reduced too. For a mobile network (such as GSM/WAP/GPRS connection services) with high charged networking service, this is a great deal.

- ***User-Oriented and autonomous information retrieval process***: Another important characteristic of this e-mail based model is that the model is a user-oriented and autonomous information retrieval process. Under the architecture, all the complex retrieval operations have to be programmed in advance. After the system have received and parsed this e-mail, it will launch a static agent to monitor and schedule the user's mobile agent to do its works, and the static agent also will report the executing status to the user periodically. All the works will be automatically run at the server or across each needed hosts. The user also can write the event handler in their commands sheets of the e-mail to guide the static agent treating every possible occurred event during his works. Finally, after all the works have been done, the static agent will send a complete and well quality result to user's mailbox. That is, if the user's commands sheet is a good design for information retrieval, the boring progress for information retrieval can be discarded. All his works have to be done are just to wait the returned results form his own mobile agent.

- ***Server scheduling optimization***: Beside, because all of the user's information is handled at the server side, the system can optimize the system's performance according to the system's states. Many technologies such as cache mechanisms, scheduling mechanisms can be applied in the system to boost the system’s performance.

2.5 Classification of e-mail based information retrieval

According to the description of the previous sections, we conclude that the e-mail based information retrieval model is the best candidate for the resource-constraint mobile devices. To utilize the e-mail by mobile devices to fetch the documents and to retrieve the information from the Internet or private local area network, according to the complexity of the model, we also divide them into three categories which are shown as follows:

- ***Data-Driven System***: All information retrieval functions are embedded in the main server. The user can only specify the interested data field, and then expects the corresponding results of those data fields returned from the remote server. In this type of e-mail based information retrieval system, the user has no flexibility to determine the control flow of the information retrieval process. That is, the user only can specify some simple query commands with simple arguments to get the information from the server. There is no any control statements and other further advanced statements can be specified by the user. Although such the system can provide the user having the ability to install their own program into the server to achieve their own information gathering purpose, in the client side, he only can just call the program with some simple arguments. He can't specify the repeated execution times for some specified statements or programs, and he also can't determine the control flow according to the return value of any program or application.

- ***Control-Driven System***: This type of e-mail-based information retrieval system is more flexible than data-driven system. It is supporting some kind of information retrieval script language. The user can utilize such script language to write some simple batch programs for retrieving information from the Internet. Moreover, the user can also operate the remote mobile agents by writing programs with this kind of language. This type of e-mail based information retrieval system at least provides basic flow control statements, such as decision statements and loop statements, letting the user has more ability to control the execution flow of process of information retrieval. Furthermore, the well-formed results can also be treated as the reference for the next operational e-mail. The user can manage all of the functions and applications built in the system by writing a program in their operational e-mail. In most cases, this type of information retrieval is enough for mobile user to comprehensively fetch their interested information from the Internet.

- ***Full Interactive System***: In most cases, the Control-Driven System is enough for most mobile users. However, sometimes, if the system has not provide the user's desired application, and there is no appropriate desktop PC for the user to install or deploy his own application, in such cases, the system also needs to provide the user the ability to install their own application into the remote server by his simple mobile device.

This is the main purpose of the invention of the Full Interactive System. This type of e-mail based information retrieval system supports all the ability of programming language. Actually, it just extracts the program from the operational e-mail, and dispatches the extracted program into the operation system. After the execution of the program has been finished, the results will be wrapped into an e-mail, and then be sent back to the user’s mailbox. In this style, the user can write any customized mobile agent with this kind of programming language. For convenience of users, the programming language can be the traditional language such as C, JAVA, Perl, and any acceptable shell language etc. In practice, the interpreter style language, such as Perl and some shell language, is easier to implement.

Different type of e-mail based information retrieval system has different characteristic and its applicable domain. The Data-Driven System is simple, and it is just for the user who only wants to query his simple interest information from the private database or some specified information sources on the Internet. Getting the item price of product of our company, getting the timetable of the school, or quoting the stock value from stock market are all the application examples which can be implemented by Data-Driven information retrieval system.

For more complex applications, such as surfing the Internet, or retrieving the information across many search engines, the Control-Driven information retrieval system is more viable than Data-Driven System since it provides more flexibility for the user to control the results from the server. In practice, the system will provide the control statements such as for/next, if/then, while/wend letting the user to control the execution flow for his information retrieval process. Furthermore, the system will also rearrange and reformat the results in a well-formed format, letting the user can directly utilize the results to retrieve the information from the Internet more intensive.

Sometimes, the simple control flow can’t fulfill all the requirements of the user. To support full function programming language is also important. The major advantage for an e-mail based information retrieval system is the great flexibility. In such system, the user can not only operate the remote application to obtain his interested information, but also write his own application by mobile device. Although the application is written on the mobile device, the execution place is the remote server. Since the programming language can do everything that the computer can do, the user can do everything by his mobile device as well.

2.6 Applying mobile agent runtime mechanism to our design

To further maximize the ability of the e-mail based information retrieval system, applying the mobile agent technologies into the system is one of the most important approaches. A mobile agent is an autonomous software program which can be migrated from one host to another host during the execution time. Since the mobile agent technologies can provide the programmer to write a program which can easily execute itself on any available host, this mechanism can achieve maximum flexibility in any stable and unstable network environment. However, applying the mobile agent technologies into mobile computing environment is very different in typical desktop environment. It still has some considerations which have to care about. Some of these main considerations are shown as follows:

1. The host always has to provide a running mobile agent runtime environment to accept the migrating request from the other mobile agent in remote host.
2. The network of hosts always has to be available.
3. If the migrating request from mobile agent has received, the host has to response as soon as possible.
4. The mechanism recommends more network bandwidth for processing the migration of mobile agents and other related works. Sure thing, the networking cost is more expensive.
5. Both the client and server side have to install specified runtime environment. Then they will have the ability to launch, migrate, and operate the mobile agents. Because the client side applications always need more computing power, lots of mobile devices can't fulfill these requirements.

The mentioned considerations let the mobile agent technologies can’t be easily adapted to mobile computing environment. That is, applying mobile technologies into mobile computing environment needs many modifications. To defeat the shortcoming mentioned above, we propose an Email-based Mobile Agent Runtime Environment called e-MARE. The e-MARE provides the users the ability to utilize the unified interface for many different kinds of devices to control, design, and launch their mobile agents. All of the mobile agents, built-in applications, or custom applications are executed in the remote server. Based on e-mail protocol, the obviously advantages are the convenience, simplicity, and uniformly for the operations of the user. The main advantages are shown as follows:

1. ***Convenience***: The user can easily operate the mobile agent on any networked devices, because the e-mail application is almost the standard application of networked devices.

2. ***Simplicity***: All of the design, manage, operate, and launch operation can be completed in e-mail as the commands or script language in standard ASCII encoding. Also the executing results or system's states are all returned via the e-mail protocol.

3. ***Uniformly***: Because the e-mail protocol is the most popular protocol for most networked devices, whether the device is mobile phone, PDA, desktop PC, or even workstation, the operation of e-mail application is similar. That is, the interface for operating the mobile agents is almost the same for most devices.

Beside, different to the traditional mobile agent system, there is no extra application has to be installed into client's devices. This obeys the thin-client requirements for Internet application. In e-MARE, all works will be done in any standard e-mail application. For most low-end mobile phone or PDA, this is the best and only mechanism to operate mobile agent for information retrieval or processing data from the Internet.

Chapter 3

Background and Related Works

The purpose of this chapter is to briefly introduce all of the background technologies which are related with our research. Many related works and some real products of the market will also be presented and described in this chapter. The mentioned technologies include GSM cellular phones, the Internet solutions for mobile phones (WAP, GPRS, 3G, i-Mode…, etc.), the PDA and desktop PC, MExE, information retrieval, and mobile agents. The related works and real products include AGORA services, Info-On-Demand, AvantGo, iSMS, semi-automatic wrapper generator, …, etc. Some references about mobile agent runtime environment technologies will also be described in this chapter.

3.1 Background and terminologies

This section will survey and explain many related internetworking devices, terminologies, technologies of the market, and then describe the possibility for operating mobile agent by mobile devices. Some important technologies and researches will be explained more detail in the following chapters.

3.1.1 Mobile cellular phone

According to the survey of the International Telecommunication Union (ITU) in 2003, the penetration ratio of mobile phone in Taiwan is 106.15%, which is the 1st all over the world. Needless to say, the applications of mobile phones are more and more popular and fantasy. Now, connecting the mobile phone into the Internet, and utilizing the great amount information and applications on the Internet is the great ambition for most mobile phone service providers. The research [44] estimates the following statistics about the mobile phone users all over the world:

- 540 million roaming cells were made in February 2000
- 750 million cells were predicted in June, July and Augusts 2000
- Data will account for between 20 and 50 percent of all global wireless traffic by

2004

- 8 billion short messages were sent in May 2000
- 10 billion short messages were sent in December 2000
- One billion short messages were sent per month in Europe alone in 2001
- GSM (Global System for Mobile Communication) grew at 80 percent in 1999: PC grew at 22 percent
- More GSM terminals will be connected to the Internet than PCs by 2005
- Wireless devices will access 30 percent of all Internet traffic by 2005

According to the previous listed information, it can conclude that utilizing mobile devices to surf the Internet is more and more important and popular activities on the Internet. If there is an easy and cheap way for mobile users with mobile phones to interact to the Internet, the mobile user's vision will be dramatically expanded. However, before the 3G era coming, use the small screen size, very limited computing power, short battery duration, and the expensive and low bandwidth network mobile phones to connect to the Internet and retrieve the information from it is very difficult. All the solutions on the desktop are no more applicable on the mobile phones. For instance, in PC, if we want to search something on the Internet, we only have to connect to the network, and then install any WWW browser. Consequently, we may view any documents and information available on the Internet via the browser. However, the same works can't be easily applied into mobile phones. It is impossible for most mobile cellular phones to install any full function WWW browser, and the very limited size display is not suitable for reading the common WWW pages on the Internet.

Furthermore, most current mobile phones of the market have no any ability to expand their function. So we only have to use the built-in components of the mobile phones to operate the remote mobile agents and applications. The well known built-in messaging functions are SMS (Short Message Service)/EMS (Enhanced Message Service) /MMS (Multimedia Message Service), WAP (Wireless Application Protocol), or E-MAIL. We have to choose one of the messaging functions as the media for operating to the mobile agents on the remote hosts. We will describe the related technologies of WAP, SMS/EMS/MMS, in later of this section.

First of all, the type of technologies for the networking connection between GSM mobile phones and the Internet are shown as follows:

- ***Wireless MODEM***: Wireless MODEM is a traditional and the most common equipment for GSM mobile phone to connect to the Internet. Actually, the function of wireless MODEM is almost the same as the wired MODEM of the traditional wired home network. Nowadays, almost current modern mobile phones are equipped the wireless MODEM. As long as the GSM mobile phone service provider provides the data transmission service, the mobile users can treat the wireless MODEM of the mobile phones just as the normal PC MODEM. That is, when the mobile phone has connected to the computer by transmission cable, IrDA, or even BlueTooth, the operating system of computer will recognize the wireless MODEM as a normal general MODEM. The user may setup the MODEM and dial up to the Internet Service Provider. Just as a normal wired MODEM does. Thus, if the user has enabled the data transmission functionality to his GSM service provider, he can connect his computer via his mobile phone as well as the wired network connection. Of course, those mobile phones with embedded WML (Wireless Markup Language) browser can direct connect to WAP sites on the Internet, but the bad news is that the network speed now in Taiwan is only 9600 bps, it is very limit for most Internet application services. Moreover, the connection cost is very expensive in Taiwan too.

- ***GPRS (General Packet Radio Service)***: Work on GPRS started in 1994, and a standardization of the GPRS had been frozen by ETSI (European Telecommunications Standard Institute) in 1999. It is a packet-based data bearer service for GSM networks, which provides both standards with a way to handle higher-data speeds. It will make mobile data faster, cheaper and user-friendlier than wireless MODEM solution. By introducing packet switching and Internet Protocol (IP) to mobile networks, GPRS gives mobile user faster data speeds, and particularly suits bursting the Internet and intranet traffic. Connection setup is almost instantaneous, and users can have always-on connectivity to the mobile Internet, enjoying high-speed delivery of e-mails with large file attachment, Web surfing, and access to corporate LANs. This technology is treated as the temporary phase before the arriving of 3G mobile phones. In the past, there were more than 6.4 million GPRS users worldwide by end 2002, and it was approximately 150 GPRS networks operational in 58 countries. Also, there were more than 50 million GRPS terminal devices delivered worldwide.[52] These numbers are all increasing in a very great pace currently.

- ***3G (3rd generation of mobile phone)***: The third generation mobile phone infrastructure is the prospect of the mobile users for transmitting information from the Internet in a very high speed. In 3G, all voice, video, and data transmissions are all wrapped together. 3G discards traditional GSM's technologies, and uses Packet IP technique as the kernel technology. It will let the users always stay on the Internet network, and can own the very high speed networking bit rate. At the beginning, the providers promise 3G mobile phones, at least, have 2Mb-bandwidth available. Thus, most applications can be done by mobile phone itself. However, the real available applications are determined by the ability of 3G mobile phone.

Although there are some limitations for the mobile phone to connect to the Internet, it isn’t means that the mobile phone has no chance to connect to the Internet. One of the feasible solutions for mobile phone to connect the Internet is to prepare a server for translating the HTML pages into some kind of documents for easy reading of the documents. That is, the documents on the Internet have to be filtered out some unwanted information (such as advertises, logo, unimportant hyper links, and java applets) and to remove those complex structures (such as table, Dynamic HTML, Cascading Style Sheets, and any script languages) by at least a specified server. Then, the filtered documents have some choices to receive.

No matter the users use what kind of networking technologies for connecting to the Internet, there are four possible types for a mobile phone to operate the e-mail. They are shown as follows:

- **Using the mobile phone's built-in e-mail application**. This is the most convenient way for user to operate the e-mail. Some modern mobile phones (such as SonyEricsson T65, T68, T230, Nokia 3650, and OKWAP 166 etc.) have provided the built-in e-mail application for users to connect to the Internet, and send/receive the e-mail from the mail server of the Internet. All the works the user need to do are to compose the appropriate arguments in their mobile phones, and then he can read the messages which are just fetched from the Internet on the mobile phone. Because the e-mail applications are the built-in function of the mobile phone, the performance and convenience are much better than other solutions. Moreover, because the process of reading and writing messages are connectionless, the cost for networking will be reduced greatly. Except for the ability for read/write the standard ASCII, some mobile phones provide the ability for user to send the attachment too.

- **Using the GSM service provider's STK (Subscriber Identity Module Application Tool Kit)**. The STK is a technology for programming some applications in SIM card of GSM mobile phones. It uses the limited available free space of SIM to install the application for providing the interface between the users and the service providers. Most providers utilize the GSM mobile phone's STK technique to provide the users to read/write the e-mail by his older model mobile phones. Almost every GSM mobile phones support the STK technique. Combine the STK and SMS, most of the application for exchanging messages and send/receive e-mail can be implemented easily. The major advantage of this solution is that almost all GSM mobile phones are supporting STK and SMS.

- **Using the WAP browser via the WAP protocol for GSM network**. If a mobile phone is WAP-enabled, it also stands for this mobile phone is equipped a WAP browser. If a mobile phone is equipped a WAP browser, it means the mobile phone can direct browse any WML site via wireless MODEM or GPRS MODEM. In this case, we can design a WML site to access any mailbox for users, thus, the user can indirect access his mailbox through the WML site. Different to the former two methods, the major different is the location of processing the mailbox. The former two methods, the place to process the user's mailbox is in mobile phone, but this method is at the WML server site. The advantage of this method is that the powerful server may provide versatile services with the e-mail service. There are lots of extra functionalities can be added into the e-mail services. We don't need to worry about the poor computing power of the mobile phones. However, the disadvantage is that for accessing the WAP services, the users have to keep the connection of the network. The connection cost is very expensive for a very long time connection via traditional wireless MODEM. Moreover, if the connection quality is poor, to write an e-mail is a very difficult thing to be done. GPRS technology solves part of the mentioned problems.

- **The ultimate solution: using the built-in WWW browser, or installing a WWW browser in high-end cellular operating system.** When the mobile computing technologies are growing mature enough, the method will be the best and killer solution for the mobile device. Actually, many high-end and smart mobile phones on the market, such as Nokia 6600, SonyEricsson P800/P900, have already built-in some sort of web browser. The user also may install his WWW browser in some high-end operating environment of high-end mobile cellular phones. However, the connection speed and the CPU speed of mobile devices are the most important consideration of this solution. The battery duration is also an important consideration in some long-term data processing application. For example, the Internet browser of Pocket PC 2002/2003 is equipped many functionalities. It is very similar to the Internet Explorer of Microsoft Windows on the desktop PC, but the CPU speed is not enough for executing some complex and fantastic functions. It is very slow for user to obtain his desired information.

As mentioned above, it is obviously that almost every mobile phone has the ability to connect to the Internet and operate e-mails of his mailbox no matter whether it supports WAP, GPRS, or not. Thus, if the remote application and mobile agent can be designed, operated, and processed via e-mail by any mobile device, it will expand the ability for mobile phones to retrieve information on the Internet.

3.1.2 WAP（Wireless Application Protocol）

The transfer rate of wireless network is much slower than traditional wired network, so it needs a feasible transmission protocol to adapt such environment. WAP is one of the best candidates. The WAP which was founded in 1997 by Ericsson, Motorola, Nokia, and Unwired Planet, is the leading standard protocol for information services on mobile phones and other wireless terminals. It is based on Internet standards an open, wireless application software protocol. The main objective of WAP is to provide an interaction between the traditional GSM mobile phones and the Internet. The system developer design WML site on the Internet, and the client user uses the micro WAP browser to browse the WML site of the Internet through the wireless MODEM or GPRS MODEM. As mentioned above, the application of WAP is not only on the mobile phone, it can also be applied at PDA and desktop PC. Through WAP gateway, the devices can fetch the WML sites on the Internet. The main advantage of WAP protocol is that most of the works of applications are implemented at the server. Because the server is much powerful than mobile devices, lots of complex works can be worked very well. The application can be designed more versatile and complex. Actually, a WAP browser can be treated as the display of the application, the core of works can all be done by the back end server. It is very easy for the developer to develop the WML sites on the Internet.

At the beginning of the development of WAP, the speed of networking for GSM mobile phones are only 9.6Kb. Although the speed is enough for many simpler applications, it is not enough for many complicate applications, especially for multimedia applications. Today, GPRS is the more popular and more acceptable for data transmission of GSM mobile phones. If GSM service provider supports GRPS technology, some modern mobile phone can at least speed up more than 100Kb.

General speaking, if we want to use and operate mobile agents on the remote host, utilize WAP with GPRS is a fair well solution, but for a heterogeneous mobile agent environment, this is not a good choice. The reasons are shown as follows:

- ■ It is not a real cross platform solution. Although most mobile phones support WAP for Internet connection, but it is not the standard for other platform (such as PDA, desktop PC, and even workstation). For those platforms, they have to install at least one specified WAP browser to browse WML-based site. Furthermore, the WML-based site is designated for mobile phones. It can't take the advantages of the other platforms. Thus, to take advantage of the characteristics of each platform, the mobile users have to know more than two user interfaces and operations of the applications. It will loose the advantage of cross platform.

- The cost of network connection is very expensive. The most problem for mobile users to access the information on the Internet is the expensive fee for GSM's wireless networking. This is also the big problem of WAP protocol. In the traditional typical browsing process, the network connection has to keep always online between the client and server, even the user is just reading or writing the e-mail.
- The server is not easy to optimize the jobs. To save the connection time and money, all the user's requests by WAP browser have to be responded as soon as possible. So there is no chance to perform maximum optimization for the following requests.
- Furthermore, the difference between WML and HTML is large. There is no convenient and efficient way for the web developer to adapt the present HTML-based web pages into WML-based pages. There are lots of things have to be done for such HTML-WML transforming process. Sometimes, many developers prefer to design a new WML page instead of transforming.

Consider to the above disadvantages of WAP protocol across multiple heterogeneous platforms, we can conclude that the pure WAP protocol is not a good solution for the mobile user to browse the information on the Internet by the versatile mobile devices.

3.1.3 SMS/EMS/MMS of GSM

SMS is one of the most important services of GSM mobile phones. All GSM mobile phones provide the ability for sending and receiving the SMS. In general case, the maximum length of SMS is about 160 characters. The length is enough to carry some simple messages, but it is not enough for more complicate applications. The contents of SMS can only be written in ASCII standard. EMS and MMS protocol conquer these shortcomings. These two standards are not only extending the ability of text, but also provide the ability for transmission of voice, and images. Of course, the allowance message size of these two standards is also expanded.

Because MMS allows the user to transmit not only the text messages, but also the voice, image, and even video clips, furthermore, there are more and more camera phones available on the market. There are more than 130 commercial MMS launches in 53 countries worldwide in May 2003. The MMS subscriptions already estimated to exceed 3 million in EMEA region. Nokia even forecasts in 2003 more than 50% of all handset sales will be MMS-enabled. [44]

However, although the SMS/EMS/MMS is a standard application for all GSM mobile phones, but for other platforms, it is not a standard application. Moreover, the size of SMS is too small to carrier all of the results after the information retrieval in a single messages. The cost for sending and receiving the SMS/EMS/MMS is also too expensive in Taiwan.

In our viewpoint, the SMS/EMS/MMS is only suitable for assisting the information retrieval works. Even though we don't consider the transmission speed and the allowance message size of SMS technology, pure SMS-based information retrieval solution is not an economic way for information retrieval because of the high charge fee for SMS services in Taiwan, but notifying the mobile user to read the information in his mailbox is a feasible and well application for SMS technology. Our prototype system, e-MARE, also implements this functionality.

3.1.4 i-Mode

i-Mode is developed NTT DoCoMo of Japan in 1999, which is designated to replace WAP. Different to WAP's WML language, i-Mode uses the cHTML (compactHTML, which is much similar the HTML) as the documents format of i-Mode website. Because the cHTML is much like the HTML, it is easy to transfer the living HTML website into i-Mode website. Of course, the ability of cHTML is much bigger than WML. In some places, i-Mode is much popular than WAP. Although the i-Mode web site is very easy to implement, the number of mobile phone models which is supporting i-Mode is few. Thus, the technology is not suitable for our research in general cases. In Japan, The users of mobile-based Internet services exceed 60 million before Feb. 2003, and the i-mode users are more than 34 million. However, until now, the i-Mode is not the popular services all over the world. In Taiwan, until October 2002, there were only 22,000 users select i-Mode as his GSM services. Today, there are only five models i-Mode cellular phones in Taiwan. Thus, it is not suitable for our approach in Taiwan.

3.1.5 PDA and PC

Whether the ability of mobile phone is more and more powerful, the limited display size and inconvenient input interface are always the big problems for most mobile users. PDA is the good choice for mobile user to keep both small and powerful characteristics. Lots of modern PDA has the ability to access the wireless networks. Also, the e-mail is always the standard application for most modern PDA. It is also a good platform for utilizing e-mail to operate the mobile agents on the remote host no matter what the OS of the PDA is.

Desktop PC is a powerful and the best platform for operating the mobile agent. There are lots of applications can operate mobile agents for users. Although the desktop PC has enough computing power to do anything about information retrieval and operating mobile agents, to keep the uniform accessing interface to the mobile agent, and take the compatibility among all types of terminals, we choose the e-mail application to operate mobile agents. Thus, it is not required for the user to install any extra application in any terminals through the whole information retrieval process.

As the description of the above sections, we understand that there are many solutions providing the user to connect to the Internet among several types of platforms. It means that no matter what the devices that the user uses, he can connect to the server on the Internet via their mobile devices. Thus, if there is a server which is on the Internet and providing the services compatible with the existing protocol of those mobile devices, the user can easily utilize these services as well. However, most of the mobile devices, especially mobile phones, are resources limited, and the networking connection of mobile phones is very unstable, slow, and expensive, to execute any customize executable application on those mobile phones is difficult or even impossible. Fortunately, the e-mail protocol is both common and popular for most mobile devices, and its interface is very similar among every platforms. Thus, utilizing the e-mail as the front-end of information retrieval on the Internet is very appropriate.

3.1.6 Mobile Execution Environment (MExE)

MExE is an exciting new specification from the 3GPP (3rd Generation Partnership Project). The MExE specification details a flexible and secure application environment for 3G (and 2G+) mobile devices.[38] The aim of the standardization of MExE is mainly to provide a standardized application execution environment for mobile device in general, including the devices between small, light and sophisticated devices. The core of the execution environment is based on the ability to communicate the MExE mobile device's supported capabilities with a MExE service provider. This allows the applications to be developed independently. Currently, MExE consists of three Classmarks, and they are based on WAP environment, PersonalJava Environment, and J2ME CLDC environment respectively. MExE technology is possible one of the most important research topics of future 3G mobile phone infrastructure, but for the current hardware supports of mobile phones, to construct the MExE environment is a very hard mission.

3.1.7 Information retrieval

Information Retrieval on the Internet [1][53] is one of the most important applications for the information exploded era. There are many researches [15][18][19][25] are focused on how to efficiently extract the useful information from immense Internet. In addition, the study on meta-search is to combine the results from several different information sources [22][23]. Reference [45] also has been proposed an integrated information retrieval system for user to intensively retrieve information on the Internet with unified user interface. A good information retrieval tool let the user getting large amount, more accurate, and more comprehensive information from the Internet. The challenges of information retrieval problem are not only facing great amount of information, but also facing the versatile data structure and document formats of the information sources. The documents on the Internet are semi-structured or even unstructured. Due to lacking of editors' proof, many researches have pointed out that the quality of the documents on the Internet is also a big problem.

So far, the documents fetched from the Internet most are encoded in HTML (Hyper Text Markup Language) standard. The HTML is a format-oriented language. That is, HTML only cares about the output style of the data, not the meaning of the data. The layout and style of the webpage is the main focus of HTML. For the application of information retrieval, the appearance of the webpage is not important. Actually, too many flourishes will let the web page hard to be analyzed. The thing that the applications really care is the contents and structures of the documents of web site. For example, if the application is processing an article from a news web site, the focus of processing is 'who is the author', 'what is the subject of the article', 'how to abstract the key point of the article', and 'what is the intent of the author want to say', etc. Adversely, the font color and font size of the title, where is the title should be positioned, are not the considerations for the information retrieval application. Unfortunately, this is not the specialty of HTML. However, there are lots of researches are dedicated in this area. They also have got some good results.

Except for the unstructured stuff of information sources, the multiplicity of those information sources is also the big problem for information retrieval. Different information source has its own protocol for retrieving its data. For the convenience of the user or application to browse the data from the versatile information sources, to design a uniform interface of meta-search system is very important. Of course, in this area, there are also lots of researches devoted. Currently, the major researches of the unified interface for searching the Internet are treated by Mediator/Wrapper architecture. They provide a unified user interface for user to enter the query commands. The mediator will parse the query commands and dispatch the appropriate wrapper to specified information source. After the wrapper has activated by the mediator, it will access that specified information source by the well-formed commands. To filter and transform the versatile format into a unified internal format of the returned results from the specified information source is also the responsibility of wrapper. Utilizing the Mediator/Wrapper technology, it will improve the access between versatile formats of the information sources on the Internet. If we want to access a new information source with different document format, the only works that system developer has to do is just to construct a new wrapper. The main architecture of the information retrieval system doesn’t need to change at all.

As mentioned in Chapter 2, there are more and more people using their mobile handheld devices to connect to the Internet, and fetching their interested information. Thus, the information retrieval via mobile handheld devices is very important. For a mobile user, to provide the ability to retrieve the information on the Internet and to unify the interface for versatile information retrieval devices are both important things have to be concerned. However, most traditional information retrieval systems are based on the desktop PC with good quality of networking and high computing power. Obviously, they are hard to directly apply these current information retrieval technologies into the mobile devices with poor quality of networking and limited computing power. That is, it is hard for mobile phones or PDA to browse the WWW site on the Internet. Furthermore, because the network always has to be kept on-line during the browsing process, the networking fee is the big burden for the mobile user. These problems can be solved by the connectionless protocol, e-mail protocol.

In our viewpoint, most networked devices provide the solutions for send/receive of e-mail. Most of the processes for e-mail are all processed in a standalone machine. In other words, when the user is composing or browsing the e-mail, it doesn’t need to connect the network between the mobile device and the server. The cost that users have to pay is only the time period during the sending and receiving the messages. Because most of the time of the mobile devices during the retrieval process is at the off-line states, the server has more time to analyze the documents requested by the user. Altogether, the user may get the higher quality results by his mobile devices.

3.1.8 Mobile agents

An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives [34]. And, a Mobile Agent, step further, is not bound to the system where it begins execution. It has the unique ability to transport itself from one system in a network to another. The ability to travel allows a mobile agent to move to a system that contains an object with which the agent wants to interact and then to take advantage of being in the same host or network as the object [6].

Again, the environment of most current mobile agent system is an active and full-connected networking running mechanism. All mobile agents have to be launched to a main host, and autonomously execute the delegated jobs for their master. If the mobile agent finds that there are lots of documents are at the remote host, it will automatically request other remote agent for help or even migrate itself to that host to save the network usage, but the premise is that all hosts have appropriate mobile agent runtime environment to accept the request from other remote host.

The computing power of mobile devices is very limited, and to input characters to the mobile devices (especially to the mobile phones) is a very time-consuming works. Although the user can write some complex mobile agent programs on the e-mail of the mobile devices, we encourage the user to write the whole complex programs on the server in advance, and write the simple commands to operate the remote mobile agent on the server.

Thus, our prototype system, e-MARE, can be treated as some kind of application gateway or mediator between e-mail to mobile agent. The user can issue the commands via e-mail by a mobile device to operate the remote mobile agent in the server. The mobile agent is delegated by the user to fetch and analyze the information on the Internet. It may migrate itself to other server when it found there are many data available on the other server.

3.2 Related works

At the beginning of the development of Internet, it is a very common idea to utilize e-mail protocol to provide many services of the Internet [35][36][37][38]. One of the major reasons is that the e-mail was the most popular application on the Internet those days. As mentioned above, if a device which is available for internetworking, the e-mail application is of course the basic and recommended application. The e-mail is in essence a cross platform application. An e-mail message can be processed at any handheld terminal, mobile device, host and client's PC. Thus, we can easily send the receive messages by the e-mail protocol anywhere at anytime. There is also lots of Internet services can be delivered by e-mail protocol in some systems. To deliver the web pages of the WWW server is one of the most important applications of e-mail except the delivery of e-mail itself. The following two applications are the introduction of the two important approaches in the earlier age on the Internet.

3.2.1 AGORA

In 1994, AGORA server was developed by Arthur Secret of the World Wide Web Consortium. It is an e-mail browser framework for the WWW. The service provides e-mail only access to information resources stored on the World-Wide Web. The servers enable e-mail only users to simulate the experience of browsing the Web by sending e-mail requests for either text or HTML versions of documents stored on World Wide Web sites [36]. Under this mechanism, the user can just to write some simple instructions or commands in the body of e-mail. There is no need to set the subject of e-mail. The user then sends the e-mail to the host which is providing the AGORA service. Few minutes or hours later, the AGORA server will send the requested web pages to the mailbox of the user. Consider to some e-mail client can't interpret the extra format of HTML, the server always remove all of the unwanted

HTML tags and script languages of the web pages in advance. It also collects all of the links of the web page, and rearranges them in to an easy-to-access format.

That is, if we want to browse the web site http://www.yahoo.com by AGORA service, the only thing we have to do is to compose an e-mail without subject, but the body of that mail has to be entered following text:

GET http://www.yahoo.com

After we have sent the above e-mail to www4mail@access.bellanet.org, several minutes later, the filtered contents of http://www.yahoo.com will be in our mailbox. Of course, if we are familiar with the URL encoding for any search engine on the Internet, looking for some keywords on the Internet through the specified search engine by e-mail application is also an easy way. The following text of the e-mail body is the example:

GET http://www.google.com.tw/search?q=chrysler+neon&ie=UTF-8&hl=zh-TW&lr=

The major advantage of AGORA service is the easy deployment for user. The user only has to prepare an e-mail read/receive environment. The requirements of client for hardware and software can be reduced dramatically. For a user, he can freely browse the information of the Internet via e-mail client application. Moreover, if he familiars the encoding method of any specified search engine on the Internet, to retrieve information from the Internet via e-mail is not a hard works neither.

The major disadvantage of AGORA service is lacking of interaction mechanism between the Web sites and the user. The instructions set of AGORA is too simple to operate the whole process of the Internet browsing. All URLs have to be entered by user, and the requested results can only reflect one-time search. If the user has to analyze much data from many information sources, it is a very boring works, and will waste very long time. Furthermore, to input many characters for mobile users is a very time-consuming job.

3.2.2 Info-On-Demand

Info-On-Demand [35] is a commercial software product. It is, in general, an e-mail agent, which it can update and provides ERP-information across platforms to users' PDA, notebook, desktop PC, or even mobile phones via WAP. It provides a WWW interface on the Internet for user to set and register his simple batch programs. To set and register these works have to be finished by desktop PC networked to the Internet in advance. After all the setting is finished, the user now can send an e-mail with specified encoded subject and well-formed mail body to the specified e-mail account. The system of the server will parse the subject and contents of the e-mail, and then fetch user-specified data field from the database according to the requests of the user's commands. When the data requested by the user has completely fetched, the system will wrap the result into an e-mail, and send it to the user's mailbox as soon as possible. The architecture of Info-On-Demand [35] is shown in Figure 3.1.

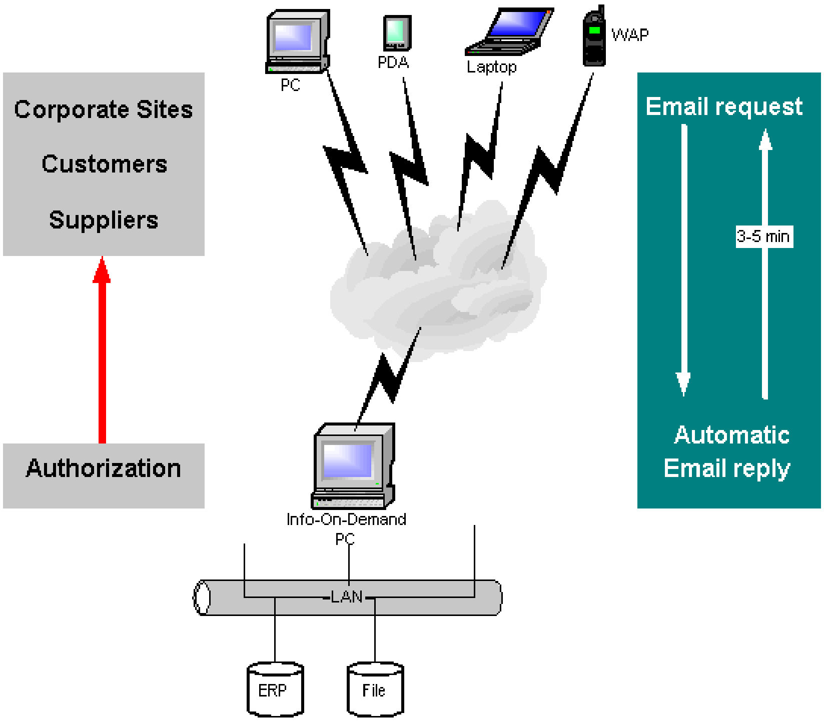
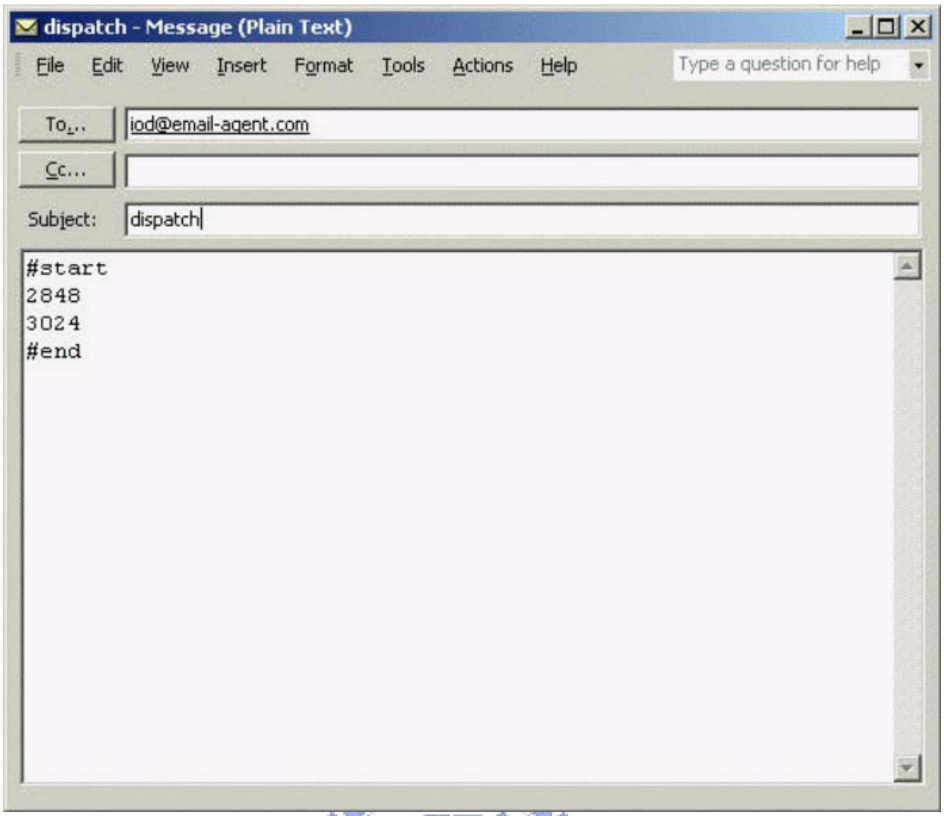


Figure 3.1: The architecture of Info-On-Demand.

As Figure 3.1 shows, Info-On-Demand provides a 24-hour customized e-mail agent for users to fetch the data in the remote databases. When the user's e-mail was arrived, only few minutes later, the result e-mail will be sent back right away. It is a great solution for the mobile user to obtain his information from his private database in his company anywhere at anytime. However, until now, the system can only be applied in the database management system. They can't retrieve the information of the Internet at all. Furthermore, the client side's instruction set is too simple to execute the complicate works. Most of the works have to be coded in the server side's environment in advance. The flexibility of the system is limited.

A simple example for the contents of e-mail of Info-On-Demand is shown in Figure 3.2.



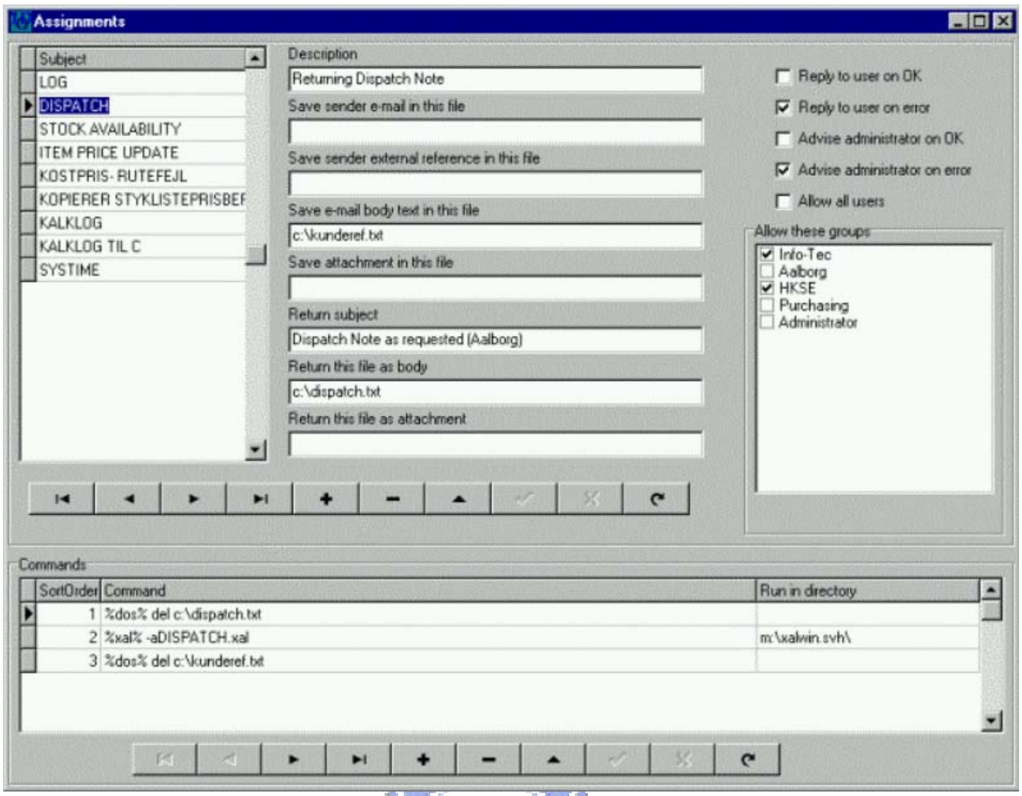
To: iod@email-agent.com

Subject: dispatch

```
#start
2848
3024
#end
```

Figure 3.2: The dialog for a simple query e-mail of Info-On-Demand.

Of course, as mentioned above, all of the built-in functions and user-defined batch programs have to be settled and registered in advance. The setting and register example of Info-On-Demand is shown in Figure 3.3.



Subject
LOG
DISPATCH
STOCK AVAILABILITY
ITEM PRICE UPDATE
KOSTPRIS-RUTEFEJL
KOPIERER STYKLISTEPRISBEF
KALKLOG
KALKLOG TIL C
SYSTIME

Description: Returning Dispatch Note

Save sender e-mail in this file:

Save sender external reference in this file:

Save e-mail body text in this file: c:\kunderef.txt

Save attachment in this file:

Return subject: Dispatch Note as requested (Aalborg)

Return this file as body: c:\dispatch.txt

Return this file as attachment:

☐ Reply to user on OK
☑ Reply to user on error
☐ Advise administrator on OK
☑ Advise administrator on error
☐ Allow all users

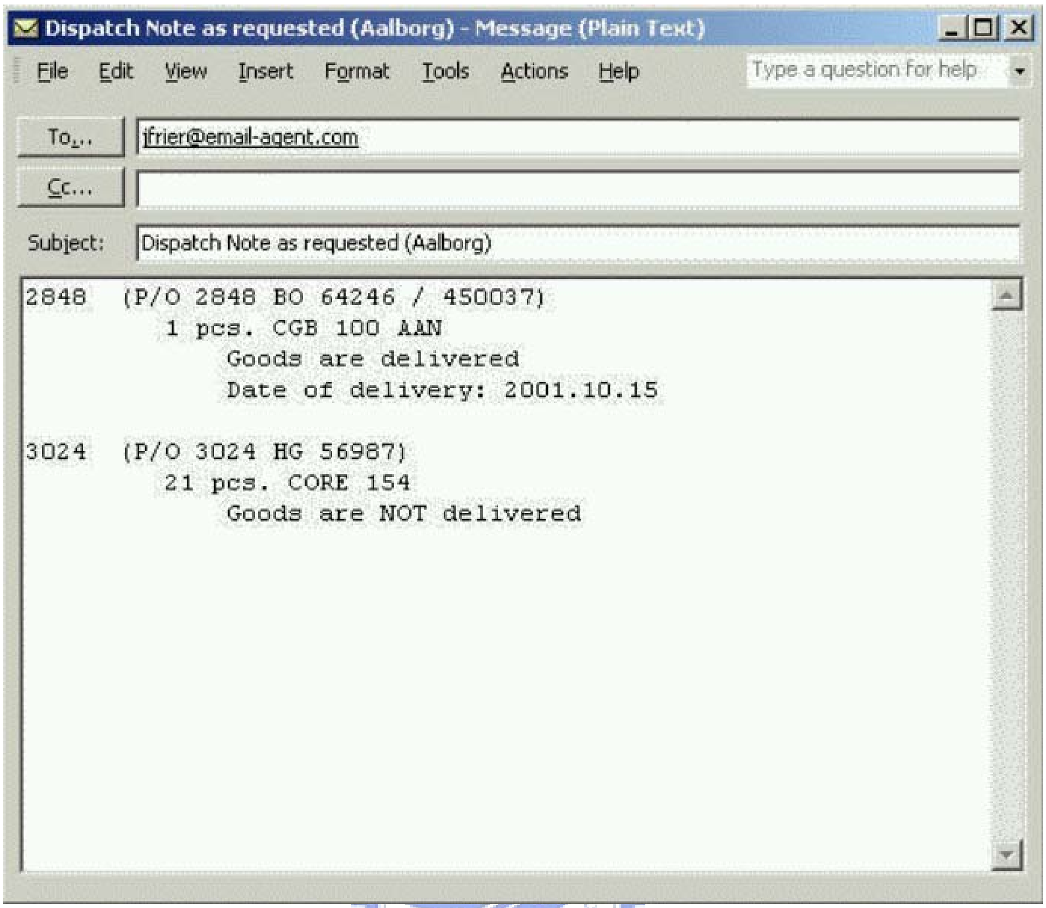
Allow these groups
☑ Info-Tec
☐ Aalborg
☑ HKSE
☐ Purchasing
☐ Administrator

Commands

SortOrder	Command	Run in directory
1	%dos% del c:\dispatch.txt	
2	%xal% -aDISPATCH.xal	m:\xalwin.svh\
3	%dos% del c:\kunderef.txt	

Figure 3.3: The setting and register dialog of Info-On-Demand.

As Figure 3.3 shows, the role of batch program is to set the actions for information retrieval. The information sources for the user to retrieve are all built in the internal database. Thus, the batch program can only to specify the data field name for fetching. The name of batch program is specified in the subject of the e-mail, and the contents of e-mail can be treated as the arguments of the batch program. Consequently, the fetched results will be wrapped into an e-mail which is shown in Figure 3.4.



To: jfrier@email-agent.com

Cc:

Subject: Dispatch Note as requested (Aalborg)

```
2848   (P/O 2848 BO 64246 / 450037)
          1 pcs. CGB 100 AAN
              Goods are delivered
              Date of delivery: 2001.10.15

3024   (P/O 3024 HG 56987)
          21 pcs. CORE 154
              Goods are NOT delivered
```

Figure 3.4: The fetched results from the Info-On-Demand system.

According to the mechanism mentioned above, the user can get the information of the databases which is located at his company in anywhere by any mobile devices (includes mobile phone, PDA, notebook, or even desktop PC and workstation) without any other extra application except simple e-mail application. Because the operation for the user is very simple and intuition, it is ideal for those people who always needs the real-time data at anyplace. However, so far, the Info-On-Demand has no solution for the information retrieval on the Internet. Thus, it is not a general solution for advanced information retriever.

Let’s consider another aspect of the problem. Although the notebook PC and pocket PC is much smaller than the desktop devices, it is always a burden for many people. Many people always hope that the handheld terminal is small and powerful. However, in most cases, the small size always means the resource-limited, or the very expensive price. Is there exists a solution for low-end handheld terminals to access the Internet? iSMS is a typical approach which is dedicated on this idea.

3.2.3 iSMS

iSMS [40] is an integration platform for short message service and IP networks. This is a mechanism for combination between the Internet and GSM network of mobile phone. Utilizing iSMS, the user may direct access this mobile phone by composing and sending SMS to the phone number of iSMS server. Some built-in program of the server will be activated for the user's requests, and then send the results back to the user's mobile phone through SMS protocol. Moreover, the user can also make a schedule in advance. Let the iSMS server actively push some periodic information to the user through SMS periodically. All the works of iSMS do not require any modification to the mobile telephone system architecture, because the SMS is the basic service for all the GSM mobile phone networks. All cellular phones of GSM will provide SMS as well. Moreover, the other mobile phone standards like DAMPS (Digital Advanced Mobile Phone Service), CDMA (Code-Division Multiple Access) are all providing SMS too.

One of the major advantages of using SMS to be the transmission protocol between mobile phone network and IP network is that the mobile phones can be considered an "always-on" device that facilitates instant information exchange. No dialup operation is required to access SMS. Furthermore, SIM itself provides the storage for short messages, and the SMS will not occupy the voice, data, and fax services of GSM. The Figure 3.5 shows the basic architecture of iSMS [40].

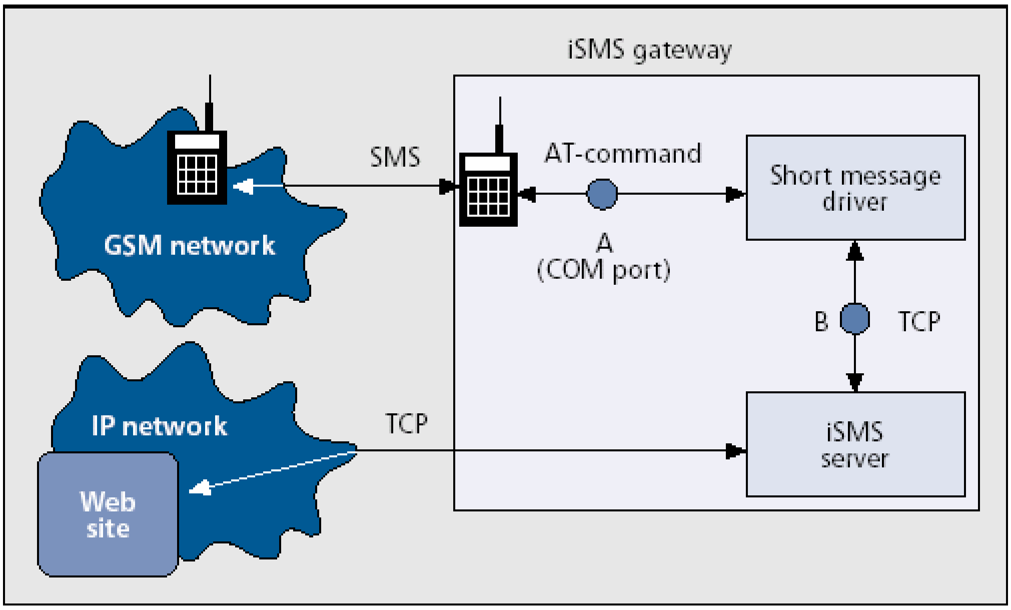


Figure 3.5: The basic architecture of iSMS.

As shown is Figure 3.5, the iSMS gateway basically requires a mobile phone with SMS service and a computer with the ability for TCP networking. The mobile phone is as the interface between the GSM network and iSMS server, and the iSMS server is response for accessing the TCP/IP network. In this way, the user of GSM network may easily access all the Internet services of TCP/IP network through the iSMS gateway.

The major advantage of iSMS is the utilization of current SMS technology. The SMS is a very basic service of GMS system. All of the GSM mobile phones are equipped the functionality for sending and receiving the messages of SMS. That is, all of the GSM mobile phones can be the terminals of iSMS. The major disadvantage is that the size of SMS is very small, and the transmitting cost for SMS is very expensive. In Taiwan, it costs NT$ 2-3 dollars per SMS message. For larger amount information retrieval, the SMS-based solution is not suitable.

Furthermore, although the SMS is intuitional service for mobile phones, it is not the basic service of PDA and desktop computer. To unify to interaction between people and devices, the e-mail is the better solution than SMS, much less the e-mail can be implemented by SMS in mobile phones.

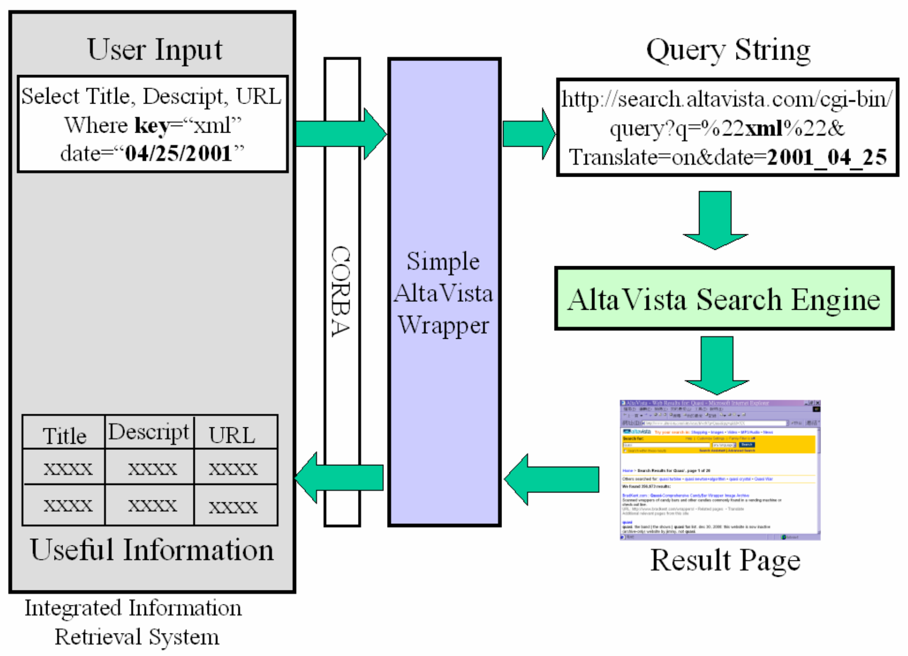
3.2.4 AvantGo

AvantGo is an acceptable solution for the resource-limited mobile device to obtain the information and documents on the Internet. Actually, it provides thousands of specially formatted brand-name web-sites ("channels") on user's PDA or smartphone. These contents of channels include news, weather, sports, stock quotes, maps, movie listings, and more. If a user wants to obtain the information from these channels, first, he has to sign in as a member of AvantGo, and then create his own custom channels to download family sites, local news, niche sites, church and club sites, workout schedules, etc. The great feature of AvantGo is that no wireless connection is necessary. However, this service needs a desktop PC to sync with the user's mobile device. That is, the mobile device of the user has to sync to the user's desktop PC for downloading the subscribed channels in advance. Meanwhile, all the information on the subscribed channels will be downloaded at once.

Obviously, the service from AvantGo is an indirect way for the user to obtain the information on the Internet. It lacks an interaction between the user and the information sources. Furthermore, the process for transferring information needs a desktop PC, the information retrieval model is not suitable for our approach.

3.2.5 Unified interface in integrated information retrieval

To efficiently and comprehensively retrieve the information on the versatile information sources of the Internet, the meta-search engine technology is one of the most important research trends. For ease to interact with the meta-search engine for the users, and for ease to adapt the new information sources for the system developers, the media with wrapper architecture is the most popular technologies. Figure 3.6 shows one of the typical workflow of meta-search services by wrapper technologies.


User Input
Select Title, Descript, URL
Where **key**="xml"
date="**04/25/2001**"
CORBA
Simple AltaVista Wrapper
Query String
http://search.altavista.com/cgi-bin/query?q=%22**xml**%22&Translate=on&date=**2001_04_25**
AltaVista Search Engine
Result Page
Useful Information
Integrated Information Retrieval System

Title	Descript	URL
xxxx	xxxx	xxxx
xxxx	xxxx	xxxx

Figure 3.6: The typical workflow of meta-search services approach in ref [18].

As Figure 3.6 shows, an integrated information retrieval system accepts some kind of high level information retrieval commands. In this example, the user inputs some kind of SQL-like querying string. It will parses these commands and dispatch some well-format queries to each specified search engine or information source according to the contents of user's high level commands. The main works for translating the high-level commands to search engine's queries are done by wrapper. Of course, the wrapper is also response for accepting and translating the results returned from those search engines. In such approach, the wrapper plays a very important role as the interface between the retrieval system and information sources.

However, there are different retrieving method and commands encoding between different search engines and information sources. Almost each connection with one specified search engine or information source needs one different specified instruction sets. For an integrated information retrieval application developer, to find out the retrieving rules and write some codes by hand to adapt their system to a specified search engines is a time-consuming work. By the way, to adapt a new information source easily and quickly, the wrapper generator technology is important. Paper [18] has proposed such a semi-automatic wrapper generating solution.

Paper [18] presents a design for an automatic XML-based framework with which to generate wrappers rapidly. Wrappers created with this framework support a unified interface for a meta-search information retrieval system based on the Internet Search Service using the CORBA standard. Greatly advantaged by the compatibility of CORBA and XML, a user can quickly and easily develop information-gathering applications, such as a meta-search engine or any other information source retrieval method. The two main things our design provides are a method of wrapper generation that is fast, simple, and efficient, and a wrapper generator that is CORBA and XML compliant and that supports a unified interface. Figure 3.7 shows a simple example of paper [18].

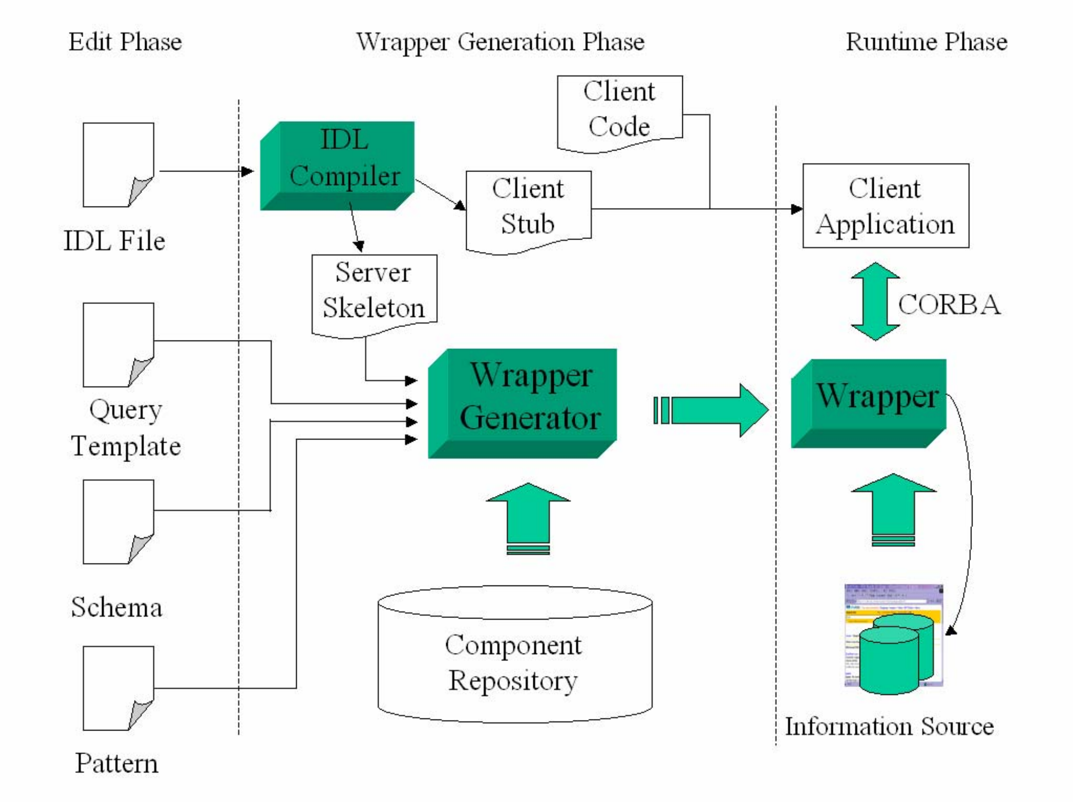


Figure 3.7: The role of a semi-automatic wrapper generator.

Through the wrapper generator shown in Figure 3.7, a user may enter a more complicate query string for querying some interested information from many information sources on the Internet. The system will parse the query string and dispatch the appropriate works to analyze and collect the requested information through an appropriate wrapper. After the requested information was fetched and analyzed, the information will wrap them as the appropriate format, and send them back to the specified user.

Utilizing the wrapper-generator technology, the system developer of integrated information retrieval application now may not worry about the versatile information sources on the network. However, the versatile of terminals that the users used is also a big trouble. Thus, we need a solution to conquer the differences among all of the terminals and mobile devices. The popular e-mail protocol is the best candidate.

3.2.6 Mobile agent runtime environment in mobile computing

Reference [45][47] are trying to propose the solutions providing the mobile agent to run at mobile devices, but they all need the mobile devices to run some kind of mobile agent runtime environments. That is, on the client side, installing an agent runtime environment for sending and receiving mobile agents is a very heavyweight approach. In Reference [45], the client side applications become capable for sending, receiving, and interacting with mobile agents by using well-defined Java Beans software components. These components are included in the application, allowing it to become agent-enabled. The authors of Reference [45] develop a component middleware which is included in the client applications. To implement this approach, the client side has to support a well-structured Java runtime environment. Unfortunately, most of the terminals and mobile devices are still not supporting the Java runtime environment. The situation of Reference [47] is the same.

3.2.7 Current commercial products for remote information accessing

Many commercial products (such like Sybase iAnywhere, Oracle 9i Lite …, etc.) provide the abilities for mobile devices to access their own database on the specified platforms and environments (PocketPC, Palm OS, EPOC, etc), but these products are hard to perfectly implement in the rigid environment of mobile computing, and moreover, they do not provide the ability for user to access the information on the Internet.

Chapter 4

Mobile Agent and Integrated Information Retrieval

As mentioned earlier, there are many researches applying agent technologies, and mobile agent technologies to improve the performance for information retrieval. Moreover, there are also many researches trying to apply some meta-search mechanisms to enhance the covering ratio for the user to do the information retrieval works. These related technologies are described in this chapter in detail.

4.1 Mobile agent concepts and their advantages on mobile computing environment

There are so many researches dedicated on agent technologies [3][5][6][26][27][29][30][31][32][33][34] and mobile agent technologies [7][8]. In those researches, they proposed many different mobile agent environments and implementations, some of those researches are successfully applied to many areas of computer science and networking systems. Some of them also present modified programming languages or new designed programming languages for mobile agents. This chapter will describe the basic concepts of mobile agents for mobile computing environment, and will discuss the considerations about applying mobile agent technologies to mobile computing environments.

Mobile agent technology is not a new idea. The concept of mobile agents is developed from intelligent agent. The definition of intelligent agent is adapted from [34]:

> *An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives.*

Moreover, reference [6] further points out that the intelligent agent is capable of flexible autonomous action in order to meet its design objectives, where flexibility means three things [34]:

- *reactivity*: intelligent agents are able to perceive their environment, and respond in a timely fashion to changes that occur in it in order to satisfy their design objectives.

- *pro-activeness*: intelligent agents are able to exhibit goal-directed behavior by taking the initiative in order to satisfy their design objectives.
- *social ability*: intelligent agents are capable of interacting with other agents (and possibly humans) in order to satisfy their design objectives.

To satisfy the above three characteristics, the platform which performs the intelligent agents should be a powerful environment. Such environment at least has to provide the ability for executing the autonomous codes of agents, and a continuous and stable network connection or interacting mechanism for the agent to interact with their environment. Of course, for the mobile devices which are circumvented some hardware constraint (CPU cycles, narrow or expensive bandwidth, small memory, and limited user interface) [48] are hard to satisfy the above requirements. However, the mobile agent technology can partly improve the situation.

A mobile agent is not bound to the system where it begins execution. The mobile agent is free to travel among the hosts in the network. A formal definition in [6] is shown as the follows:

> *A mobile agent is not bound to the system where it begins execution. It has the unique ability to transport itself from one system in a network to another. The ability to travel allows a mobile agent to move to a system that contains an object with which the agent wants to interact and then to take advantage of being in the same host or network as the object.*

That is, the mobile agent is indeed a software program. In most cases, its size is far smaller than the normal application of Windows. It can be executed in different host. Furthermore, in most cases, there is at least one mobile agent manager can dispatch, activate, suspend, stop, migrate those agents. Of course, the mobile agents can migrate, or even destroy themselves too. In a good implementation of mobile agent environments, the agents can be suspended or migrated at any time and any place. That is, a mobile agent may be suspended by manager or itself. In this time, the system will preserve all the states of the execution so far. Meanwhile, the system will pack the codes and execution states, then transfer them to the target host. After the codes and states have arrived, the mobile agent will be resumed to execute at the break point of last execution. The mobile agent is aware of the migration between hosts, and will continue running the same project across some different hosts. The concept of mobile agent is shown in the Figure 4.1.

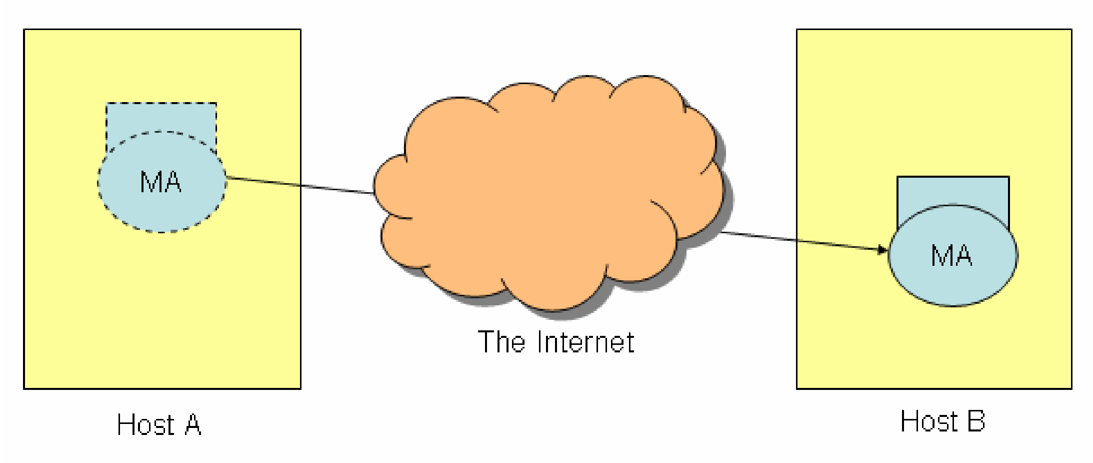


Figure 4.1: The basic concept of the migration of mobile agent.

As shown in the Figure 4.1, the contents of a mobile agent (MA) are not only includes the codes, but also includes the states of execution this time. MA may migrate itself from Host A to Host B. In most cases, MA may aware of occurrence of the migration, and will resume the execution from the last break point for migration. The things that have to known are the Host A and Host B can be the different host connected by the Internet or local area network. Of course, the host should all have an appropriate mobile agent execution environment. In an ideal situation, the mobile agent can migrate itself to another hosts at any point during the execution time. It is depicted in Figure 4.2.

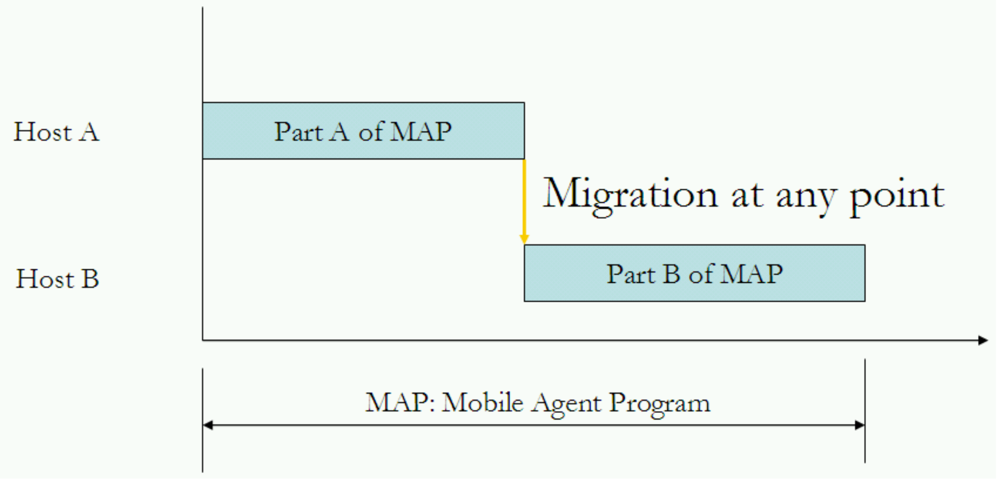


Figure 4.2: The ideal mechanism for mobile agent migration.

As Figure 4.2 shows, a mobile agent program (MAP) which is run at Host A may migrate itself to Host B at any point during the execution time. That is, there is no extra effort for

MAP while it is at migrating process. The part B of MAP will soon be executed after the migrating process. All the execution state and related data of MAP will also be migrated from Host A to Host B. Figure 4.2 shows a very ideal situation. It is very hard to achieve. Another acceptable and common migration mechanism is shown at Figure 4.3.

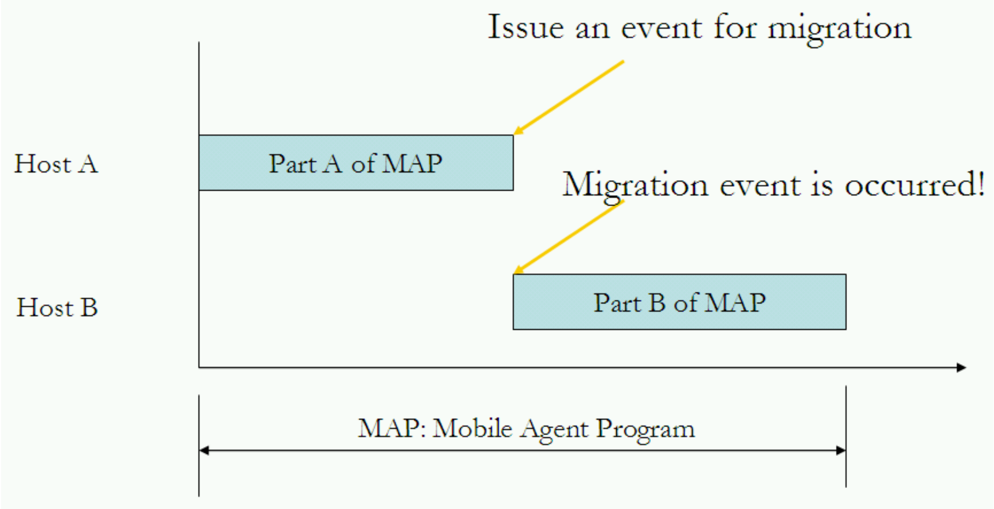

Issue an event for migration
Host A
Part A of MAP
Migration event is occurred!
Host B
Part B of MAP
MAP: Mobile Agent Program

Figure 4.3: The popular event-based migration mechanism.

Figure 4.3 shows a very popular event-based migration mechanism for mobile agent. In this mechanism, if the MAP wishes to migrate itself from Host A to Host B, it has to throw a migration event. After the system is aware of the occurrence for migrating request from MAP, it prepares all the works about migrating process, and then transfers the whole MAP and its process state from Host A to Host B. At the Host B, the migrated MAP will be executed from the start entry point of the program. Because the MAP is an event-based program, now the mobile agent runtime environment is aware of the occurrence of migrated event. Thus, the control flow of the program will soon be redirected the codes which is response for migrated process, and this process can take care all the chores about the migration. This mechanism is popular. Many Java-based approaches are used this mechanism to migrate their mobile agent.

The third mobile migration mechanism is e-mail based migration mechanism. It is shown in Figure 4.4.

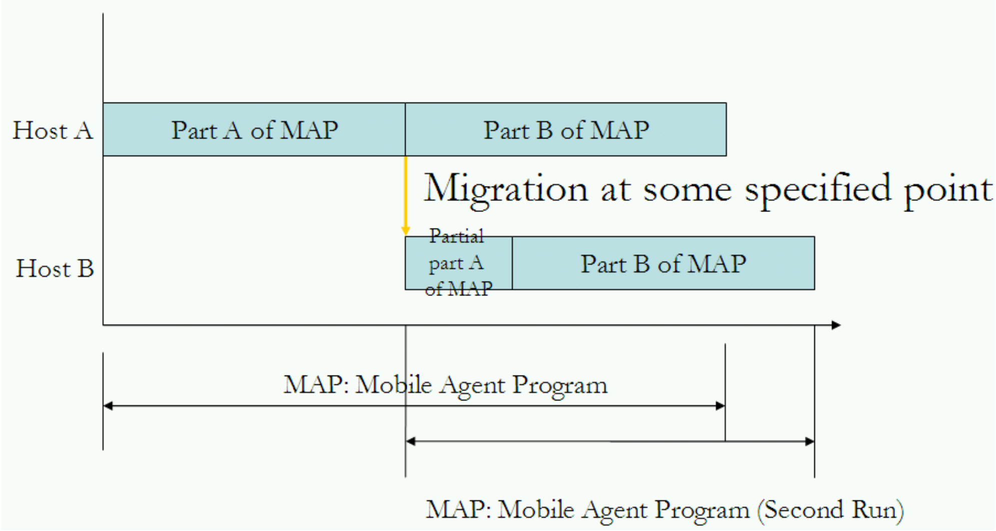

Host A
Part A of MAP
Part B of MAP
Migration at some specified point
Partial part A of MAP
Host B
Part B of MAP
MAP: Mobile Agent Program
MAP: Mobile Agent Program (Second Run)

Figure 4.4: E-mail based migration mechanism.

As Figure 4.4 shows, if a mobile agent which is executed at Host A wants to migrate itself to Host B, it has to prepare an operational e-mail to the system due to the Host B is lacking a daemon which can accept the incoming migrating request. After the system of Host B has received the operational e-mail, it will parse the contents of this operational e-mail, and extracted the mobile agent codes and its status to execute at Host B. Because the environment of Host B is different to Host A, the mobile agent will soon aware the occurrence of migration. Therefore, the mobile agent will adjust the control flow to part B of MAP. This is why the system will execute some part A of MAP which is depicted in Figure 4.4.

Applying the migrating mechanism which is shown in Figure 4.4, the mobile agent runtime environment of Host A and Host B are not required to be a specified running daemon. All the works can be done by e-mail protocol. In fact, the message redirection mechanism of sendmail can be applied to this mechanism. That is, each host which wants to provide the mobile agent runtime environment is just enabling the sendmail daemon. It is very easy to install for the lots of UNIX-based servers. Thus, this kind of mobile agent environment is very easy to deploy for the system developer. Our approach is applying such mechanism.

Because of the great ability of migration for mobile agents, the major advantage for applying mobile agent is to dramatically save the usage of network bandwidth. For example, suppose a company has a CTS (Customer Transaction Server) to store the each transaction of customers, an IDS (Internal Database Server) for storing the internal information of the company. The objective is that we need another DMC (Data Mining Computer) to comprehensively analyze recent databases both CTS and IDS. In most cases, the DMC will fetch the data in CTS and IDS through the networks to the local storage of DMC, and then analyze these data locally. This is an accepted situation when all these computers are in the same LAN (Local Area Network), but when these servers are dispersed on the WAN (Wide Area Network) or even on the Internet, especially when the network bandwidth is not enough to carry the great amount data, the mobile agent mechanism will be a good solution for the situation.

In the case mentioned above, the system implementer can install the mobile agent environment for each server. Thus the mobile agent can be migrated across these servers as user's goal. In such situation, we may design a mobile agent for data mining and dispatch it to either CTS or IDS. Then the mobile agent may retrieve and analyze the information of the CTS of IDS locally. Furthermore, for improving the efficiency of data mining, the user may dispatch more than one mobile agent across the network to do the same works, and finally combine the returned results of the mobile agents.

How to enable the ability for providing the mobile agent to remotely execute at a computer? Based on the consideration of security, most computers do not provide the ability for supporting the remote execution of programs. Thus, we have to install a specified environment as a gateway for executing of mobile agents. The possible solutions are shown as follows:

- ***Running a daemon under operating system***: In the solution, the system developer runs a daemon program under any operating system. It acts as a mobile agent executing server for accepting the client's requests. The client can send a piece of program or runnable program image to the server; the server then will compile and run the program pieces. The daemon will collect the results to client by the redirect characteristics of standard input/output for most modern operation systems. This is an ideal solution for mobile agent runtime environment. The main disadvantage of this solution is the serious security problem. To directly run an un-verified external program is very dangerous for a server. It is more dangerous for those servers which were directly supported by the operating system. Any malicious program may let the server run into a big disaster. Thus, many researches design a whole new safe script language, or apply the present safety programming language, such like Java, to implement their mobile agent runtime environment framework. Fortunately, the Java runtime environment applies some security mechanism to prevent the latent security problems.

- ***E-mail based mobile agent runtime environment***: The other major solution is much the same as the first method. The only different is the daemon of this method will not directly receive the request from the client. Instead, it will check the specified mail account. If the daemon has found a specified subject of e-mail with some kind of format and identification, it will parse and filter e-mail to extract the piece of program for executing. Again, the executing results will be collected by the redirect characteristics of standard input/output for most modern operation systems. These results will be packed as an e-mail, and sent back to the client's mailbox. This is an off-line solution for mobile agent mechanism. The mobile agent runtime environment of this solution is divided into two parts. The first part of the environment is a passive program which can be activated by the incoming message. The other part is the mobile agent execution environment which is managed by the first part. That is, the first part of the solution will take care of that operational e-mail, and dispatch those extracted results to the second part. All the results of the second part will be pack into an operation e-mail and they will be sent to the initiated system.

The first solution of mobile agent environment is running a daemon as a gateway. The gateway will retrieve the mobile agent program from the client's direct requests. The retrieved mobile agent program will be dispatched to the operating system for execution. To execute mobile agents in this way has at least two defects. First, if there is no appropriate security management mechanism, after the agent program is executed by the operation system of the computer, in general, the program will get all the trusts of the operation system. Most of the resources of the computer will be allocated to the program too. This is very dangerous for the possibility of attacks from a malicious program.

The second defect is that if the mobile agent is programmed by some kind of normal traditional procedural programming language, it is hard to suspend the execution of the mobile agent, migrate it to the other host, and resume the execution from the previous break point. If we want the agent can be resumed for execution from the suspended point, it will come with lots of works have to take care. All these derived works have to be done by the mobile agent itself. There are very heavy extra works for the system developers.

For mobile devices, the e-mail based resolution is much prefer. Most of mobile devices have no enough computing power and stable network connection to execute a complex application for information retrieval and other advanced applications. Thus, the better solution is to execute most part of works to a desktop server. The role of mobile devices is to be the front-end of the information retrieval process. The basic idea had been shown in Figure 2.5. However, there are more and more mobile devices supporting Java runtime environment. Thus, the Java based mobile agent solution is more popular nowadays.

There are lots of researches dedicated to solve this problem [6][7][8][34]. Utilizing the characteristics of high compatibility and high mobility of Java language, the IBM Aglet Project provides an event-oriented execution environment, let the agent program code can be migrated to another host, and trigger a dispatch event, and let the migrated agent aware the migration and can resume to execute the program from this point. The implementation of Aglet will be described at section 4.4.

4.2 Migration for e-mail based mobile agent across mobile devices

To be a good mobile agent runtime system for mobile computing environment, the system should at least have the following characteristics:

- Easy deployment both at server and client side.
- Providing the complete execution environment for mobile agent, not limited by the operation system of mobile devices.
- Providing a safely executing environment for mobile agent, including on the mobile devices and servers. Thus, the systems and the mobile agents will be protected safely.
- Providing a high-performance executing environment.

At the server side, many platforms today can satisfy the above requirements in hardware consideration. About the software, especially considering the operation system, the modern multi-user and multi-tasking operating system is much prefer. In this dissertation, we choose the Microsoft Windows 2000 server and RedHat Linux as the test bed. To fulfill the easy deployment requirement, we choose the e-mail protocol as the interface between the users and the mobile agents. In such case, no matter what the mobile agent is, and no matter where the mobile agent at, if a mobile agent wants to migrate itself to another host, it may just wrap itself into an operational e-mail, and sends it to its target e-mail account. That is, except a suitable operating system and suitable hardware, an e-mail server or daemon is required. In our dissertation, we choose the Sendmail^{TM} as the implementing tools. There are two possible models for the e-mail based migrating mechanisms: (1) distributed model, and (2) centralized model. The first model is depicted at Figure 4.5.

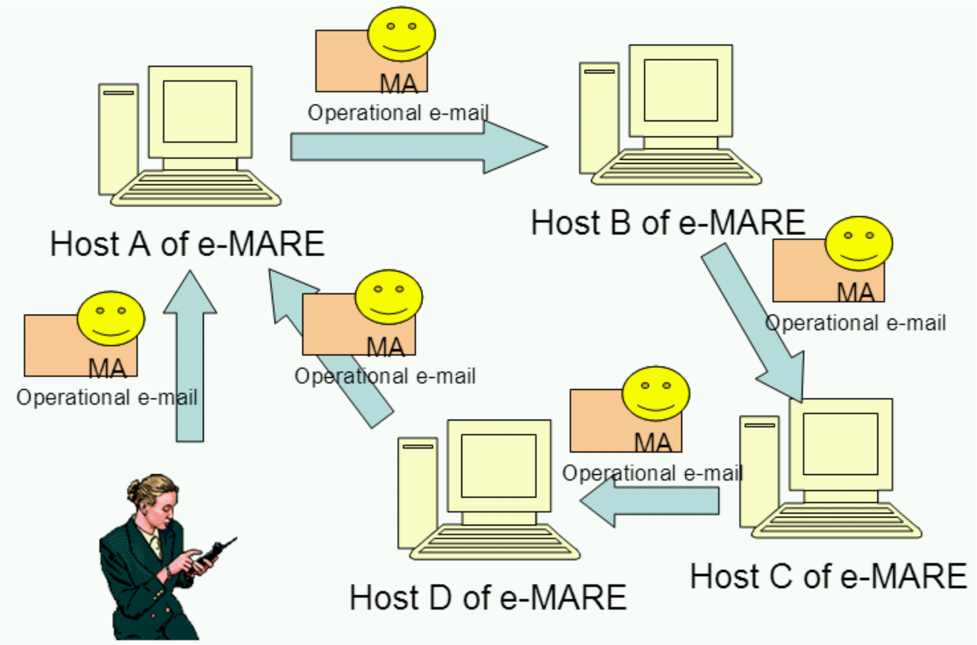


Figure 4.5: The distributed model of e-mail based mobile agent migrating mechanism.

As shown in Figure 4.5, it is a distributed model of e-mail based mobile agent migrating mechanism. In this model, every hosts (in this example, host A to host D) are all installed e-MARE *ManagerAgent*. The MA which is initiated by the user via a mobile device is sent to Host A. If the MA decide to migrated itself to host B, it may just wrap itself into an operational e-mail, and then sent itself to Host B via e-mail protocol. The *ManagerAgent* of host B will now take care of the other chores. Of course, the same way will be applied into host C and host D. Thus, the MA will travel from host A to host D, and finally back to host A by the same way.

However, by default, every e-mail will be directly replied back to the real sender. Thus, the *ManagerAgent* has to identify the desired destination of the operational e-mail. In our implementation, we design an argument called ‘X-Target’ to achieve the migration purpose. That is, every operational e-mail which wants to migrate itself to different host has to specify the ‘X-Target’ argument in the codes of the embedded program, and then send itself to the other host which is installed an e-MARE environment. Thus, after the specified works have been done, the *ManagerAgent* will issue the results or anther programs to the specified e-mail accounts of the remote host. A simple PHP example is shown as the follows:

```
#!/usr/bin/php -q
<?php
   $to = 'emare@mail2.twvs.tnc.edu.tw';
   $subject = "&&SDB;";
   $message = “mail body, may be some programs";
   $email = 'minhuang@mail.twvs.tnc.edu.tw';
   mail($to,
          $subject,
          $message,
         "From:$email\r\nReply-to:$email\r\n".
         "X-EMARE:e-MARE v1.0a\r\n" .
         "X-Target:minhuang@mail.twvs.tnc.edu.tw");
?>
```

The centralized model is depicted in Figure 4.6. It is different to the first distributed model. The centralized model assumes that only one host is installed both the e-mail service and e-MARE *ManagerAgent*. Thus, all hosts, include host B to host C, have to contact the host A if it wants to migrate itself to other hosts.

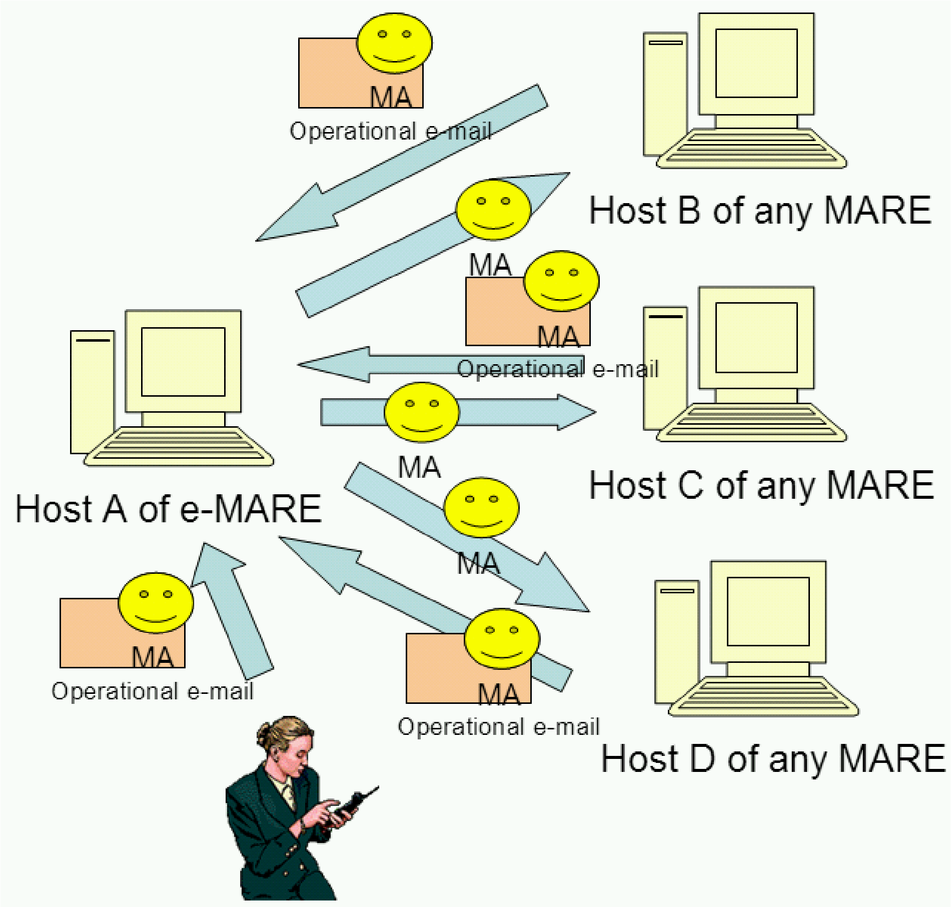


Figure 4.6: The centralized model of e-mail based mobile agent migrating mechanism.

As shown in Figure 4.6, the user sends an operational e-mail to the host A, the *ManagerAgent* of host A will parse the operational e-mail from the user. If the mobile agent which is executed in host A determines to migrate itself to host B, because the host B is not the server of e-MARE, it should follow the requirements of the MARE (Mobile Agent Runtime Environment) of host B. That is, the mobile agents in host A must follow the programming criteria of MARE of host B, and connect to host B by the API provide the MARE in host B. In the situation which is shown in Figure 4.6, the type of MARE among host B, C, and D may be different, if the mobile agent B wants to migrate itself to host C with different MARE system, it can just send itself to host A via e-MARE, let the host A as the gateway for host B and host C. The same mechanism can be applied host D too.

Conversely, at the client side, many platforms of mobile devices are resource-constraint. That is, they always can not fulfill the above requirements. Fortunately, in our dissertation, we conquer this situation by according to the following criteria:

- The role of mobile device is only to be a dummy terminal of the framework.

- All works are executed at back end server.
- All of the needed information are stored at database of the Internet.
- The database of the information is accessible from the Internet by HTTP protocol.
- To simply the input of terminal, all returned information are all specified in a serial number for identification.
- All the requests of the user can be specified by commands or identification number, the identification are corresponded to the information of the database.

Due to the resource limitation of mobile devices, one of the most important issues for mobile devices to retrieve information from the network and the Internet is to reduce the efforts as least as possible. Of course, we assume the mobile devices used in our approach are all network enabled. Currently, most of the networked mobile devices allow the mobile users to send and receive the e-mail on the Internet. Thus, the e-mail protocol is the best candidate for the mobile device to connect to the Internet.

Today, however, most of the e-mail applications of mobile device are very simple. They can only recognize the standard text format of e-mail, the extra functions such as HTML syntax, attachment processing, script language implementation, etc. are not supported. In such the case, the system can not directly execute the program which is embedded in the operational e-mail.

4.3 The idea of hybrid mobile agent runtime environment

We classify the mobile agent runtime environment into two categories, the on-line mobile agent runtime environment and the off-line one. The traditional mobile agent system is always an on-line system. That is, no matter what the processing node is at local or at remote, they always have to keep the services available. Both of the client and server have to activate a specified service or daemon for sending to or receiving from the agents. This is an acceptable solution for a desktop PC and server; but not suitable for a mobile device platform.

On the mobile devices, the mobile agent runtime environment can be treated as the off-line and passive architecture. In the on-line runtime environment, the mobile agent stays on an environment with well-work networks and high computing power. In such case, migrating mobile agent from one host to another is easy and fast. On the contrary, the mobile agent on the off-line runtime environment stays on the mobile device with unstable networks and low computing power. In the cases, the network connection is usually disabled. Consequently, the mobile agent can not be executed on such mobile devices. They are always stored in the mailbox of those mobile devices, and waiting the user to manually activate them by sending them to some activating server.

The off-line mobile agent architecture can minimize the utilization of expensive wireless networking for low-end mobile devices, but keeps the flexibility of mobile agents, while the on-line mobile agent architecture can maximize the efficiency of the system's performance. Getting the advantages of both types of mobile agent runtime environments is to combine these two architectures into a framework. The proposed framework is a hybrid architecture of both type mobile agent runtime environment for mobile users and shown as Figure 4.7.

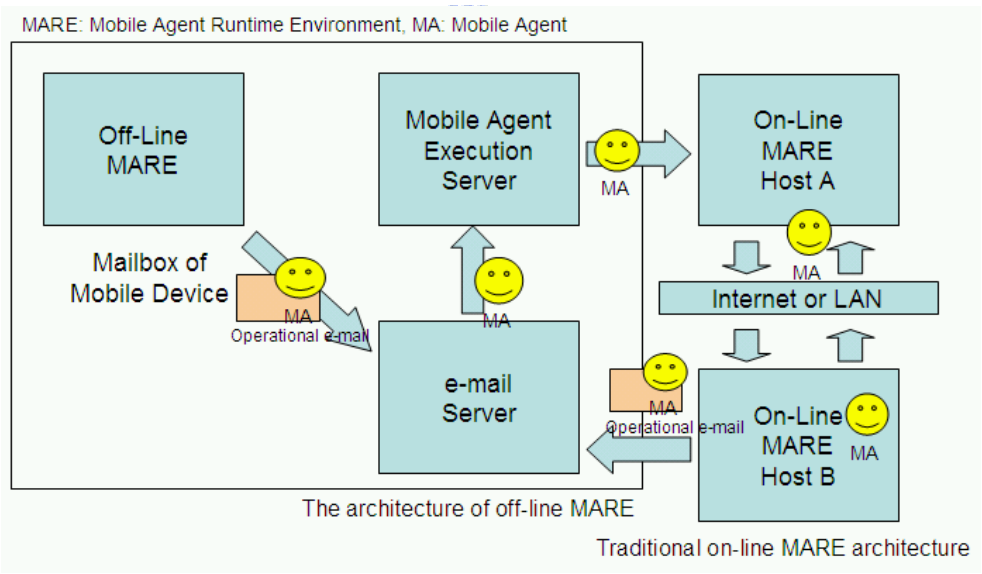

MARE: Mobile Agent Runtime Environment, MA: Mobile Agent
Off-Line MARE
Mobile Agent Execution Server
MA
On-Line MARE Host A
Mailbox of Mobile Device
MA
Operational e-mail
MA
MA
Internet or LAN
e-mail Server
MA
Operational e-mail
On-Line MARE Host B
MA
The architecture of off-line MARE
Traditional on-line MARE architecture

Figure 4.7: The hybrid architecture between on-line and off-line MARE.

The left side of Figure 4.7 is the typical off-line mobile agent runtime architecture, and the right side is the traditional on-line mobile agent runtime architecture. As shown in the figure, the mailbox of mobile device is a pool for storing the mobile agents, and it can be treated as a typical off-line mobile agent runtime environment. Each mobile agent embedded in an operational e-mail that is stored in the mailbox of mobile device is not alive until the user pressed the send button of the e-mail application. After the user sent that operational e-mail, this e-mail will be sent to the e-mail server of e-MARE framework. Then, the e-mail server will activate a manager program to parse the operational e-mail, to extract the mobile agent, and to dispatch it to mobile agent execution server for executing. The mobile agent in the mobile agent execution server also can be dispatched to the available host of some kinds of on-line mobile agent runtime environment. Of course, the mobile agent can start its itinerary among the available hosts of the on-line mobile agent runtime environment to extensively retrieve the user desired information on the Internet. Finally, the results collected by mobile agent can be returned to the user's mailbox by packing these results into an operational e-mail.

From the mobile agent point of view, a mobile agent in Figure 4.7, initially, is stored in the mobile user's mailbox of her mobile device. It is embedded with an operational e-mail. The user may edit this operational e-mail and sent it to the main service account of the system. By the way, the mobile agent is sent to the server altogether. After the parsing phase of the operational e-mail, the mobile agent will be executed in the mobile agent execution server independently, thus, the mobile agent can migrate itself to any on-line mobile agent runtime environment available on the network. The user's interested information can be returned by the running mobile agent to compose an operational e-mail and to send to the e-mail server of the system.

The key to combine these two different architectures is the mobile agent execution server. It is the main gateway between the on-line and off-line architectures. It is a full function execution platform that can run any mobile agent program or application on it. Thus, the user of mobile device has the full ability to execute the appropriate mobile agent program to activate the facility for mobile agent's itinerary, or to directly execute the information processing application for retrieving the useful information from the network or the Internet. To reduce the user's efforts, the system can also be preinstalled some useful programs for later use.

4.4 A unified interface for integrating information retrieval

As mentioned above, to intensively explore the information on the Internet, to provide a unified interface for integrating information retrieval is urgent. In this section, we will outline the approach of unified interface and a flexible architecture for querying various information sources on the Internet and the WWW using both a popular object model and a data model. We have proposed an Integrated Information Retrieval (IIR) service based on the Common Object Service Specification COSS for Common Object Request Broker

Architecture (CORBA) and apply the Document Type Definition DTD of eXtensible Markup Language XML to define the metadata of information sources for sharing the ontology between mediator and wrappers. The objective of using the IIR design is not only to provide programmers with a uniform interface for coding a software application that can query a variety of information sources on the Internet, but also to create a flexible and extensible environment that easily allows system developers to add new or updated wrappers to the system.

To provide this mechanism, it can achieve the following objectives:

1. To propose a uniform interface for information retrieval and gathering in an approved standard of distributed object-oriented environment. This offers a programming interface to retrieve what applications are wanted, and uses agent technology to implement the infrastructure of IIR.
2. Each type of information source has its own query language, schema and attribute. With this approach, it is necessary to support an extensible environment that will allow integrating various information sources in the future.
3. The Document Type Definition (DTD) announced by the World Wide Web Consortium (W3C) is a popular description language of scheme. We apply the DTD of eXtensible Markup Language (XML) to define the schema of information sources, and to provide the interface in the IIR for managing metadata.
4. Due to the unity of interface, a service provider can easily implement a wrapper for their speed up the system development.
5. Both the object model of CORBA and data model of XML are the approved standards and are widely accepted by the industry and so will be by users and programmers.
6. We adopt Structure Query Language (SQL) in the IIR for transparently querying information from various sources on the WWW and the Internet. IIR can seamlessly combine references to the Web with references to the relational database. Anyone familiar with SQL can create programs using IIR easily.

A flexible architecture and framework can improve access transparency, system scalability and extensibility. We dedicate our IIR design to such participants as data providers and information inquirers who can dynamically join the system flexibility, no matter what types of information are concerned. Any information source can at any time be joined dynamically into the system. An IIR client can obtain the information about information sources by inquiring their metadata. A service provider can also replace, access, and maintain the metadata of information sources and provide an adaptable environment. Metadata management and the extensibility and scalability of system critically enforce the IIR. In addition, the approach also allows that the system should be capable of retaining the autonomy of a jointed local query system, that is, that the IIR and local query systems should co-exist.

IIR architecture is simple and complete. From the client's perspective, the requirements are a uniform access interface as well as unified data model for representing the results of queries. With IIR, clients use a standard query interface to acquire information based on a CORBA object model.

Figure 4.8 depicts IIR architecture. It comprises *InformatiionRetriever*, *MetaData*, *Wrapper* and *Collector*. *InformationRetriever* acts as a mediator for dispatching query requests to the wrappers of information source and collects the results. A client program sends a query request to the information sources by first obtaining the *InformationRetriever* object from a *Factory* object. Due to the access transparency, the operations for querying all information sources using IIR are the same. A client queries information sources by mean of invoking *InfomartionRetriver*. The *InformationRetriever* activates corresponding wrapper(s) according to the query string involved in the parameter of query operation. It is obvious that clients accessing information sources are completely transparent by invoking an *InformationRetriever* object.

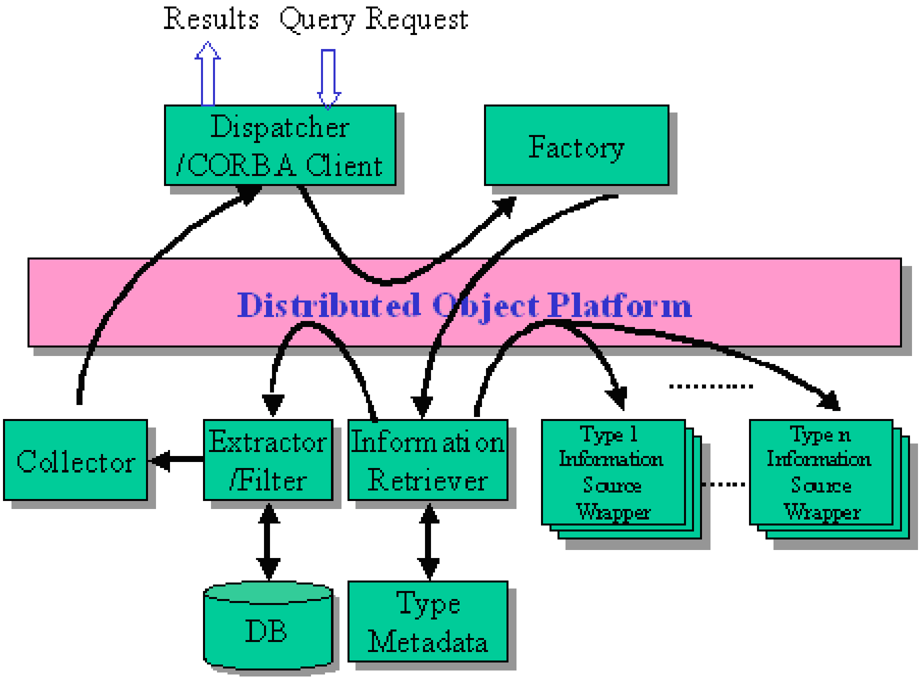


Figure 4.8: Architecture of IIR.

MetaData is the management of metadata. Its purpose is to minimize the degree of complexity fro federating heterogeneous information sources. With IIR, it has three following functions. First, a client can query the *MetaData*, construct a world-view and formulate the query string when it is unfamiliar with the schema and the semantics of accessed information sources. Second, the metadata is the ontology with respect to information sources. The *InformationRetriever* and the *Wrapper* shares the metadata in querying information sources and in translating the content of the query. *InformationRetriever* will refer to the metadata in judging the query string and determining the related wrapper when it receives a query request. *MetaData* provides the ability of access transparency for the IIR. Finally, IIR is needed to enable management of the metadata when the source dimension changes, that is, for example, for adding or deleting a wrapper of information source. Neither the query operations in client nor the objects in IIR are necessary to be changed. The *InformationRetriever* refers the metadata and judges the meaning of the query operation. Obviously, the IIR have extensibility and scalability. There is a need to have some supporting methods for the management of metadata in IIR.

The *Wrapper* is responsible for translating the query request into the request format associated with the information sources and the results from the local system data representation to the IIR system. If the results are from multiple similar sources, they are filtered. The wrapper activates the *Filter* object according to the kind of sources. The result is packed into a standard format, for example, XML, and put into the *Collector* object. Finally, the *Collector* collects the results and translates it into export view. For the client, IIR supports a unified invocation approach for querying source and obtaining results.

Owing to the benefits and the integration of various information sources that maybe configured on the WWW or the Internet, the approach adopts the SQL as the query language rather than invent a new one. In this way, the query language of IIR provides programmers with the illusion that the information sources are stored and organized in a relational database.

As we know, a schema describes the structure of a relational database, i.e. the tables, fields, and the relationships between them. Generally, a Web-based documents or a Web-based processing system involves a table, even if it has multiple backend physical database because it has a single interface to query inner data via the Common Gateway Interface (CGI) program. For example, search engine and biographical query system. Such the systems, we can suppose the whole system contains only a single table. The table name is defined as the service name. Some querying examples of this approach are shown in Table 4.1.

Table 4.1. Some querying examples of integrating information retrieval approach

Querying string	Purpose
SELECT * FROM Yahoo WHERE Keyword=”CORBA”;	Show the information about the “CORBA” from Yahoo.
SELECT URL FROM Altavista WHERE Keyword = ”MP3” and Tag = “text”;	Show the URL about the “MP3” from Altavista where tag is “text”.
SELECT Title and URL FROM Yahoo and Altavista WHERE Keyword = “Programming Language” and Keyword = “Object-oriented”;	Show the Title and URL for the “Programming Language” and “Object-oriented” fro Yahoo and Altavista.

The example shows that the query operating is invoked in two search engines – Yahoo and Altavista. IIR will accept the request and dispatch it to the wrappers associated with Yahoo and AltaVista, respectively. When obtained, the results from two wrappers are merged.

In addition, combining the second and third examples creates a problem in which a condition might conform to the rule of one search engine but not the other. For example, a user can query a certain keyword placed in the “Anchor” tag from the Altavista search engine, but may not be able to do so with Yahoo.

To facilitate the ability of our e-MARE framework, we will apply the IIR service in the back-end server of e-MARE. Thus, the e-MARE framework will integrate both the front-end and back-end for the integrated information retrieval works.

4.5 Agents technologies for information retrieval by applications

About in the last fifteen years, there are so many researches [4][9][10][11][12][13][14][15][16][17][19][20][21][22][23][25][26][28][39][42] dedicated in Information Retrieval. Some of those researches [5][26][27][29][39][42] utilize the agent and mobile agent technologies to retrieve the information on the Internet or over the network. Whether the stationary agents, mobile agents, or multi-agents, the user may delegate the agents to autonomously retrieve the information over the network. The only differences is that the mobile agent can migrate itself across many hosts. In many cases, the mobile agent's performance is better than stationary agents. Moreover, the mobile agents are not bound to the stable network connection.

Utilizing stationary agent to retrieve information over the network, the agent should access the remote information through the stable network connection. Meanwhile, the system environment for the stationary agent to execute should have enough computing power. For most mobile devices, especially for the mobile phones, it is an impossible mission for executing the mobile agents on the devices. The solution for mobile devices to execute the mobile agents is to send these mobile agents to remote servers. The role of mobile devices, especially the devices with limited computing power, is as the interface of the mobile agents. That is, the user may design, manage, and operate the mobile agents by the mobile devices. The mobile agents will really run at the remote sites instead of at the mobile device. Because most of the works can be done at the server, the traditional approaches for information retrieval on the Internet all can be applied in our framework.

Utilizing the agent as the information retriever is important. However, most of search engines and multi-search engines are developed only for WWW users, not for application programs that need to exploit data from the web. They also have no a uniform interface while accommodating new and powerful search engines in future, so that most multi-search engines are less the extensibility. We have proposed a uniform interface - Internet Search Service (ISS) [19] that follows the COSS of OMG’s CORBA [24] to solve the problem described above. And, an experimental ISS-based multi-search engine termed Octopus has been built. With that Octopus can accommodate new search engines easily and support application programs to exploit data from the Internet. Most returned results from search engines could be not useful for users even if these results are ranked higher index. Search engines with the function of personalized search are strongly necessary for experts. Some well-known search engines have supported this function, such as SavvySearch, MyYahoo etc. The main reason of personalized search is to offer most suitable query results to user.

In this section, we will outline the design and implementation for supporting personalized search in the Octopus. In Octopus, an absolutely irrelevant filtering approach used to support the personalized search. Moreover, in order to balance system load and user requirement, the filtering mechanism is divided into three levels – URL, Description/Context, and Content respectively. In addition, the feedback mechanism is in cooperation with the filtering mechanism to achieve the functionality of personalized search in a search engine.

This kind of search service is favorable for WWW user, not for application programs. Therefore, this function is independent of the ISS. To support such service is only to redesign the architecture of Octopus. The original advantages of ISS should be reserved in providing other functionality. All the interfaces will be not modified. The major contribution of the research is providing an approach to support the personalized search service based on the ISS without change its interface of the Octopus.

The ISS is designed by following the style of CORBA’s COSS. Its major goal is providing a uniform interface for most search engines. The details of ISS Octopus’s scenario please refer to [19]. In the section, we describe the personalized support on the Octopus.

In keeping the advantage of the ISS, a personal information-filtering agent is added into the Octopus instead of modifying architecture or interface of ISS when adding personal functionality. Figure 4.9 shows the preliminary design of the Octopus with personalized search. The major difference between this architecture with original Octopus version is adding a personal information-filtering agent that used to filter users favor. Such design philosophy is in order to reduce the search overhead when similar query is requested repeatedly.

The feedback mechanism is in cooperation with the filtering mechanism to achieve the functionality of personalized search. The latter used to find out adequate results, while the former let the user to respond what his/her favor is. In this approach, two mechanisms are adapted implicit feedback approach and absolutely irrelevant filtering approach respectively.

Using implicit feedback approach instead of explicit one is in order to go with the filtering mechanism properly. The absolutely irrelevant filtering approach is based on the custom of user in searching information from large amount of URLs and descriptions. In generally, users will first visit those deemed more suitable of URLs and skip the others that symbolize irrelevant. The visited web page may represent the page is interested by user in some extent. To analyze those fully irrelevant URLs or descriptions may find out more relevant to what don't he/she want than relevant approach and act as the filtering basis. This is the spirit of absolutely irrelevant filtering approach.

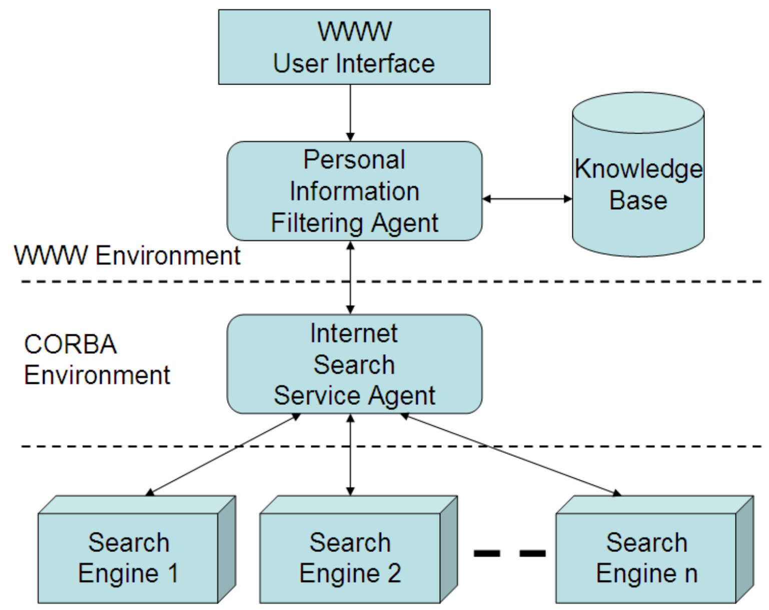


Figure 4.9: The preliminary architecture of supporting personalized search.

We can utilize classical Boolean model of Information Retrieval [1] to explain the concept. If $\vec{K}_i$ is the set of relevant terms and $\vec{K}_j$ is the set of non-relevant terms. Then $K = \vec{K}_i \cup \vec{K}_j$ will be all of terms that are included in returned results or documents. And $K - \vec{K}_j$ will be more relevant terms. So that we use absolutely irrelevant filtering approach to filter the non-relevant terms will get more relevant terms.

In the design, the personal information-filtering agent analyzes the factor of these non-relevant URLs or web pages and stores it into knowledge base for future searching requests. The knowledge base keeps personal filtering information. When the Search Engine Agent replies the user request, then the Information-Filtering Agent will filter the result in accordance with the personal filtering information.

The filtering mechanism is the corpus in supporting personalized search. In order to balance system load and user requirement, the filtering mechanism is divided into three levels – URL, Description/Context, and Content respectively.

1. *URL*: This is a simplest filtering level. This level filters the URLs of result that are selected and non-visited by user, into personal database. Those non-visited URLs imply absolutely irrelevant and act as filtering base for future searching request. Because this level is simplest, it has slightest overhead.

2. *Description/Context*: Almost all returned results of search engines consist of URL and description. In the level, filtering mechanism analyzes vocabulary in description that excludes non-stop term1 of all the non-visited web sites and applies the index model [1] to create an index for each term. When the index of certain term exceeds the pre-determined cutoff threshold, the filtering mechanism will keep it into the list of filtering terms and stores the analyzed results into filtering base. When a user issues a search request with this filtering level, the Information Filtering Agent will utilize the filtering base to filter the query result and to discard those irrelevant results.

3. *Contents*: The same technology as second level is applied to this level with the exception of analyzed target is the full content of web site. Because the size of analysis is the largest in three levels, the overhead is also largest. All of filtering level is based on the implicit feedback mechanism that feed back the selected web sites implicitly. The detailed description is in next subsection.

Figure 4.10 shows the detail architecture in supporting the functionality of personalized search in Octopus. Based on ISS, some components are added into the system to support this function, such as *User Profile*, *Filtering Database*, *Feedback mechanism*, and *Result processing mechanism* etc. Follows describe the system scenario.

The system first checks user identifier through *User Profile*. Once the user passes the check, the system will generate a query page for user to post the query string and wait for query request. Then, *Result Filter* will look for the query result from *Result Cache*. If missing the expected information, then *Query Page Generator* will submit this request to ISS's mediator for searching new information and get the result through the *Result Aggregator*. The *Result Filter* thereat filters the results based on the information of personal profile and filtering database and passes the filtered results to user. Once the user receives the results, *Feedback Mechanism* is activated to monitor the user's feedback. When a user wants to visit the web page through *Result Display* page, the visiting process will physically link to server's CGI program that can log the visited history and redirect the visited URL to the web site. The recorded information just represents that the URLs are related. So that the situation to filter those irrelevant information are the user complete a review session after visit those related web pages and press the "Next Page" or "Back" bottom.

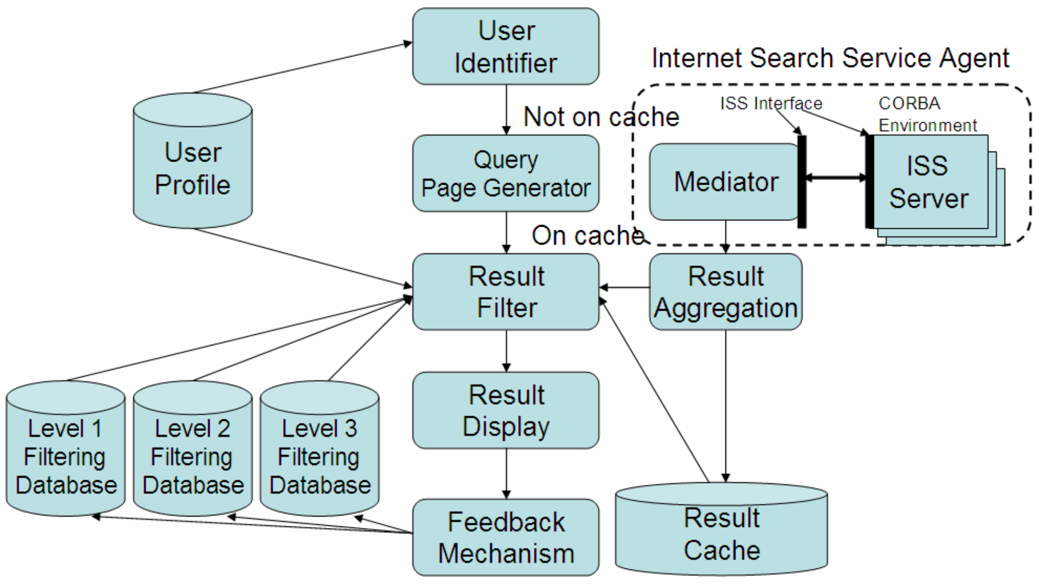


Figure 4.10: Personal information filtering agent.

In this approach, we have described the policy of supporting personalized search based on ISS. We also have implemented these functions into Octopus. In order to keep advantages of ISS, a personal information-filtering agent is added into the Octopus instead of modifying architecture or interface of ISS. In Octopus, an absolutely irrelevant feedback approach used to support the personalized search. Many advantages of using the ISS to build a multi-search engine have been raised in [19]. There are other advantages that are discovered in the design. First, a personal information-filtering agent is added into the Octopus instead of modifying architecture of interface of ISS when adding personal functionality. We believe this design is more suitable to exploit useful data in the other application. Second, because the interface of ISS is based on the distributed object-oriented technique and the modules of personalized search are implemented as replaceable components. It is ease to replace these components when a new and more suitable algorithm is proposed. Third, each user with specific domain has individual profile in Octopus. It might avoid the

Octopus return unsuitable results to users. Finally, because the filtering mechanism is divided into three levels, it can balance the load and the user requirement.

4.6 Automatic wrapper generator for integrated IR

The approach presents a design for an automatic XML-based framework with which to generate wrappers rapidly. Wrappers created with this framework support a unified interface for a meta-search information retrieval system based on the Internet Search Service using the CORBA standard. Greatly advantaged by the compatibility of CORBA and XML, a user can quickly and easily develop information-gathering applications, such as a meta-search engine or any other information source retrieval method. The two main things our design provides are a method of wrapper generation that is fast, simple, and efficient, and a wrapper generator that is CORBA and XML compliant and that supports a unified interface.

The effort has mainly gone into designing wrappers to translate the returns into a specific representation for queries from the mediator. In fact, for the retrieval application developer faced with multiple information sources, it is important for the available retrieval applications to have a uniform programming interface. It is for that we propose our integrated information retrieval methodology with a unified interface, as shown in [18]. The flexible architecture here has a unified programming interface and an information retrieval application for querying a variety of sources. We use an IIR (Integrated Information Retrieval) service based on COSS (Common Object Service Specification) of CORBA (Common Object Request Broker Architecture). The metadata of the sources is defined by DTD (Document Type Definition) of XML (eXtensible Markup Language). With this system, an information retrieval developer can easily design applications or agents to collect desired information via a high-level uniform programming interface.

The proposed architecture is ideal for the information retrieval task. However, because of the multiple sources, a supportive framework is necessary. In addition, the framework must ensure that information retrieval application developers can generate wrappers that are simple and fast and that are both XML and CORBA compliant. Being XML-compliant enables data exchange between different information sources, and being CORBA-compliant enables communication between heterogeneous systems. With practice, employing this framework in an SQL-like high-level query scheme, the user or the client program (e.g., an information retrieval application) can perform the extraction from a variety of sources.

Most of the current wrapper development methods have difficulty with designing query and extraction rules because a good knowledge of web documents and of the syntax of rules is required. Wrapper implementers find designing such rules difficult and tedious. In many systems on the Internet, the returned information is designed for user, not for a program. In addition, a wrapper is an important software component between the information retrieval system and the information source. A well-defined wrapper with a uniform communication interface improves the performance of a heterogeneous information retrieval system, but writing this kind of wrapper increases the workload of the wrapper programmer. The solution in this approach to that problem is an automatic generating framework for an XML-based wrapper with a CORBA-based unified interface. With this framework, the XML data model is used to express the metadata of information sources, and the output file of the results is also in XML format. CORBA is an open system model that supports communication between the software components within distributed environments and is used to define the uniform interface for the meta-search system proposed in [18][19]. With this framework, an information retrieval application can use the CORBA standard to communicate with a variety of wrappers and acquire the results based on a standardized object model. In addition, because XML is now a popular standard for representing and exchanging data, the wrapper programmer has no need to learn a new extraction language to generate wrappers.

A wrapper is a software component that embraces an information source. Its main objective is to be the interface between the client program and information source. Because of the heterogeneous information sources, it is best to support a wrapper with a uniform interface. A typical uniform interface wrapper has three tasks. First, it receives user requests from a mediator and then translates them into a query string format (typically into URL form for web information sources) or into a query command (typical in the RDBMS system) acceptable to the information source. Second, it retrieves Internet or the Intranet documents from which it extracts desired information according to extraction rules provided by the user. Last, it stores the desired information in a specific form and provides an interface to allow a user or client to retrieve data in a high-level and structured way.

The typical architecture of a wrapper supporting for [18] is shown in Figure 4.11. It has seven components: (1) Query String Translator, (2) Parameter Encapsulator, (3) Document Parser, (4) Information Extractor, (5) Result Packer, (6) Network Transmission Interface, and (7) Server Skeleton Interface. Each component is a stand-alone Java class, and can be developed and replaced independently. All the data passing through these components are created as an XML DOM objects. Consequently, the wrapper uses standard DOM API to develop the application. Because XML is a structured and meaningful data format, each component easily understands the content of received data and treats it appropriately in an explicit and precise way. Our automatic framework for wrapper generation uses the advantages of XML.

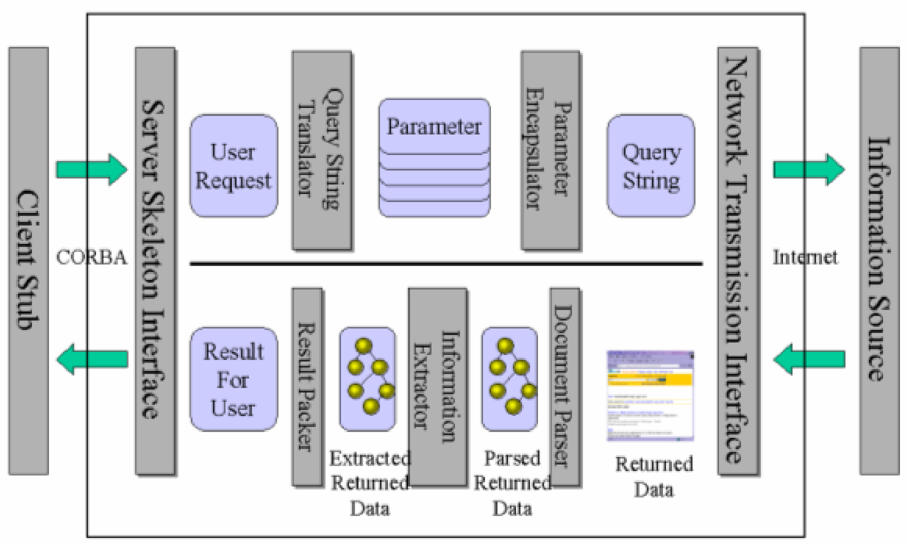


Figure 4.11: Basic architecture of XML-based wrapper.

We now discuss the architecture of the XML-based wrapper shown in Figure 4.11 in more detail. On the right of the figure is Network Transmission Interface, which is the gateway between the wrapper and the information source. Its task is to provide an interface such that the wrapper can send the appropriate query string to the individual information source and then retrieve the results through the interface. On the left is Server Skeleton Interface, which is the interface mapped to the stub of the client program based on CORBA standard. It follows that the interface defined by IDL is such that the wrapper receives through the Server Skeleton the SQL-like request (e.g., SELECT URL FORM ALTAVISTA WHERE KEYWORD=wrapper) from the Client Stub of the client application. The Query String Translator has the task of parsing the user request, extracting such parameters as URL, ALTAVISTA, KEYWORD=wrapper and packing them into a DOM object. It then passes the object to Parameter Encapsulator, the main task of which is to encapsulate the parameters into an appropriate query string and send that to a specified search engine or information source. For example, the base query string for AltaVista (http://www.altavista.com) is in the form of http://www.altavista.com/cgi/query?q=DT…, and all the parameters (e.g., KEYWORD=wrapper) are appended to that string in the appropriate way. The query string with parameters is sent by Network Transmission Interface to the specified search engine (e.g., AltaVista). Next, Network Transmission Interface retrieves the results and passes the data to Document Parser.

Document Parser then parses the content into an XML parse tree. Information Extractor extracts the desired fields according to user-defined extraction rules. Obviously, four major fields (URL, TITLE, DESCRIPTION, and RELATED PAGE) are given in the returned AltaVista document, and Information Extractor consequently extracts from it any URL that contains the keyword, 'wrapper'. Finally, Results Packer packs into standard format all the information needed by the user into the Client Stub that is generated by CORBA IDL compiler.

To support the uniform interface of the integrating information retrieval environment proposed in [18], the generated wrappers have to be reconciled with IIR (Integrated Information Retrieval) interfaces. There are five IIR interfaces: InformationRetriever, Wrapper, MetaData, Collector, and Iterator. The first of these is explained as follows:

```
Interface InformationRetriever
{
   MetaData Get_meta(in QuerySourceType qsType);
   Wrapper prepare(in ParameterList pl,   in QuerySourceName qsName,
                                                   in QueryLanguageType qlType);
}
```

The interface of InformationRetriever has two methods, Get_meta() and prepare(). The purpose of Get_meta() is to obtain a object reference of MetaData object. The purpose of prepare() is to prepare a query request with appropriate parameters for a specified information source and then to obtain a object reference of Wrapper object to start the query process. The Wrapper interface is explained as follows:

```
Interface Wrapper
{
   Collector Query() raises(QueryProcessingError, QueryInvalid);
}
```

The interface of Wrapper allows a mediator to start the retrieval process by invoking the Query() method of the Wrapper interface. InformationRetriever then dispatches an appropriate wrapper to handle a specified information source according to the information obtained from Get_meta(). The interface of MetaData is explained as follows:

```
Interface MetaData
{
        Boolean QL_Available(in QuerySourceType qsType);
        Boolean Registry(in QuerySourceMeta metadata);
        Boolean Unregistry(in QuerySourceName qsName);
        Boolean Replace(in QuerySourceName qsName, in QuerySourceName metadata);
        QuerySourceMeta get(in QueryLanguageType ql_type);
}
```

The interface of MetaData provides data retrieval robustness while retrieving any information source. By the appropriate assignment of each field of MetaData, the client can obtain the format of a query request and the schema of the result. For the detail of MetaData, please refer to [15].

The last two interfaces, Collector and Iterator, are responsible for collecting information from sources. They are explained as follows:

```
Interface Collector
{
     MetaData GetMeta(in QueryLanguageType qsType);
        Readonly attribute long Result_size;
        Result retrieve_element_at(in long where);
        Iterator create_iterator();
}

Interface Iterator
{
     Result next();
        Boolean reset();
        Boolean more();
}
```

As may now be seen, a client program has two methods of retrieving desired data. The first is random retrieval, which uses Result_size() and retrieve_element_at(). The second is sequential retrieval, using create_iterator(). This last creates an Iterator object, which provides a simple interface for data retrieval. For detailed information about the unified interface integrating information environment, please refer to section 4.4.

To generate a wrapper for a specified search engine or information source correctly, the wrapper implementer must provide adequate information to the wrapper generator in the following forms: *Query Template* file, *Pattern* file, and *Scheme* file. The role of each in our wrapper generation framework is shown in Figure 4.12. Within the framework, wrapper generation has three phases: (1) *Query Translation*, (2) *Documents Retrieval*, and (3) *Results Translation*.

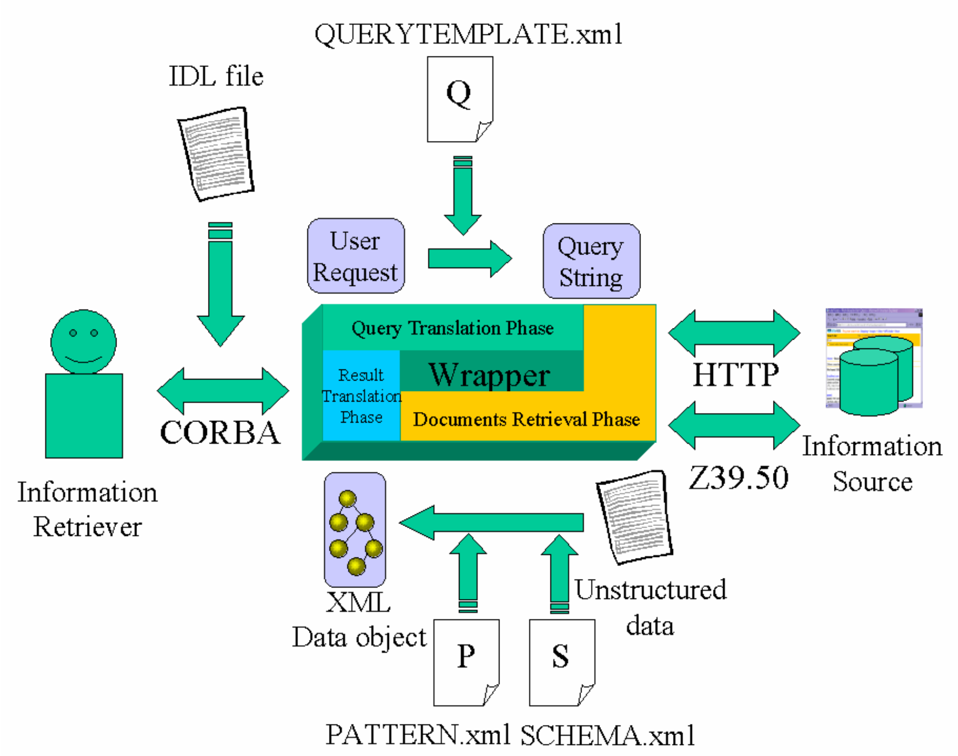


Figure 4.12: An overview of the wrapper generator.

In Figure 4.12, we consider the developed information retrieval application. The application communicates to the wrapper according to the interface defined by IDL files. In the *Query Translation* phase, the wrapper translates the user's high-level SQL-like queries from the application via the CORBA standard into an acceptable format for the specified search engine or information source, for both of which the wrapper generator must also know the query command patterns. The wrapper implementer in the QUERYTEMPLATE file must provide the query format.

In the *Document Retrieval* phase, using the QUERYTEMPLATE file, the wrapper attempts document retrieval from a specified information source by the appropriate query string or command. If the information provided by user is correct and the network connection is good enough, the desired documents is quickly retrieved into local data storage. Most returned documents from information sources are either semi-structured or unstructured. One of the responsibilities of the wrapper generation in this phase is to parse documents, extract desired information, and then store it into XML DOM objects. The wrapper implementer must provide in advance a PATTERN file that regulates the handling of the specified data fields for the document parsing process. In addition, for proper management of the data in the specified fields, detailed information about the data type and variable name of the user-interested fields has to be provided in the SCHEMA part of the PATTERN file. The QUERYTEMPLATE, PATTERN, and SCHEMA files are all written in XML syntax.

Following the *Document Retrieval* phase, the user-interested information is stored in the data structure of a wrapper in the form of XML DOM objects. In the *Result Translation* phase, obtaining the information from these objects via the CORBA standard is fast and simple for the client program. One of the responsibilities of the wrapper in this phase is to prepare the results extracted from DOM objects in appropriate format and send them back to the client program via CORBA.

The interface between the wrapper and the client program is also an important part of the framework. A concise and standard interface is needed here, so we adopt IDL (Interface Definition Language) of the CORBA standard to define it. The CORBA standard is language-independent, so that the wrapper generated within the framework can be communicated to the client program whatever the language or operating system. That is, a wrapper generated within the framework is an XML-based and CORBA-enabled component over the network. In this way, wrappers fully support the integrated heterogeneous information retrieval system we propose in [18].

As discussed earlier, most information sources constantly change content or even structure. Wrapper codes are tightly coupled with the structure of a specified information source. If the information source changes the document structure, the wrapper implementer must also change the wrapper codes. Such constant modification is both time-consuming and tedious. Consequently, an information retrieval application developer welcomes an automatic wrapper generation system that decreases the workload. Unlike what is proposed in other works, the XML-based wrapper generation framework we present is automatic and consequently answers the need. Not only is it XML-compliant and CORBA-enabled, but the generating procedure is simple and fast. The workflow for the framework is shown in Figure 4.13.

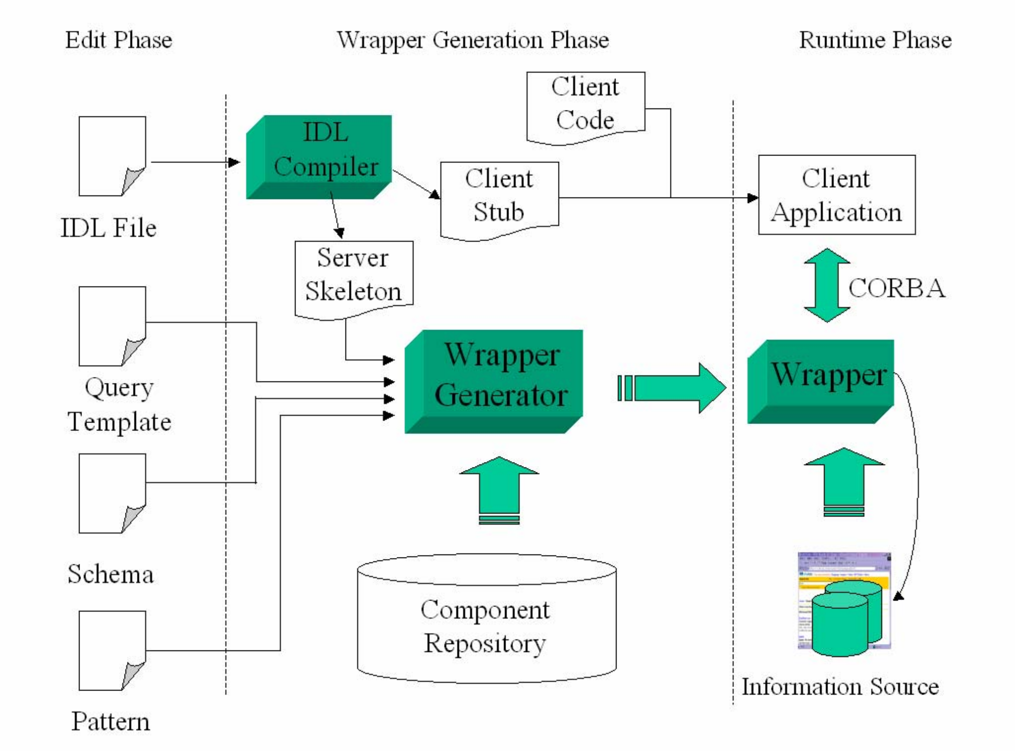


Figure 4.13: The workflow of the proposed wrapper generation.

There are many editing tools for XML and IDL files. The wrapper implementer must select the appropriate tools to define and prepare the QUERY TEMPLATE, PATTERN, SCHEMA, and IDL files shown in Figure 4.13. Consider the *Wrapper Generation* phase shown in this figure. First, an IDL compiler is employed to compile a user-defined IDL file and produce a client stub and server skeleton for the interface between the client application and the CORBA-based generated wrapper. The information retriever uses the client stub to develop a client program that communicates to the wrapper in an appropriate way. The server skeleton together with the QUERY TEMPLATE, PATTERN, and SCHEMA files are required in the wrapper generation procedure. Employing these user-defined files, the generator selects the appropriate components stored in the *Component Repository* to produce the desired wrapper. At *Runtime Phase*, the new wrapper can be invoked by to retrieve the requested in the client program via the CORBA interface defined in the IDL file formation.

The architecture of our wrapper generator is shown in Figure 4.14. Three XML-formed files, QUERY TEMPLATE, SCHEMA, and PATTERN, are required. An XML parser parses them and stores the desired information into DOM objects defined by W3C. That is, following XML Parsing Phase, three DOM objects are produced. Then, *Query Rules Analysis* phase creates the *Query Transfer Rules* and the *Query String* according to the DOM objects coming from the QUERY TEMPLATE file. The *Data Extract Rules* are generated at *Data Extract Rule Analysis* phase according to the DOM objects generated from the SCHEMA file. The *Pattern Sample* is produced by *Pattern Analysis* phase according to the DOM object generated from PATTERN file. The *Code Generator* shown in Figure 4.13 then chooses suitable components from the Component Repository in accordance with all these information (Query Transfer Rules, Query String, Data Extract Rules, and Pattern Sample). Finally, Binder binds Code Generator-chosen components and Server Skeleton code fragments produced by the IDL compiler into the desired new wrapper.

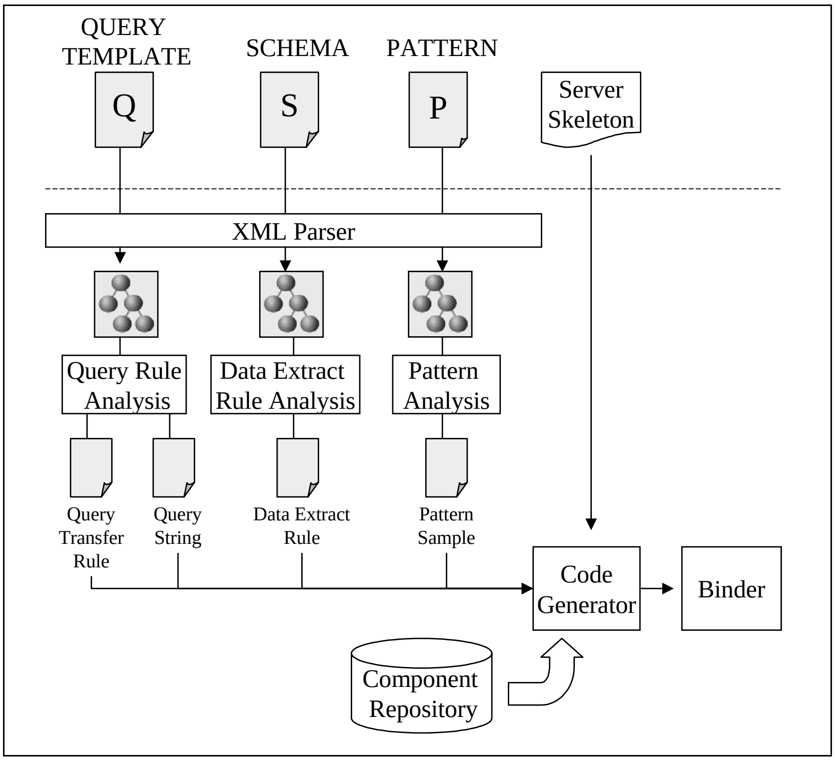


Figure 4.14: Architecture of automatic wrapper generator.

In summary, the usual method for extracting information fields of interest in the framework is by pattern matching. Using it, a wrapper implementer quickly and easily prepares wrapper specifications for a specified information source. This differs from previous approaches in that a wrapper generated in our framework focuses on extracting fields of interest from returned documents and not on analyzing their content. An implementer is not required to understand the whole structure of a specified information source. Consequently, time and cost in generating new wrapper are greatly reduced.

All the data structures, including imported files and the representation of objects are XML compliant. Since XML is a widely popular standard nowadays, most developers are familiar with it and many software applications process data with it perfectly well. Combining the advantages of both XML and Java, the *Component Repository* used in the framework is flexible and extensible. The codes of the *Component Repository* are easy to be extended and managed. Finally, wrappers generated in this framework are all CORBA-enabled. That is, for information retrieval application, they are already language independent, which merit adds to their communicative capabilities and spares the wrapper implementer much effort.

Again, for our proposed e-MARE framework, the wrapper generating technology proposed in this section provides the system developer a tool to construct a multi-search e-MARE service as more as possible in an easy and convenient way.

4.7 Overview of IBM Aglets

Aglet is a java-based internet agent approach. It is a lightweight mobile agent technology from IBM's Tokyo Research Laboratory early 1995. With ASDK (Aglet Software Development Kit), the user can write a pure java, lightweight object migration, and event-driven mobile agent named aglet to do something across hosts on the Internet. They provide a set of Java classes and interfaces called Aglet APIs that allows user to create mobile Java agents. In the framework of Aglet, the mobile agent can be halted suddenly and be dispatched into another host with its state, then, it can be resume execution immediately from the interrupt point. In other words, aglet can make an itinerary by traveling across the Internet Aglet Server to do anything he wants.

In the definition of Aglets [49], an ***aglet*** is a mobile Java object that visits aglet-enabled hosts in computer network. It is autonomous, since it runs in its own thread of execution after arriving at a host, and reactive, because of its ability to respond to incoming messages.

A ***context*** is an aglet’s workplace. It is a stationary object that provides a means for maintaining and managing running aglets in a uniform execution environment where the host systm is secured against malicious aglets. One node in a computer network may host multiple contexts.

A ***proxy*** is a representative of an aglet. It serves as a shield for the aglet that protects the aglet from direct access to its public methods. The proxy also provides location transparency for the aglet. That is, it can hide the real location of the aglet.

A ***message*** is an object exchanged between aglets. It allows for synchronous as well as asynchronous message-passing between aglets. Message-passing can be used by aglets to collaborate and exchange information in a loosely coupled fashion.

A ***message manager*** allows for concurrency control of incoming messages.

An ***itinerary*** is an aglet's travel plan. It provides a convenient abstraction for non-trivial travel patterns and routing.

An ***identifier*** is bound to each aglet. This identifier is globally unique and immutable through the lifetime of the aglet.

Many behaviors supported in the aglet object model include ***creation, cloning, dispatching, retraction, deactivation, activation, disposal of,*** and ***messaging***. Most of these behaviors can be managed by Aglet Viewer. The architecture of Aglet Viewer is shown in Figure 4.15. [41]

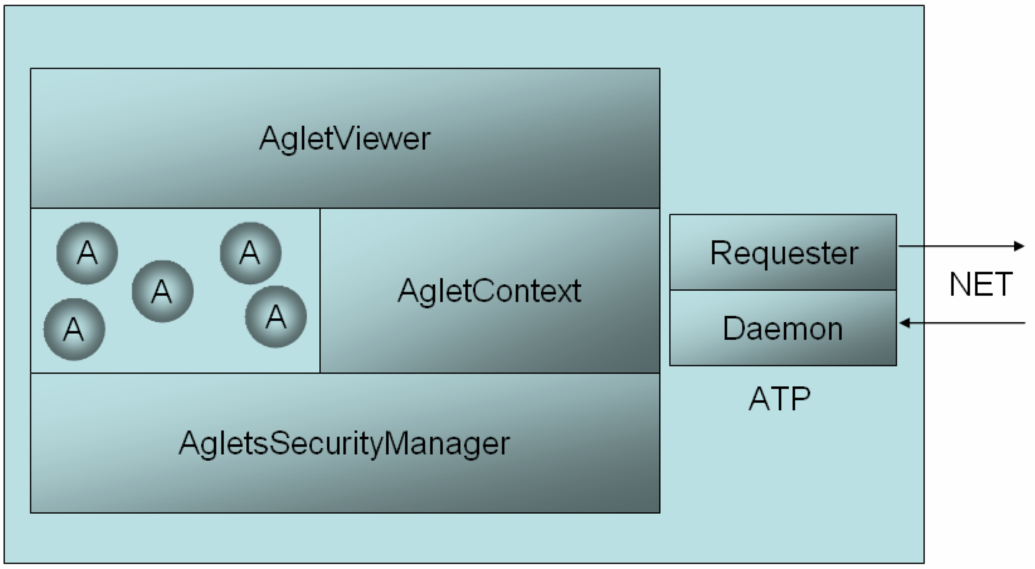


Figure 4.15: The architecture of Aglet Viewer.

As Figure 4.15 shows, an aglet viewer such as Tahiti server contains many aglets. The aglet viewer will access and accept the remote requests through the network. Actually, the aglet viewer is in sense a program running controller. It allows us to create, retract, activate, deactivate, and dispatch aglets. The aglets are migrated across hosts by the ATP protocol. Every hosts have to be installed an ATP server, and it will has the ability to accept the aglets. The Aglets Infrastructure is shown in Figure 4.16. [41]

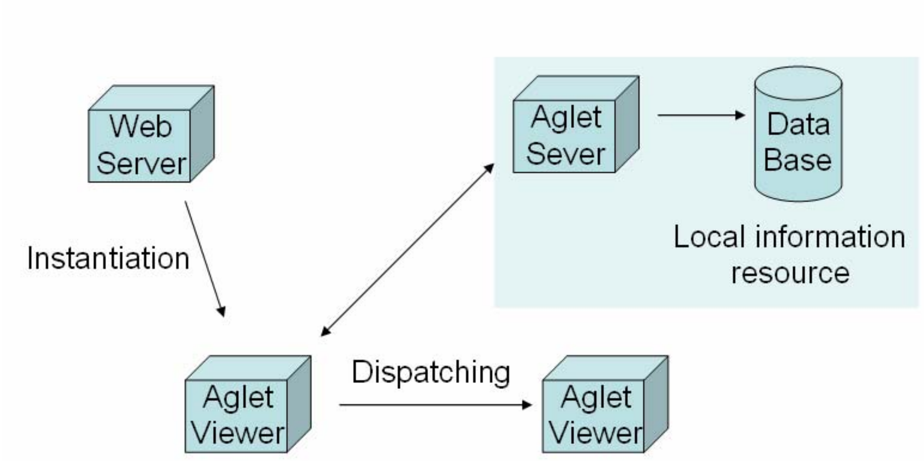


Figure 4.16: The infrastructure of Aglets.

As Figure 4.16 shows, there are several aglets from Aglet Severs are running at distinct servers. They are managed by themselves or by the users through Aglet Viewers. Each aglet has the ability to retrieval and to analyze the information from the Internet. They also have the ability to access the resource on the local hosts and the ability to communicate each other as well. The aglets as like as the traditional programs, they can do everything that the user wants they do.

The framework of e-MARE may easily integrate the Aglets host through mobile runtime server by SMTP, POP3, FTP, TELNET, HTTP, CGI protocol.

ES 1896

Chapter 5

System Design and Implementation

This chapter overviews the prototype system of e-MARE, and describes each design and implementation issues in detail. Basically, the prototype system is composed by several servers. These servers include Linux and Microsoft Windows 2000 server operating system. They are connected by high speed Ethernet connection, and cooperating each other. At the last section of the chapter, the complete instruction set which provided by e-MARE server will also be described in detail.

5.1 System overview

One of the most important objectives of our research in this dissertation is to propose a framework for mobile user who wants to design, operate, and execute mobile agents by mobile device (mobile phones, PDA, desktop PC, or even workstations and servers). Moreover, the framework should provide a unified interface for the user to access the Internet by any distinct mobile devices. This means the framework should provide the **user mobility** for the mobile users. User mobility allows a user to move and to use one or more terminals in order to retrieval information from the Internet. Terminals can be of different types having different capabilities, varying from the mobile device with very limited computing power, memory and display capabilities to a powerful multimedia desktop PC. It should be possible for a user to use several terminals at the same time. Conversely, several users can share the same terminal. The characteristic of user mobility is especially useful for the mobile device users. It is an easy way for these users to connect to the Internet and fetch the useful information from it.

In our design, the role of mobile devices is mainly to be treated as an interface between the users and the servers. The mobile agents are all executed at the server. We assume the back-end servers of our proposed framework is a desktop computer with high computing power and a well network connection, thus, the agents can achieve most users' desire, includes information retrieval, timetabling, price haggling, online auction, etc. Of course, all the advanced technologies of information retrieval can be applied into such powerful computer.

To achieve real cross platform mechanism, we propose e-MARE (Email-based Mobile Agent Runtime Environment) framework in this dissertation. According to the previous researches, we found that the e-mail protocol is the most popular application for all the networked devices. Almost all the networked devices provide the e-mail client to access the e-mail messages from the Internet. Thus, based on the e-mail protocol, the users can easily manage, design, and dispatch the mobile agent on the remote server. The basic idea of infrastructure of our proposed framework is shown in Figure 5.1.

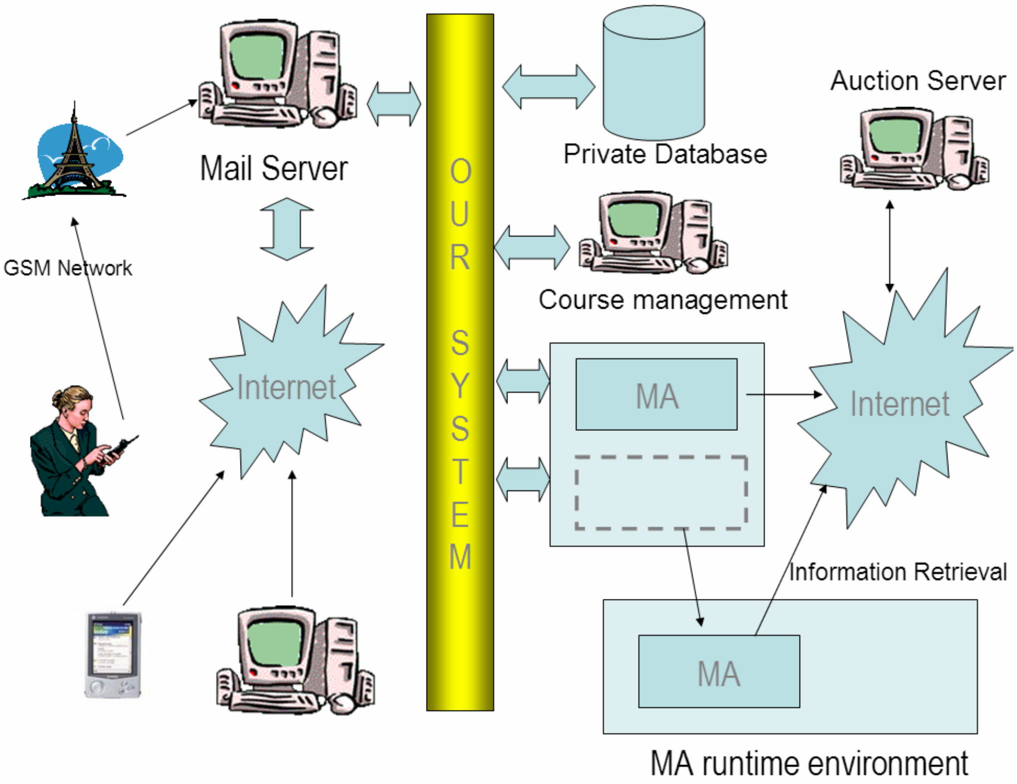


Figure 5.1: The basic infrastructure of our proposed e-MARE framework.

At the first, the *Coordinator Server* of our proposed framework is waiting the messages from the mail server. In most cases, the *Coordinator Server* of our system is also equipped a standard e-mail service. After the e-mail with specified commands (we call such specific e-mail '***operational e-mail***') is received from the specified mail server, the system will parse and analyze this operational e-mail. It will extract the specific commands or even mobile agents from the e-mail, and prepare them to be executed on the remote server. If the extracted commands are simple commands, they are always executed in the *Coordinator Server*, and the *Coordinator Server* will actively collect the execution results into an e-mail, and send it back to the requester's mailbox.

In some cases, the extracted commands may be the commands for controlling the remote mobile agents. In such cases, the system will redirect these commands to appropriate agent which is response for management of those mobile agent runtime environments. As mentioned above, the mobile user can operate the remote mobile agent via mobile devices by those management commands.

On the other hand, if the extracted command is a complete program codes, it means the user have programmed his own application or mobile agent for executing at the remote server. In such case, the framework will pack these program codes into a distinct file, and send it to the remote execution server of the framework via *File Transfer Protocol* (FTP) protocol. After the file has sent to the remote execution server, the system will start the execution of that program, and collect the executed results via both FTP and TELNET protocol. If the executed program is a mobile agent, it should has the ability to autonomously migrate themselves to another hosts by themselves or by the users. To extend the power for the execution of mobile agents, we may add more than one mobile agent running servers into the framework. These servers are also the computer with high computing power and well networking connection.

Through our system mentioned above, the possible applications can be referred in Figure 5.1. As shown in Figure 5.1, a mobile user may use mobile phone, PDA, or desktop computer to connect to his prefer e-mail server by the e-mail client application of those various terminals. Utilizing the instruction set, the user may directly query the desired information of his private database on this private network, or the public web documents on the Internet. Moreover, by encoding some simple commands, the mobile user also may directly connect to some information providers or information sources, i.e. *Course Management*, or *Online Stock Quote Web Site*, which are supporting HTTP/CGI protocol on the Internet. Applying both two mentioned techniques, the mobile user also can write a information retrieval application to directly execute on the remote server to fulfill the user's information retrieval purpose.

Because the system also allows the user to attach his programs in the operational e-mail, thus, the user may also write some programs which follow some sort of mobile agent system, and then sent them to the remote mobile agent runtime environment for execution. Through this mechanism, the mobile agents can do every information retrieval works. Of course, these mobile agents can contact other information system to achieve any user's purpose via the

Internet and computer network.

Although our framework provides the ability for a user to write some commands or programs on their mobile device directly, the characteristics of most mobile devices are not suited for input lots of text by their limited input interface. It is not convenient for user to write a large program for the complicate works of the user. In most situations, the user may write and store his agent program on the desktop PC in advance. These user-defined agent programs will be stored on the server, and can be operated by the user through his mobile device to achieve his information retrieval project. Thus, the system has to provide a mobile agent runtime management system on the server for the execution of the user-defined mobile agents. In our framework, we utilize a pure Java based mobile agent runtime environment, IBM Aglet, to be the platform for user's agents to be executed. The framework can connect more than one Aglet server for maximize the utilization and flexibility of the system. We also provide a gateway to be the interface between the mobile device's operational e-mail and the Aglet server.

5.2 System workflow

In the design of our approach, after the system has received the operational e-mail from the user, the itinerary of the user's request will be started right away. If the user is first time to request the service of the system, he now has to make a registration to the system. The main purpose of registration is for both the security and user profiling concerns. All the users have to register to the system, this will prevent un-authorized person to access the system. The further advantage is that every registered user will has his or her account/password pair. In general, this pair is automatically provided by the system after the user has been completed his registration process. The pair is the major identification for using the system's services. Moreover, the pair is not only for identification of the system, but also for the *User Profiler* to record the user's personal preferences for further access. Then *User Profiler* will fine-tune the display preference of results according to the user's history accessing profile.

In our framework, if a user wants to access the information on the Internet via his mobile devices, firstly, he may write an operational e-mail with specified encoded subject by the mobile devices, providing his account, password, and his desired operation for mobile agent on the remote server. After he has finished the operational e-mail, he has to send the e-mail to his own target e-mail address specified by the system while he is registering. Typically, the system's service e-mail account is identical. From the viewpoint of the user's request, the basic system workflow of e-MARE is shown in Figure 5.2.

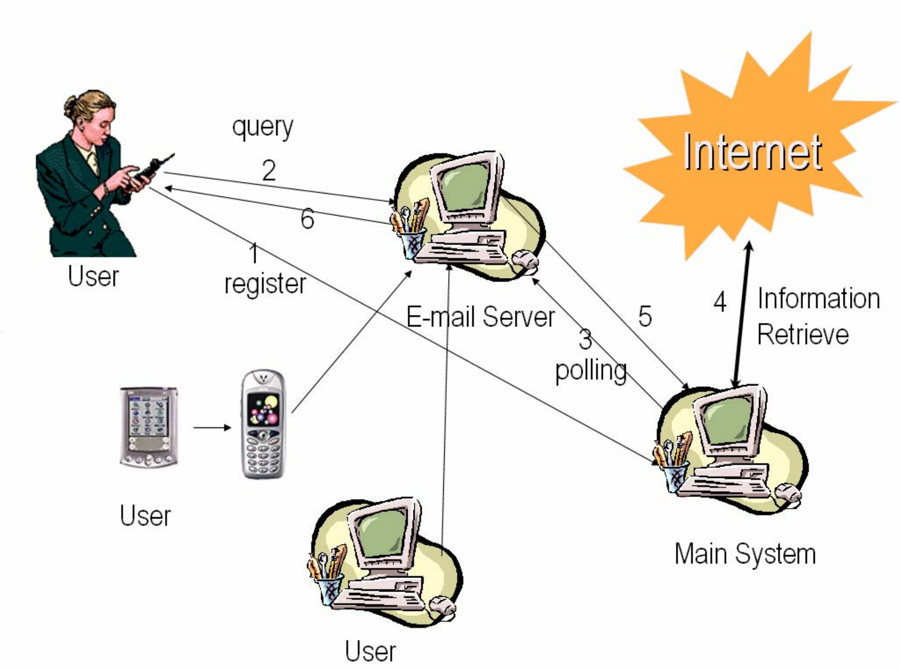


Figure 5.2: Basic system workflow of e-MARE.

As Figure 5.2 shows, after the first registration step, the user may send an operational e-mail to the service e-mail account by her favorite e-mail server in second step. In this case, the main system will poll the specific registered e-mail account to filter the operational e-mail of the user from the e-mail server, and this is step 3. When the system has received the operational e-mail from the user, it will first extract the specified commands from the e-mail. The commands may be a simple command sheet or a complex mobile agent program. No matter what the kind of the command is, the system will dispatch a stationary agent to response for the execution of the user's commands. The stationary agent will monitor the commands or the mobile agent's execution, and collect the results into an e-mail. In this case, we assume the user's purpose is to retrieve the information of the Internet. Thus, this is the 4th step which is shown in Figure 5.2. The 5th step of the system is to pack the results to an e-mail when all the related works have been done. After the 5th step has been completed, the results will be already in the user's mailbox. The final step, 6th step, is the user to execute her e-mail client application on the mobile device for fetching her e-mail, the results returned by her request in the last operational e-mail.

Of course, the first consideration of the workflow of this framework is the networking ability and usage of the mobile devices. That is, firstly, for the mobile user, the networking connection will be needed only at the time that e-mail was being sending and receiving. The networking connection is needless when the user is designing the mobile agent or reading the results. This mechanism will greatly reduce the cost of networking connection for mobile users. Secondly, the e-mail application is the most popular application for networked device. It is an easy and fast implementation for any terminals. Moreover, since the user interface of e-mail application is simple and similar, the e-mail based framework is very easy to practice. Finally, the operational e-mail can treated as the normal e-mail. It can be read, replied, or even forwarded to other's mailbox. Then, the system can achieve the characteristics of reusability, cross-platform, and simplicity. Of course, all the works can be done at desktop PC or notebook, and their user interfaces and programming formats are the same.

5.3 System architecture

To achieve the functions and abilities which are described in the section 5.1 and 5.2, we design the system architecture of e-MARE framework, which is shown in Figure 5.3.

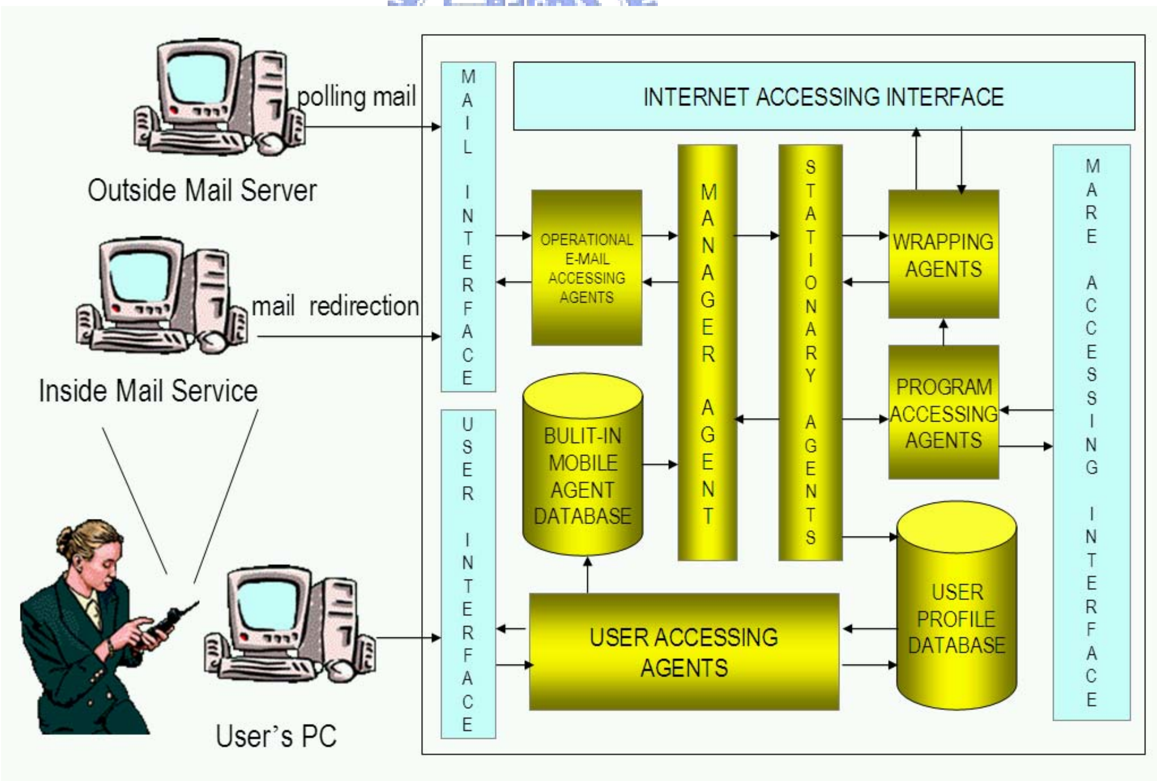


Figure 5.3: The basic system architecture of e-MARE.

Figure 5.3 shows the basic system architecture of e-MARE framework. For the convenience of the user, we admit the user to choose either creating a new e-mail account (for example emare@mail.twvs.tnc.edu.tw) called *Inside Mail Service Model* (short for IMSM) or using old one as target e-MARE service account called *Outside Mail Service Model* (short for OMSM). In the IMSM, we apply a mail redirection mechanism for the server to trigger the service while the specified operational e-mail is incoming. The new created e-mail account is the main interface between the mobile user and e-MARE. In general cases, the user also can choose the public e-mail account as the target service account. All requests will be sent to that e-mail address via operational e-mail. In the OMSM, the user has to provide his e-mail address and password for accessing his mailbox. And then the system will activate an event monitor process to regularly poll and check the user's e-mail in that external mail server. Either IMSM or OMSM, the operational e-mail will be accessed by *Operational E-mail Accessing Agents* through *Mail Interface*.

Through the message flow as the Figure 5.3 shown, behind the *Operational E-mail Accessing Agents* is the server's coordinator called *ManagerAgent*. It is responsible for determining the type of the operational e-mail, and dispatching these requests to appropriate *Stationary Agents*. From the system's viewpoint, the operational e-mail can be divided into two major categories: they are for accessing the Internet and for accessing some traditional mobile agent runtime environment systems respectively. If the operational e-mail is the former, the activated stationary agent will invoke the some specified *Wrapping Agents* to prepare the appropriate querying commands for specified Internet services.

Conversely, if the type of operational e-mail is the latter one, the system will, in most cases, need to extract the program which is embedded in operational e-mail. Then the extracted program will be dispatched to the remote host in those environments by *Program Accessing Agents*. In some cases, one of *Program Accessing Agents* also has to compile or prepare the extracted program. All of these works can be completed in an elaborate design under Linux environment by FTP and Telnet protocol. Eventually, the *Program Accessing Agents* will connect to the specified mobile agent runtime environment, and will dispatch the user-defined mobile agent or application to remote host through the *Mobile Agent Runtime Environment Accessing Interface*. Finally, all of the results have to be collected by *Program Accessing Agents* and be sent to one of *Wrapping Agent* which is responsible for returning these results to the user via e-mail. To ease the reading for mobile user with limited screen size, we also apply the *Result-Slicing* mechanism in *Wrapping Agents* to handle the results.

The *Result-Slicing* mechanism will be described in latter section.

As mentioned in Section 5.2, the first step to request the service of e-MARE is to register the related information by *User Accessing Agents* through the *User Interface*. In this step, the user has to provide her personal information, such as account, password, preferred mobile devices, habits, arguments, and other preferences. The *User Accessing Agents* will record the information into *User Profile Database*. This will help the system to improve the precise of the results which is prepared for the user. Furthermore, the user also can utilize the *User Accessing Agents* to add/modify/remove the *Built-In Mobile Agent Database* for reducing the future information retrieving works. All these works can be completed through the Web-based interface, or even through the operational e-mail.

The major advantage of utilizing *User Profile Database* to record the user's information and the user's requesting history is to improve the precise for personal information analysis. That is, the framework of e-MARE for information retrieval is a kind of off-line information retrieval system. Thus, when the user wants to gather and to analyze the information more detail, the system have more time to prepare the information. Moreover, the system can analyze the collected information according to the personal information from the User *Profile Database*. Thus, the results are not only providing complete information for the user's requests, but also being tuned for the specified person. That is, although the information fetched from the Internet by mobile agents is same, the display order of the information is different for each distinct user. The mechanism could be achieved that is due to the off-line characteristic of the e-mail protocol. The off-line informational retrieval architecture with personal profiler can provide the in depth analysis for the user's requests.

For example, when a user wants to write an advanced query command in some kind of integrated meta-search service [45] as follows:

SELECET URL, TITLE, DESCRIPTION FROM YAHOO WHERE KEYWORD=’SQL’

If the user is interested in the information about SQL Server, in general, it usually means that the user may more prefer the information about SQL Server on the Internet this time. So the collected information from the Internet about the keyword 'SQL' will be arranged by the priority according to the user's history profile. As the previous example explained, the information about Microsoft SQL Server will be listed before than the information about SQL language. Again, to improve the efficient and precise of the user profile mechanism, the relevance feedback mechanism [25][50] is also applied in our framework.

To design and operate the mobile agent of remote host, through the mobile device is not the best way. Actually, we suggest the user to design, operate, upload, and setup the mobile agents by desktop PC through the WWW interface in advance. The WWW interface also provides the function for the user to assign specified mobile agent to a specified ID. Thus, the user may simply use the ID to operate the mobile agent on the mobile device. This mechanism will greatly reduce the user's effort on the mobile devices. That is, through the mechanism of user-defined mobile agents and applications, the e-MARE can offer the built-in mobile agents and programs in *Built-In Mobile Agent Database*. The user also may install and edit her own mobile agents or programs by *User Accessing Agents* through the *User Interface*.

5.4 Design Issues

To achieve the objectives of our framework according to the present mobile network environments, there are some challenges have to be conquered. They are shown as the follows:

Security and priority

To be a good system on the network, the security is a very important issue. However security issues are the most important part of a mobile agent system, they are mostly outside the scope of our approach due to limited time and resources. In this dissertation, we only focus to explore all the possibilities of mobile devices for accessing the information outside the GSM network, the related security issues are considered with less effort.

Nevertheless, to keep the basic security of our proposed framework, not only each user has to be authenticated, the service of the system but also has to be protected in different priority with different user. Our system offers the following solutions for security issue:

- Each user should be authenticated by distinct account / password pair.
- The system allows different user to have different priority for accessing the system's services.
- To those limited users, they will not access the local resource (such as disk, printer, and monitor) which is protected by the system.
- Any incoming operational e-mail has to be verified by the *Coordinator Agent* of the system. This mechanism will prevent the malicious codes

from the ill-disposed user.

Devices diversity and heterogeneousness

Different to the desktop computers, the mobile devices are very different from display ability to operation system. If the user's device is powerful and equipped a large and colorful display, the user of course desires to get more fantastic contents form the Internet, but if the user's device is a mobile device (such as mobile phone and PDA) with limited computing power, limited display size, and expensive and unstable networking connection, the concise contents will be prefer than the fantastic one. Plus, if the information is too long to display on the mobile device at once, the system should have the ability to aware the situation, and split these information into several pages automatically. After the user has read the first page of the returned results, he can evaluate the usability of the returned results according to the contents and quality of first page. If the usability of the results is below some threshold, he can stop the further transmission of other pages. It will save both time and cost of networking connection. Moreover, the users also may decide which page of the remains should be displayed next, thus the user's browsing flexibility will be enhanced as much as possible. This mechanism is called result-slicing, which will be explained in latter section.

Consider to these related issues, our system provides the following mechanisms:

- If the user has not specified the target devices at the client side, the contents for returned results are encoded in only ASCII by default.
- It should allow users to specify the distinctive arguments of his device. The e-MARE system will find the best way for presentation of the returned results according to the user's request.
- All the built-in functions and mobile agents are encoded into some specified distinct identifications (IDs). The user can specify the operations and commands by ID instead of command name. It will simplify the input of the mobile devices.
- The return results e-mail will also be encoded in some specified distinct identification number. The user can also specify the operations and commands with the identification for further operating to the specified return results e-mail. This skill is not only reducing the efforts for user to input the arguments of command, but also enhancing the interaction

between the user and the information sources or providers.

Inconvenience for characters input

To input too many characters by mobile phones is a very time-consuming and boring work. Composing the operational e-mail with the limited keyboard of mobile device will waste too much time. To conquer the input obstacle issue, we apply the following mechanisms:

- Simplify the commands. Most commands of our design are composed of only few alphabets. Moreover, many of those commands can be instead of identifications.
- The list of built-in functions and built-in mobile agents can be downloaded into the user's mailbox. Each function or mobile agent will be encoded as an ID number. Thus, the user may simply reply the e-mail and assign the ID number to fulfill the user's requests.
- The body of e-mail is the best place to be the interface between the mobile user and server. The user may issue a command for listing all the services of the server in the operation e-mail. Then the server will reply a list for all services of the server with a distinctive identification number to the user. Each service of that list will be combined a number for identification. Thus the user may specify both the e-mail identification serial number and the service number; therefore the server will activate specified service of the server only by the numbers.
- To further simplify the input of character about Chinese and English, we also encoded all the Chinese and English characters into the combination of numbers. That is, in the ideal case, all the Chinese and English text can be enterer by numbers. For mobile phones, input numbers is the easiest way for mobile users. It will dramatically reduce the efforts for input.

Preloaded and device capacity

Although our system allows the user to directly write the programs via the mobile devices by composing the operational e-mail, to write the programs via the mobile devices is a very time-consuming work. To avoid the boring works, our system also provides the web-based interface for user to upload his programs. By the interface of web browser connected to our system, the user can write the programs, and install them into the system as the built-in service of the server. When the setup operation is finished, those programs will act as the built-in mobile agents or applications. Those programs can be activated and operated through the mobile devices. The mechanisms of our system related these issues are shown as the follows:

- The system provides a web-based interface on the server. The user may write the mobile agent programs through the interface. Those mobile agents act as the built-in functions and mobile agents for the mobile user to access.
- To the mobile phones users, the system also provides a WAP-based interface. It is the same as the web-based interface except the appearance. The user may program, operate the mobile agents through the WAP-based interface. However, due to the display ability limitation of mobile phones, using WAP-based interface to directly design the mobile agent or application is not recommended, but it is also a feasible way for the mobile user to remotely manage their applications anywhere at anytime.
- The services of server can be an external program, an internal system service and function, or a user-defined scripts or programs. That is, because our mobile agent execution server is a UNIX-based server, all the built-in commands and shell scripts of that server will be treated as the callable applications for the mobile users. Furthermore, the user also can write his own program, and upload it to the server. The uploaded programs are also treated as the callable programs. Of course, this skill is convenient but with some potential security problems.

The interaction between mobile devices and remote server

Using the e-mail to be the medium for information retrieval on the Internet, the major shortage is the lost of real-time interaction between the mobile user and the information provider or information retrieval systems. To avoid the problem and reduce the bad effects, e-MARE utilizes the subject of e-mail as the main command list by default. The role of the body of e-mail has two possibilities: 1) to be the data which will be accessed during the information retrieval process while the user is initiating the operational e-mail; 2) to be the results of the reply e-mail initiated by the server corresponded to the requests of the user. Parsing the subject from the e-mail is very fast, the server can reply an e-mail for acknowledgment to the user in a very short period. We summarize the issues as follows:

- When the user wants to issue some operating commands to server, the operating commands have to be encoded in the subject of the operational e-mail. To avoid the server process the wrong e-mail, all the subject of operating e-mail have to be started by '&&'.
- If there is the data has to be processed altogether, the data have to be embedded into the body of the operational e-mail. The system will scan the body of the e-mail to extract the required data, and tie it to the commands or mobile agents.
- When the user's requests have been fulfilled, all the information will be prepared on an e-mail. Except the requested information, the e-mail also encodes a specific identification serial number at the subject, and attaches this message ID to the results e-mail for future utilization. The user's information will be attached to the e-mail as the user's wish.

Poor platform and interface of mobile devices

The mobile users are not only move around with her mobile devices. Actually, most of the mobile users own more than one computer and mobile devices. That is, they also use the multiple computers in different time or different places. Moreover, some mobile devices provide a very simple e-mail client (especially for some mobile phones). These e-mail clients are too simple to reply an e-mail directly. To solve situation is very urgent for mobile users.

However, the method for solving the problem is easy. The return message identification serial number mechanism can also be applied here. Because each returned e-mail has its own serial identification number, the user may utilize any terminal to reply or process this specified message. The e-MARE will keep the returned results e-mail in the database for a specified time period. The stored e-mails are waiting the authenticated users to fetch any time. By this way, the user doesn’t worry about the type of devices he will use, the results are always waiting for the user's accessing.

Commands, Scripts, and Programs

By default, the user is allowed to write some kind of high-level commands on the subject of any operational e-mail. Our system also allows the user to write advance script languages and full-function programming language. The advanced programs can be written on the body of operational e-mail. We summarize the mechanism as follows:

- If the text in the body of operational e-mail are between the marks stared by '&&(Language type)[' and ended by ']&&', we call these text *operational codes*. The operational codes will be treated as the scripts or programs of the operational e-mail. Our system supports Perl, and PHP now, we also allow the user to send some shell commands of Linux to the server for execution.
- After the system has received the operational e-mail, it will extract the operational codes first. Then the system will judge the language types by the *Language Type* part in the start mark. Currently, our e-MARE supports three types of language, Perl, PHP, and shell script program of Linux.
- In general situation, all the results are collected through standard I/O by the system. That is, if we don't consider the security of the implementation, all programming languages can be adapted in our framework.
- To maximize the flexibility of the system, e-MARE also allows the user to specify an e-mail address or URL as the gateway between the user and the program. In some situations, the user also can specify some filename as the result of the program. The system may attach these files as the attachment of the result e-mail according to the type of mobile device of the user.
- Before the design of the user, he can use the &&PES (Programming Environment Status) commands to get the detail environment of the server. Again, all the services the system provided will be encoded into distinct identification serial number. These numbers can be used

in the user's operational e-mail.

Implementation of Multi-e-MARE

Due to the limitation of the network environment and the computing power of the server, to solve a very complex problem may need more than one e-MARE server. In such case, we propose the multi-e-MARE solution. In our design, all the e-MARE frameworks can be treated as an independent operating unit which is called e-MARE server. Each e-MARE server at least is composed of three component servers. They are *E-mail Server*, *Mobile Agent Execution Server*, and *Database Server*. These component servers can be settled in either one computer or three different computers. Because all the e-MARE servers are independent, these e-MARE servers may communicate each other independently. Of course, to migrate the application and mobile agent through these e-MAREs are also feasible. These multi-e-MARE models are shown in Figure 4.5 and 4.6.

All the design issues related to our system will be implemented in the following sections.

5.5 System implementation

5.5.1 System implementation basics

According to the objectives of our framework, the requirements of software and hardware for client side is low. The basic requirements of client side are the Internet networking ability and an e-mail client application. There are lots of mobile devices fulfill these basic requirements. The detailed implementation hardware and software which we were used in our framework are listed as follows:

- **Implementing programming language**: Java, Perl, and PHP.
- **Present mobile agent runtime platform**: IBM Aglet.
- **Web and database server**: Intel Pentium III 1.13 GHz, 346MB RAM, Microsoft Windows 2000 Server with IIS 5.0, and MySQL.
- **Mail server**: Intel Pentium II 450MHz, 512MB RAM, RedHat 7.3 with Sendmail.
- **Mobile agent execution server**: Intel Pentium III 733MHz, 512MB RAM, RedHat 8.0 with Perl 5.0.
- **Laptop computer**: Compaq Presario 800, Intel Pentium III 650 MHz,

196MB RAM; Asus S-series, Intel Pentium 1.03 GHz, Microsoft Windows Me, and Microsoft Windows XP Professional Edition, Outlook, and Outlook Express.

- **GSM cellular phone** (Build-In Email Application): Ericsson T65, SonyEricsson T230.
- **GSM cellular phone** (WAP-based Email Application): Siemens 6618.
- **High-end GSM cellular phone**: Nokia 3650.
- **Palm PDA**: TRG Pro, Palm OS 3.51, Eudora Email client software.
- **Pocket PC PDA**: Acer n10, Pocket PC 2002.

In summary, in our framework, the device of client side should be a device with the Internet networking ability, and it also has to provide the e-mail send/receive client application. Conversely, the device of server side can be any desktop computer which is also connecting to the Internet with reliable and high speed networks. The e-MARE is mainly implemented by Perl and PHP programming languages. Furthermore, except the Perl and PHP, some of built-in mobile agent and applications are encoded by Java and C which are aimed to execute at Linux server.

5.5.2 System server deployment

The major objective of e-MARE is to provide uniform interface of the heterogeneous mobile devices to retrieve the information form the Internet or private information sources on the local area networks. Because the e-mail is the most popular client application of the devices with networking ability, the user doesn’t need to do any extra effort for their mobile devices to surf and to access the Internet or the information sources of the private networks. For the developers, our proposed framework, e-MARE, will also save lots of works and costs for deployment of the client side. Because there is no extra deployment for most networked mobile devices, it is an easy and feasible solution for mobile devices to access the information on the Internet or outside the low and unreliable wireless networks.

The proposed e-MARE framework is implemented in several programming languages under some different platforms. Thanks to the simplicity of e-mail protocol, most of the mobile devices provide e-mail client. However, the back-end server of e-MARE has to be installed some programs to activate the consequence processes. Firstly, we implement

ManagerAgent by Perl language. This agent is responsible for accepting the operational e-mail, parsing and extracting the commands of the e-mail, and dispatches the delegated works to appropriate agents or servers. The blackboard agent is implemented in PHP homepage language under RedHat Linux operating system with Apache and MySQL server. It provides HTTP/CGI interface for mobile agents to access the information of the Database Server.

As mentioned earlier, user can install her services via the WWW interface of e-MARE. All of the services are registered in the database server. To facilitate the interface, we design a set of PHP and Perl programs under Apache of the database server. These programs provide the user a WWW interface to add, remove, and edit all services in a single uniform environment. The user also can upload her program to her home directory for later executing.

The program runtime servers named *Mobile Agent Execution Server* are installed on a Linux server. These servers will be accepted the requests to upload some Perl programs into the system, executing these programs and collecting the results. Some built-in mobile agents are coded by Java, they are also provided in one of those servers. Of course, the coordinator server of e-MARE for coordinating all business is also a Linux server. The sendmail service of e-mail server will be always activated for accepting all the incoming messages.

The server deployment of our prototype is shown in Figure 5.4.

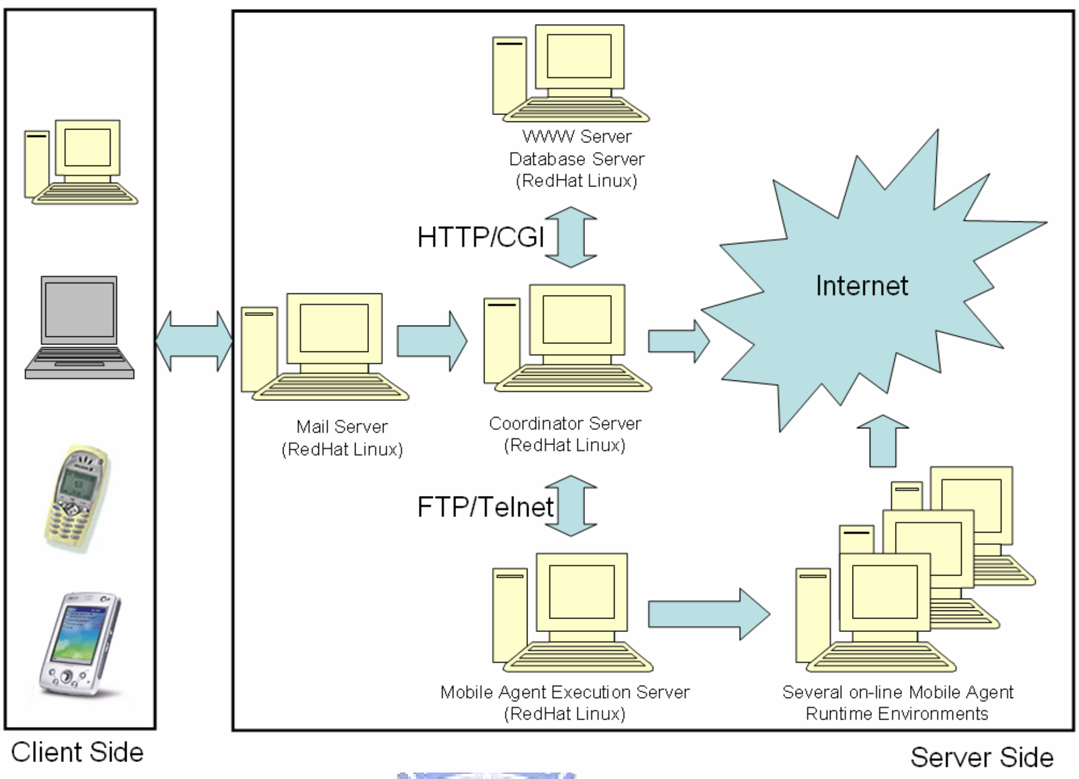


Figure 5.4: The server deployment of e-MARE.

In the e-MARE, all of works are done at server side, thus both the computing power of server and the services provided by the server are very important considerations. Different type of commands and data has different processing procedure. The *ManagerAgent* dispatches those commands and data to one of the appropriate stationary agent (*SystemStatusAgent*, *QueryAgent*, *IssueAgent*, or even itself) according to the type of their commands.

The front end of e-MARE framework is basically a mail server. In the beginning, it needs to modify the mail forwarding setting to give the system with the ability of awaking the *ManagerAgent* and of forking a child process to handle the user's request. We use the standard protocol of the Internet among the *Mail Server*, *Coordinator Server*, *Database Server*, and *Mobile Agent Execution Server* to keep the system's flexibility. Consequently, the system developer may easily add the extra server to expand the capacity of the e-MARE framework.

The protocol used between the coordinator server and database server is HTTP. The *QueryAgent* encodes the querying URL string according to the user's request. The URL will be sent to the web server of the database server and *QueryAgent* will retrieve the interested information from it. In this case, the database server accepts the querying commands and extracts the information from MySQL server. The results will be sent back to the *QueryAgent* through the HTTP protocol. Based on the HTTP protocol, in the e-MARE, all the queries that users have to know is the encoding rule of URL. It is easy for system developer to add more servers in our framework.

In order to balance the load from multi-users, the protocols between coordinator server and mobile agent runtime environments are also the standard protocol of the Internet, such as FTP and Telnet. FTP protocol is utilized to upload the mobile agent program to mobile agent runtime server when the *IssueAgent* is delegated to issue a mobile agent program. The *IssueAgent* uses Telnet protocol to execute uploaded program remotely, and collects the results from the standard I/O of that server. For some advanced mobile agent program, the result of execution also can be stored at the user directory of the server, and then download it back to the *IssueAgent*. All the results from the standard I/O and the files can be wrapped into an e-mail by coordinator server, and send it to the user's e-mail account through the e-mail protocol of the Internet.

5.5.3 System components and services supports in e-MARE

Figure 5.5 shows the system's components and their relationship. The stationary agents that include *SystemStatusAgent*, *QueryAgent*, *IssueAgent*, and *Result-slicing Agent* are all implemented in our proposed framework. The *Mail Server*, *WWW Server*, *Database Server*, and the *Mobile Agent Execution Server* are connected by high-speed local area network. Because the client may desire to access the Internet through e-MARE, the mobile agent execution server have to equip the Internet access ability. Of course, it can prepare more than one mobile agent execution servers in the framework across the Internet.

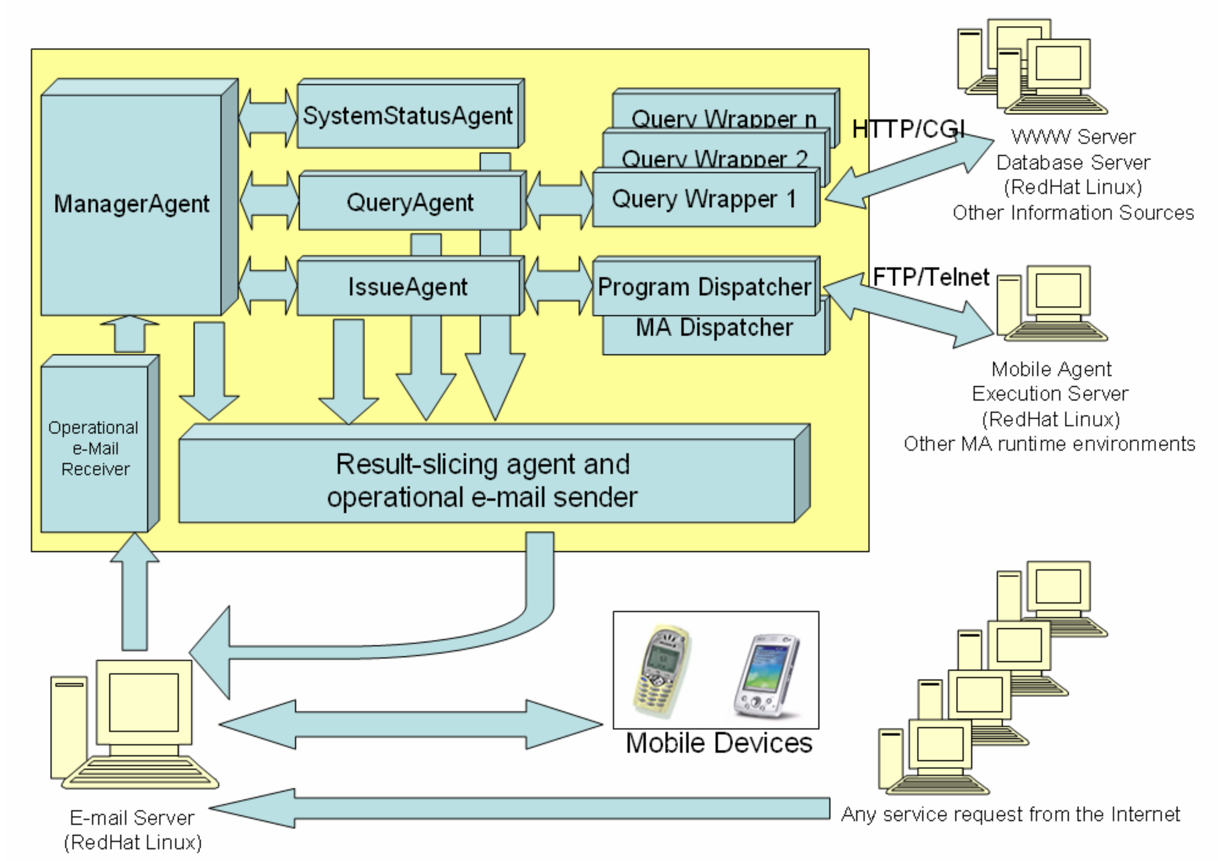


Figure 5.5: The system components of e-MARE.

The *ManagerAgent* parses operational e-mail and make a decision to dispatch the appropriate commands and data to one of the three stationary agents, *SystemStatusAgent*, *QueryAgent*, and *IssueAgent* respectively. There are three cases may happen in the succeeding process as the follows:

Simple status query or setting: If the contents of operational e-mail are just some simple queries about the system status, or the simple status setting operations, the works could be done by *ManagerAgent* itself. The *ManagerAgent* will transfer the results to *Result-Slicing Agent*. The *Result-Slicing Agent* will wrap the query results as the contents of e-mail, and reply them to the user directly.

Information retrieval services: If the contents of operational e-mail are the query commands for the information retrieval provided by the system, the *ManagerAgent* will wrap the related query commands, and then dispatch them to *QueryAgent*. Applying XML-based wrapper generator [43], the system developer can add some new services into the server. The specified wrappers for access these services can be easily plugged into the system. In our system, there is one RedHat Linux server with MySQL database system as the platform for executing the database queries. For some simple queries, the *QueryAgent* will access the database server by appropriate commands through some specified wrapper via HTTP/CGI protocol. All of these wrappers are treated as the components of the system. Again, the returned results are also transferred to *Result-Slicing Agent* for replying to the mobile user. For those complex and advanced queries, the system developer can plug the specified wrapper into the system.

Simple and advanced commands and script programs: If the contents of the operational e-mail include simple commands or even a script program, the *ManagerAgent* will dispatch the extracted commands or scripts to *IssueAgent*. The *IssueAgent* will prepare the commands or scripts as the distinct files, then send these files into the mobile agent execution server by FTP protocol and Telnet protocol. Because the *IssueAgent* has the user accounts of that server, and it also have the rights for executing any programs on the server, the mechanism will works on any UNIX-based with Telnet and FTP protocol with the execution and accessing rights. It is very flexible for mobile user to utilize this mechanism to execute any program on the remote UNIX-based server. That is, every program which can be executed on the UNIX-based can be written and be controlled via mobile devices.

In execution server, we do not restrict the ability of the remote executed program. That is, every program issued by *IssueAgent* run at execution server may contact to the outside of the server. Many mobile agent runtime environments and multi-search information retrieving systems available on the Internet can also be connected from these programs. Furthermore, if the mobile user knows how to connect these outside services, she also may program a mobile agent, and dispatch it to those systems to achieve her purpose.

It is not the only way to design and operate the mobile agent of remote host through the mobile device. Actually, we suggest the user to design, operate, and setup the mobile agents at desktop PC through the WWW interface. The WWW interface also provides the ability for the user to assign specified mobile agents to some specified ID. Thus, the user may simply use the ID to operate the mobile agent on the mobile device. This method will greatly reduce the user's effort on the mobile devices. We provide some instructions for the user to query the ID of user-defined programs and mobile agents.

5.5.4 Commands supported in e-MARE

The e-MARE supports a complete set of commands for user to use mobile device to conveniently operate and manage the remote mobile agent through e-mail protocol. These commands are embraced by ‘&&’ and ‘;’, and need to be put into the subject of operational e-mail. There are four subsets in the commands set. We briefly describe the command set as the following paragraph. The details are listed and explained in the Section 5.8.

Basic command subset: The subset of commands without any extra arguments provide user to query the system’s status or to set some simple operations to the remote agent. The basic commands have &&SS, &&SF, &&SAW, &&SAP, &&SASE, &&SCS, &&PESS, &PESF, etc.

Advanced command subset: Many commands of the subset provide user to access the information on the Internet or private database on the local area network. Actually, many information retrieval works can be done by specifying the appropriate built-in basic and advanced commands. These commands need extra arguments which supporting users to operate the mobile agents and system more flexible. Next section shows some useful examples. The commands of the subset includes &&AFW(no,id), &&AFW(no), &&AGL(no), &&AQUERYC(no, keyword), &&AQUERYL(no, keyword), &&ARE(no, arg1, … ,argn), &&PASSWORD(password), &&ACS(target, time), &&ACSD(target_name, width, height, color_depth), &&APE(no), &&APES(no), etc.

Scripting language subset: The framework also provides users to write the scripting language for executing information retrieval works or launching other types of mobile agents. The script language provides more flexible and complex than the advanced commands for the user to finish her works. We choose the Perl language and Unix shell script as the script language in this subset to quickly access to e-MARE.

Programming language layer: To maximize the flexibility of e-MARE, the environment also supports expert programmer to encode their program executed on the remote server to solve her problem through mobile devices. The performance and power of traditional C-like programming language, such as PHP, is also good than the script language on solving problems. The program can be attached to the operational e-mail and write the command &&(language type)[/]&& to specify the language which the program will use. The e-MARE will extract the attachment and wrap it as a program, send it to a remote server for execution.

Finally, the results of execution will be sent back through the e-mail protocol.

In addition, considering to the difference of display on each mobile device, the system utilizes context-aware computing technology [47] to distinct the difference of each mobile device from device profile and user’s historical preference. The context-aware procedure will analyze device’s profile and user’s preference. So the result formation executed by *ManagerAgent* can be well done for each user. Of course, the user can also specify her preferences for displaying the results.

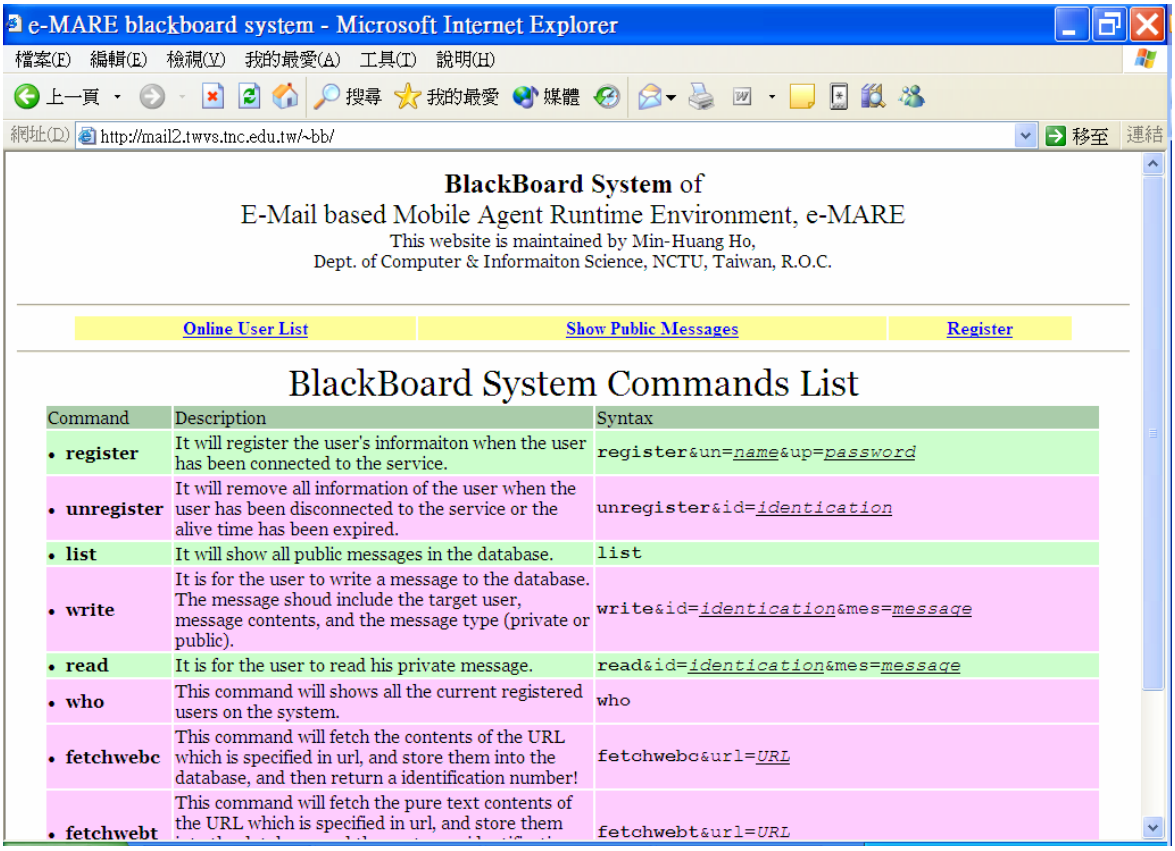
If the user does not specify the specific destination for receiving the reply, the default is to send the results to the user’s original e-mail account. The user also can specify the destination of results through e-mail or the user’s profile. Of course, the results can also be sent through the mobile phone’s *Short Message Service* (SMS) too. For example, if a user write an operational e-mail to retrieve the information on the Internet. After she has sent her operational e-mail, then maybe she will quit the current Internet connection later. As mentioned earlier, most mobile devices can’t easily connect to the Internet. In such situation, the best destination for receiving the results e-mail is the SMS of her mobile phone. In our design, the user can specify the command in the subject of operational e-mail as follows:

&&(0921222333)SS

The command means that the result e-mail should be sent to the mobile phone numbered 0921222333, and the command is ‘&&SS’. That is, the user wants the server to execute the command ‘SS’, and then return the results to the mobile phone numbered 0921222333 by short message service of GSM. Our framework also allow the user to specify the prefer returning method by WWW interface.

5.5.5 Communication mechanism of e-MARE

In general, the I/O ability of mobile device is restricted. To reduce the numbers of character is one of the most important considerations. In our framework, every data, arguments, and mobile agent programs can be stored in the database server of e-MARE. To manage these databases, we apply the modified blackboard mechanism. The blackboard is a database, and it is managed by a stationary agent, *BBCoordinatior*. That is, every mobile user and mobile agent wants to access the database must through *BBCoordinator*. To maximize the flexibility and compatibility of the accessing to the blackboard, the system mainly provides the HTTP/CGI interface. Every program or agent under Internet can read and write the information from the blackboard by the specified encoding URL string. For a mobile user without HTTP protocol, she also may write an operational e-mail to *BBCoordinator*, the results will be wrapped to an e-mail and will be replied to the user's mailbox. The commands for accessing *BBCoordinator* are also built in *ManagerAgent*. In our prototype system, we also implement a WWW interface for the user to easy access the basic information about the blackboard system.


e-MARE blackboard system - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http://mail2.twvs.tnc.edu.tw/~bb/
移至 連結

BlackBoard System of
E-Mail based Mobile Agent Runtime Environment, e-MARE
This website is maintained by Min-Huang Ho,
Dept. of Computer & Informaiton Science, NCTU, Taiwan, R.O.C.

Online User List | Show Public Messages | Register

BlackBoard System Commands List

Command	Description	Syntax
• **register**	It will register the user's informaiton when the user has been connected to the service.	`register&un=name&up=password`
• **unregister**	It will remove all information of the user when the user has been disconnected to the service or the alive time has been expired.	`unregister&id=identication`
• **list**	It will show all public messages in the database.	`list`
• **write**	It is for the user to write a message to the database. The message shoud include the target user, message contents, and the message type (private or public).	`write&id=identication&mes=message`
• **read**	It is for the user to read his private message.	`read&id=identication&mes=message`
• **who**	This command will shows all the current registered users on the system.	`who`
• **fetchwebc**	This command will fetch the contents of the URL which is specified in url, and store them into the database, and then return a identification number!	`fetchwebc&url=URL`
• **fetchwebt**	This command will fetch the pure text contents of the URL which is specified in url, and store them	`fetchwebt&url=URL`

Figure 5.6: The first web page of BlackBoard System of e-MARE.

The first page of our proposed blackboard system is shown in Figure 5.6. In the blackboard system, the first page will show all the available commands and their specified syntax of the system. All listed commands of the system stands for all functions of the *BBCoordinator* (bbcoordinator.php) As the Figure shows, to identify each user's right and available services, the first step to access the blackboard is to register the user's information to the system. That is. The user or the user's mobile agent or customized program has to write the following URL string to contact with the blackboard system:

http://blackboard_url/bbcoordinator.php?cmd=register&un=*username*&up=*secret*

In general, the *BBCoordinator* will first check the user's information of the e-MARE database. If the user is already the member of e-MARE, her information will be loaded into the *User Active Profile*, and waiting the further operations from the user. To identify the ownership of the user during the accessing process, *BBCoordinator* will return an identification serial number to the user. That serial number is the key for further accessing with the system. That is, the following commands (including send, write, and unregister) to connect to the database have to provide this serial number. We use the following instructions (PHP code) to assure the unicity.

```
<?php
   $id = hexdec(substr(md5(uniqid(microtime(),1)), 0, 6));
?>
```

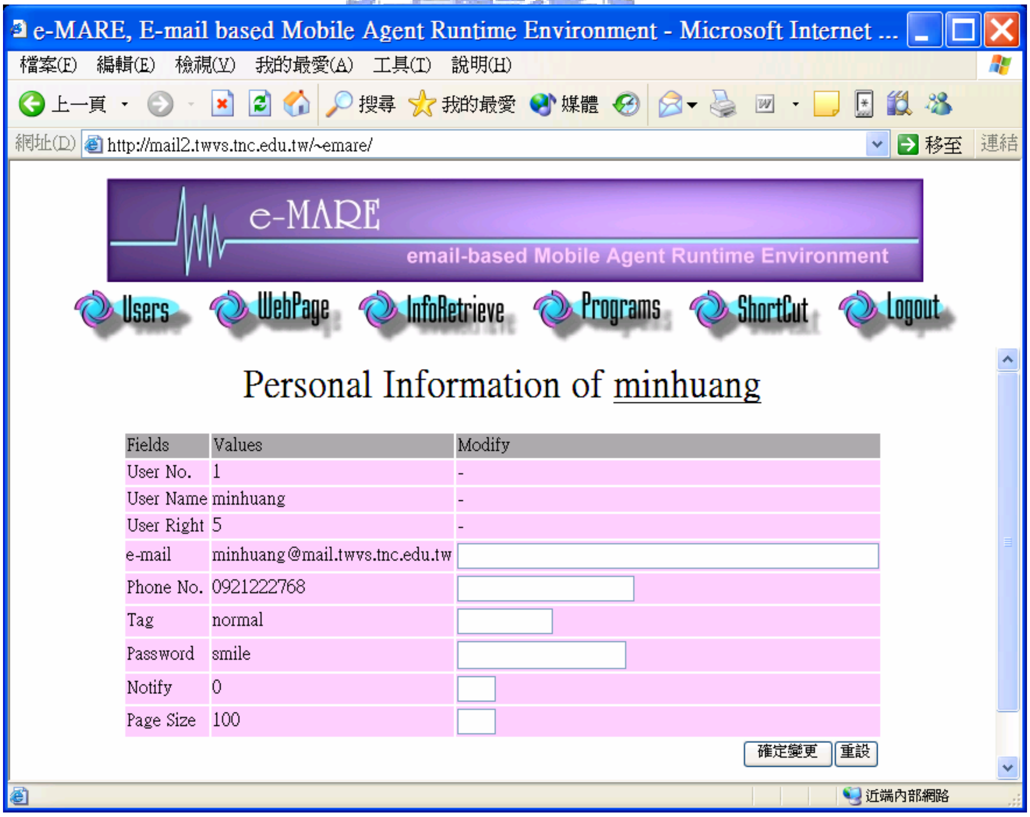
Of course, all values will be further verified the unicity. If the system has found that the $id has been duplicated, it will re-generate another serial identification number. On the other hand, if the user is not the member of the e-MARE server, an error messages will be returned to the caller (the screen of WWW browser, or even the return value of the function of the user's mobile agent or applications).

Just like the blackboard or notice board in the real life, each user can read and write the blackboard's database during the accessing process individually. The user may choose accessing the blackboard system either through the WWW interface of Internet browser or through a program accessing method. If the user has registered by browser, the session mechanism will be applied. Reversely, if the user has registered by programming interface, the identification serial number mechanism will be activated. However, to prevent the un-consistency occurs, the blackboard will restrict the hybrid accessing model. That is, if the user has been registered the blackboard system by WWW interface, the user will not allow to access the blackboard through the program until the user has been unregistered.

Moreover, we also apply the mailbox-based scheme [51] to establish the further communication between mobile agents and mobile users. In our design, every mobile user can create a new mailbox for her user-defined mobile agent. These mailboxes are tightly associated with its mobile agent. *MBCoordinator* is responsible for managing these mailboxes. It will accept any incoming messages and redirect these messages to appropriate mobile agents or mobile users. Any mobile user can send a command requesting the *MBCoordinator* to provide the directory of the mailbox for every mobile agent.

5.5.6 Result-slicing mechanism of e-MARE

Due to the limitation of display size of mobile devices, it is hard to read the whole results at once for most mobile devices. To ease reading the results from e-MARE, we also develop a result-slicing mechanism. That is, the user can specify the results return method. There are two return methods: slicing method and full method. The slicing method is special for the small display size devices. If the user chooses the slicing method, the system will buffer all the results into the database, and give the user one page document and an identification number. In normal case, a page has about 100 characters parts of original documents. According to our experiments, the first 200 characters of original result document usually are not the best desires for the user. So we provide the user the ability to specify the position of original result document for the fist returning page. The remaining pages can be accessed by specifying both the identification number and the page-turning direction. For a larger display devices or little return results, the user can also specify the full method to display all of the results at once. The user can modify his preference by WWW interface, which is shown in Figure 5.7.


e-MARE, E-mail based Mobile Agent Runtime Environment - Microsoft Internet ...
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http://mail2.twvs.tnc.edu.tw/~emare/
移至 連結
e-MARE
email-based Mobile Agent Runtime Environment
Users
WebPage
InfoRetrieve
Programs
ShortCut
Logout
確定變更
重設
近端內部網路

Personal Information of minhuang

Fields	Values	Modify
User No.	1	-
User Name	minhuang	-
User Right	5	-
e-mail	minhuang@mail.twvs.tnc.edu.tw	
Phone No.	0921222768	
Tag	normal	
Password	smile	
Notify	0	
Page Size	100	

Figure 5.7: User's personal information page.

As shown in Figure 5.7, the user may modify the field, Page Size, to specify the return result characters in one time. The default value is 0, which means returning the whole results. Furthermore, to ease the implements of the system, the system applies some distinct commands for slicing the return results. For example, if the user wants to get the whole page in one fetching process, the commands, &&AFW(*no*) fulfills his requirement. Reversely, if the user is using the small display mobile devices, the appropriate commands instead of &&AFW(*no*) is &&AFWDB(*no*). Different to &&AFW(*no*) commands, the &&AFWDB(*no*) will first fetch the whole page of the target WWW page, then partition them into many pieces according to the user's preference, and store them all into database of blackboard. After finished fetching and storing, the system will specify an identification number to the user for future accessing.

For example, if the user composes an operational e-mail with the command ' &&AFWDB(9)' and sends it to the e-MARE system, the system will fetch and store the contents of specified webpage, and return the following contents to the user's mailbox as soon as possible:

```
The fetching result for http://news.yam.com/general is shown as follows:
    Access id is 4169272
    Page size is:100

    Page 2
    -5-21  星期?09:21
    即時總覽
    ｕｄｎ｜中廣新聞網｜
    中央社｜法新社｜路透社｜民視 | TVBS | ETtoday
    中央社
    台股開盤上漲 20.75 點  為 5836.08 點
    《05/21 09:20》
    氣象局發布中南部大雨特報  今明各地有陣雨
    《05/21 09:20》

Service Time:0.425701141357422
```

As shown in previous paragraph, the access id is 4169272, and the current page number is 2. That is, if the user wants to view another page of the contents, all he needs is just to compose another operational e-mail with the subject '&&AFWIDP(*no*, *pagenumber*)' , for example &&AFWIDP(4169272, 10), and the e-MARE system will soon send the following contents to the user's mailbox:

```
ID is : 4169272
Page No.: 10
    《05/21 09:20》
    喬丹今天將在港出席其產品展覽會
    《05/21 09:20》
    Ｇｏｏｇｌｅ將推免費 PC 搜尋軟體  對微軟「先發制人」
    《05/21 09:20》
    美國海 x 官校改校歌  取消「男兒」字眼
    《05/21 09:20》
    雅典奧運排球館失火    延燒一小時後撲滅
    《05/21 09:20》
    華衛二號的火箭載具    啟用四項新設 p
    《05/21 09:20》
    娃    洧    涪羌艨 O,法理難容
    《05/21 09:20》
    新任教育部長杜正勝
    《05/21 09:00》
    WHO 兩年內入會?  民間團體認為仍需努力
    《05/21 09:00》
    華衛二號升空  國科會強調是與國際接 y 而非太空競賽
    《05/21 09:00》
    全球第一座虛擬教堂遭自稱撒旦的民眾入內大放厥辭
    《05/21 09:00》
    一千美元魚子醬蛋餅終於有人點用了！
    《05/21 09:00》
    TOP
    路透社
    高爾夫-伊梅爾曼藉頂腹式推桿轉運,歐洲球手錦標賽瑤    疆 C 領先
    《05/21 09:20》
    ＜美國經濟焦點＞地區製造業成長放緩,初請失業金人數增幅超預期
    《05/21 09:20》
    網球-智利擊敗德國,進入世界團體錦標賽決賽
    《05/21 09:00》
    [基金觀點]政治及美升息因素消化後,科技及出口類股料引領韓股反彈
    《05/21 09:00》
Service Time:0.186314105987549
```

Of course, the fetched documents should not always be kept in the database. Actually, the e-MARE system will specify one TTL (Time To Live) value (default value is 20) to every document in the database, and a background daemon will reduce these values every five minutes and monitor the threshold of those value. If the TTL of any documents have reached to zero, they will be wiped out from the database.

5.6 Detail workflow of e-MARE

In our implementation, we construct a Redhat Linux server to be the main server of the e-MARE framework. Lots of works of e-MARE are worked at the server. From the system developer's point of view, the detail workflow is shown as follows:

1. At first, we construct an e-mail account on a mail server which its platform is RedHat Linux with sendmail service. In our prototype of e-MARE, we construct a public service account 'emare@mail.twvs.tnc.edu.tw' at the server named

203.71.176.4. That is, if any client needs to access the service of e-MARE, all they need is to send an operational e-mail to this e-mail address ‘emare@mail.twvs.tnc.edu.tw’.

2. Second, we modify the forward setting for the e-mail address. That is, after the account is received some messages, the sendmail daemon of Linux will forward that message to the *ManagerAgent* (managerAgent.pl) This is the start point the e-MARE framework, and many works which have been introduced in Figure 5.5 are implemented in this program.

3. The subject of the mail is the identification for the operational e-mail. If the subject has the specified id, that stands for this e-mail is operational e-mail. Then, the system will begin to parse the subject and the body of the operational e-mail, and activate the appropriate stationary agent to handle the special contents encoded in the body of e-mail. If the system can’t find any id in the subject of e-mail, this e-mail will be treated as the normal e-mail, this message will simply be dropped.

4. After the extraction phase of the system, the *ManagerAgent* will judge the type of the operational e-mail. If the extracted commands are just the simple status query or status setting commands, these jobs will be done by *ManagerAgent* itself. That is, *ManagerAgent* will return the requested status or modify the system status as the user’s desire, and reply the results to the client through the e-mail protocol.

5. If the purpose of the operational e-mail is to access the private database of the private network, *ManagerAgent* will assign *QueryAgent* (queryAgent.pl) to handle the succeeding works. In the RedHat Linux database server of our proposed prototype, we provide a website (Apache Web Server) for database querying. All of the queries can be done through the web interface. That is, the *QueryAgent* has to encode the appropriate URL string with appropriate authentication mechanism, the web server will try to get the user’s desired information and return them to *QueryAgent*. After the *QueryAgent* has received these results, it will send them to *ManagerAgent*, and *ManagerAgent* will send them to the client through the e-mail protocol.

6. If the operational e-mail contains a more complex script program, the works for treating these script codes will be delegated to the *IssueAgent* (issueAgent.pl).

The script program extracted from the operational e-mail will be uploaded to home directory of runner in 203.71.176.6 (*Mobile Agent Execution Server*). *IssueAgent* will execute the 'sendcmd.pl' to execute that script program remotely, and collect the output of that program. If the result of execution was produced into a file, the system will also download that file by FTP protocol. All the return data or files will be transmitted to the *ManagerAgent*. The *ManagerAgent* will take care of replying process for the client user.

7. Actually, in *ManagerAgent*, we design a *ReplyAgent* (replyAgent.pl) to take care of all the works about the reply to the clients.

5.7 Instruction set

In our framework of e-MARE, to comprehensively serve the mobile user and to allow the user to operate and manage their mobile agents remotely through the e-mail protocol, we design and propose a complete instruction set to apply to the system. Before list and explain the detail instruction set, we classify the control commands into four layers:

1. **Basic command layer**: the basic and simple commands without extra functions and arguments. This layer of the commands is always the commands that provide the user to query the system's status or to set some simple operations to the remote agent.

2. **Advanced command layer**: the more complex commands, in general speaking, most commands of this layer are accompany with some arguments. These arguments allow the user to operate the mobile agents and system more flexible.

3. **Script language layer**: our framework also allows the user to write the script language for operating his information retrieval projects or his mobile agents. The script language has most characteristics of the normal programming language. It provides more flexibilities than the simple commands for the user to finish his works, but it also means that the process for script language is more difficult than the commands. For quickly access to e-MARE in our framework, the specification of the script language used in the framework is a new design especially for our framework.

4. **Full function language layer**: for a power user of the computer area, sometimes he will wants to write some programs to solve his problem by his mobile devices. In our proposed prototype system, we also support Perl, PHP, and shell script language.

The instruction set of our framework is classified into four layers as mentioned above. The detail descriptions for these instructions of first two layers are shown as follows:

Basic command layer: This kind of instructions is the basic and easy instructions. They always have no extra argument. The major purpose of these kinds of commands is to obtain the system status easily and quickly.

- &&SS: return the brief current status of the system.
- &&SF: return the whole current status of the system.
- &&SDB: return the debug information of the system.
- &&SAW: return the available WWW pages for retrieval. These WWW pages can be settled in advance via WWW interface of e-MARE server.
- &&SAP: return the current available services of the system. The services usually are the system shell commands or remote programs provided in advance via WWW interface of e-MARE.
- &&SASE: return the current available search engines of the system. These search engines also can be settled in advance via WWW interface of e-MARE.
- &&SCS: return the supported client type and content of the system. It is convenient for the user to specify the built-in type for displaying preferences.
- &&PESS: return the brief status of the programming environment of the system
- &&PESF: return the complete status of the programming environment of the system.

Advanced command layer: This kind of the command is the command that with some arguments. The user can control the remote mobile agents through these commands or accessing to the private database and the document on the Internet.

- &&AFW(*no*): fetch the ASCII text from the *no*th web page. That is, the tags and the script language statements of the fetched document will be cut out. Only the pure text will be kept, and returned to the user’s mailbox.
- &&AFWDB(*no*): almost the same as &&AFW(*no*), except this command will partition the results and store them into the database. That is, the results will be partitioned into several pages. Every page will be stored into the database of blackboard system. The commands will return the contents of second page and the reference id to the user’s mailbox for future referencing. The page size is the user’s preference, and it can be retrieved from user’s profile

database.

- &&AFWIDP(*id*, *pno*): the command provides the user to fetch the sliced web pages which were stored in the database in the previous fetching process. Where the *pno* is the page number and the *id* is the identification for the fetching process.
- &&AGL(*no*): listing all the links of the *no*th web page.
- &&AGLDB(*no*): listing all the links of the *no*th web page, and store those links into database of blackboard system. After the system has finished the storing process, it will return an id to the user's mailbox for future querying.
- &&AQUERYC(*no*,*keyword*): searching the *keyword* on the *no*th search engine through the Internet, and return all the contents of the search pages. If the specified search engine is a meta-search engine and providing some kind of SQL-like querying language, the *keyword* can also be SQL-like querying language string.
- &&AQUERYL(*no*,*keyword*): searching the keyword on the *no*th search engine through the Internet, and return all the links of the search pages.
- &&ARE(*no*, *arg1*, *arg2*, ⋯, *argn*): to execute the *no*th services provided by the system. This command also can specify the arguments for the demand of the service's execution. The service number can be got by the command &&SAP. For example, if a user wants to quote the current price of her stock on the stock market. We assume the service number for quoting stock is no. 2. Obviously, the argument of this service is the stock id. In such example, the command line can be encoded as '&&ARE(2, 2303)', and this line has to be settled as the subject of the operational e-mail.
- &&AREP(*arg1*, *arg2*, …, *argn*): this command provides the mobile user to write a program in the body of the operational mail. After the system has received the mail with this command, it will first extract the program, and packs into an executable file, and put it into the temporary directory of the execution sever. For the distributed running model, the output of that program will not be captured by system. Thus, returning the results to the user is the responsibility of this program. Combine with the blackboard messaging mechanism, this command let the user has the ability to write the operational e-mail based mobile agent, and traverse across the e-Mare servers on the

network. The code will be transported by e-mail, and the temporary data, messages, and results can be exchanged by blackboard mechanism.

- &&PASSWORD(*password*): this command is used to transfer the user's account and password to the system. The system will obtain the user's right, and user's profile according to the user's e-mail account and password. If the user does not provide her password for verification by this command, the user's right will be very restricted.
- &&ACS(*target*, *time*): the user can utilize this command periodically active transferring the client's information to the system. In most cases, the context-aware procedure will automatically find the best way to adjust the format of context for the client's mobile device, but finally the user's opinion has the first priority that the system has to obey. So the system provides this command to let the user specify the client's display type. Of course, the type is also a number settled by the system. The user can obtain these numbers by command &&SCS too. In our design, the client's display type can also set on the web site through the Web browser on the Desktop PC. Beside, the argument *time* is the specified response time.
- &&ACSD(*target_name*, *width*, *height*, *color_depth*): this command lets the user to detailed specify the characteristics of the client's devices. This is required if the user want to adjust the displaying arguments for her mobile device more accurately.
- &&APE(*no*): this command provides the user to specify the parallelism for program execution. If the user specifies this command, the following *no* number of commands will be executed in parallel. The following commands are all independent, the execution result will be sent to the user respectively.
- &&APES(*no*): this command is the same as the &&APE(*no*), except the result return mode. Different to the previous command, the results for different command will be collected to one single result, and then send it to the mobile user's mailbox.
- &&SYS(*command)*: this command provide the user to directly execute the system commands of the operating system of e-MARE. In general, the commands which can be launched are dependent to the kind of operating system which the e-MARE used. The executing results which were outputted

to standard I/O will be directly transmitted to the user’s mailbox through the e-mail protocol. In general, the commands always are applied to the cases which the user wants to know the status of the operating system of e-MARE server.

- &&RDON(*no*): activating the specified number of remote displaying site. The system will check the specified site if it is registered or not. If the specified site has not been registered, the system will register the site for the mobile user, and allow the user to access the information on the *Public Display Point*.
- &&RDC(*id, siteno, no*): fetching the document of on the web page of specified site of *Public Display Point*. The *id* is an identification number for verification between the mobile user and the system. It is generated while the mobile user has registered to the system. The *siteno* is the number of *PDP* among all the available remote display points, and the *no* is the number of the URL item which is listed on the web page of specified site.
- &&RDL(*id, siteno, no*): almost the same as the &&RDC command, except the return results of the command is the filtered links rather than the original document itself.

There are some accounts for the specific usage for e-MARE, they are also listed as follows:

1. **emare@mail.twvs.tnc.edu.tw:** The main entry of the e-MARE. It is the public account for registered and unregistered user. All the operational e-mails have to be sent to this account. There is a Perl program called managerAgent.pl which will be initiated when a legal operational e-mail has arrived.

2. **worker1~worker2@mail.twvs.tnc.edu.tw**: If the operational e-mail specifies some works which have to be executed in parallel, the *ManagerAgent* will dispatch those works to worker1, worker2, and worker3 respectively. It also has two parallel mechanism choices. The one is letting the workers reply the results to the user directly. The other is letting the workers reply the results back to the *ManagerAgent*, and then, the *ManagerAgent* can get the opportunity to collect and analyze all the results from the workers. This mechanism can optimize the quality of the results for the user’s requests.

3. **magent@mail.twvs.tnc.edu.tw:** In this account, all the built-in stationary agents and mobile agents are settled. The user can get the directory of these agents by the simple commands provided by e-MARE. Then the user can just specify the number of the lists that he wishes to execute. The only work that the user has to do is just to compose a well-format operational e-mail, and provide the appropriate data for the agent's execution.

4. **runner@mail.twvs.tnc.edu.tw:** In this account, we provide the user has the ability and flexibility to install the user-defined agents. In our design, we provide a WWW interface for the user to install his programs (mobile agents or applications) into the directory of this account. These agents can be called by the user soon or later. The screen snapshot of WWW interface will be shown in Figure 6.8 and Figure 6.9.

Chapter 6

Experiments and Examples

In this chapter, we present many practical and useful examples for the mobile users to access the information on the private network and the Internet via the e-MARE framework. In e-MARE, to access the information via the mobile devices, the network connection between mobile devices and e-MARE server has to be created while the operational e-mail is transmitting. The network connections can be traditional GSM/Internet networking, GPRS, IrDA, or BlueTooth. As we can see the examples in this chapter, through e-MARE framework, the user's mobile devices will be leveraged into a mobile information provider which may provide the user's desired information anytime anywhere.

6.1 Software and hardware details

To demonstrate the feasibility of e-MARE, many experimental demonstrations are described in this chapter. The client platforms which were tested all these demonstrations are Ericsson T65, SonyEricsson T230, Siemens 6618, and Nokia 3650 cellular phones, TRG pro and Acer n10 PDA. Of course, the MS Outlook of MS Windows XP can also be the appropriate platform of these demonstrations. T65 has a simple built-in e-mail application. Siemens 6618 provides a simple but feasible WAP browser. Nokia 3650 is a typical advanced mobile phone which is equipped a powerful operating system (Symbian Series 60) and comprehensive e-mail sending/receiving client. We select TRG Pro as the example of Palm OS, and Acer n10 as the example of PocketPC. Because e-MARE is an e-mail based framework, in general, any platform with e-mail sending/receiving application can also be the appropriate client. That is, for those cellular phones which are not equipped e-mail application, they also can send and receive the e-mail by STK (SIM Tool Kit) or WAP browser, which are provided by the GSM network provider.

In our experiments, the mail server of e-MARE is a Linux server, which is Intel Pentium II running at 450MHz (Compaq Proliant 800). The agent execution server is the same Linux server. The blackboard and database server is also a Linux server, which is Intel Pentium III running at 700 MHz (Compaq Proliant DL380). The major e-mail service on the mail server is sendmail. The HTTP service on database server is provided by Apache, and the database which is executed at Linux server is MySQL. To provide the HTTP/CGI facility, we programmed some PHP files with Apache web server. The client mobile device can connect to the mail server of e-MARE through the GPRS via GSM network. These servers of e-MARE are all connected via TPC/IP with 100Mb Ethernet.

To receive the incoming message from the mobile users, the default public service e-mail account has been constructed in the mail server, which is emare@mail.twvs.tnc.edu.tw. That is, the mobile user has to write an operational e-mail to this account if she wants to obtain some services from e-MARE.

6.2 The measurements for composing the operational e-mail

To demonstrate how long the composing time is for a typical operational e-mail, in this section, we show some measurements for composing an operational e-mail by several different mobile devices. In this experiment, we assume the user wants to obtain a list about all of the available search engines of e-MARE framework. Typically, the user has to compose an operational e-mail with subject titled '&&SASE;', and send it to 'emare@mail.twvs.tnc.edu.tw'. We test several devices with different services and applications to accomplish the query purpose. The measurements are shown as Table 6.1.

Table 6.1: The measurements for composing a simple operational e-mail.

Mobile device Name	Composing Time (minutes:seconds)	Descriptions
Nokia 3650	1:44	Through the built-in e-mail application with GPRS services.
Siemens 6618	2:40	Through the built-in WAP browser with GRPS services.
SonyEricsson T65	1:50	Through the built-in e-Mail application with GPRS services.
Acer n10 PDA	0:28	Through the built-in e-Mail application with 10Mb CF Ethernet Network Interface Card.
TRG Pro	0:32	Through the built-in e-Mail application with IrDA connection to notebook PC.

The input ability for PalmOS-based PDA is the same as PocketPC OS. Thus, the composing time is about equal to the Acer n10 PDA. It is worth to say that the measurements are only measure the composing time for a typical operational e-mail. The listed time is not including the actual connection time between the mobile devices and the network service provider.

No matter what is the mobile device that the user used, the results are already in the user's mailbox only few seconds later. That is, in general, the user always obtains the desired information at the user’s next mail receiving step.

6.3 The steps of the mobile users for accessing e-MARE

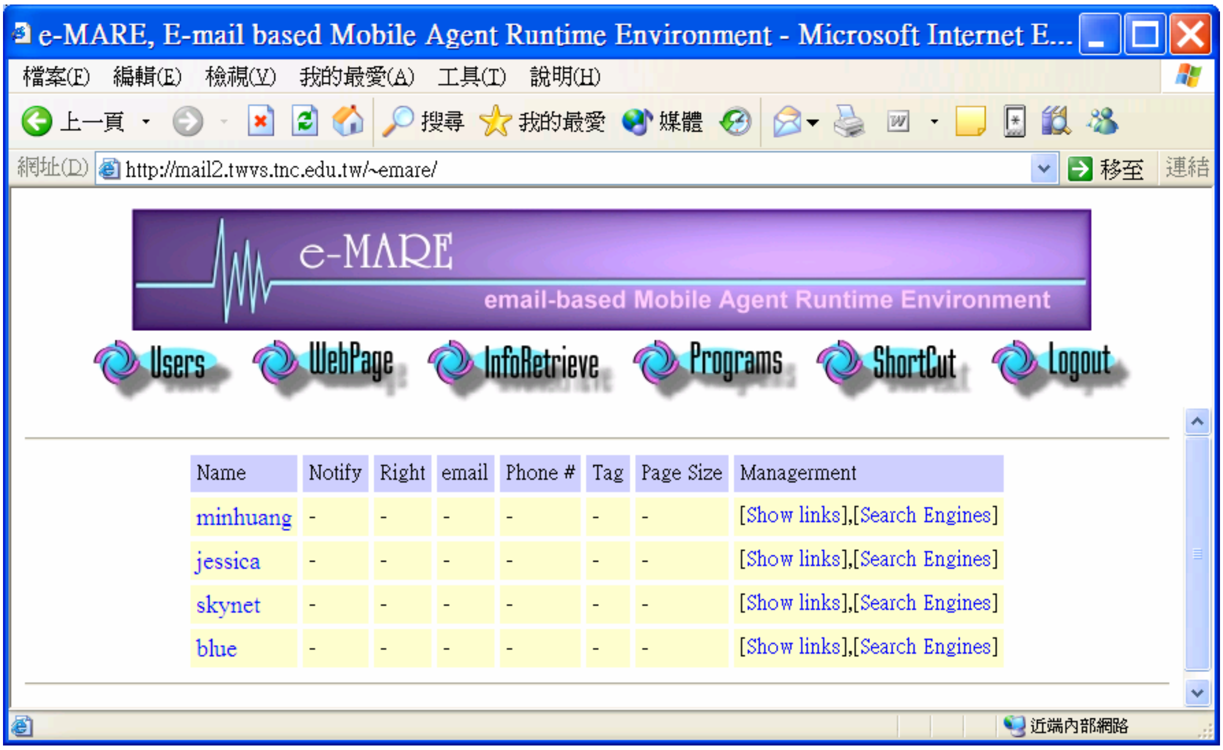
From the user’s point of view, if a mobile user wants to query some information on the Internet via her mobile devices, her mobile device has to allow her to connect the Internet. In this case, we suppose her device is SonyEricsson T65 with GPRS over GSM network, and this device has a built-in simple e-mail application for sending and receiving the e-mail on the Internet. The steps for the mobile user to access to e-MARE are listed as follows:

1. The mobile user composes an operational e-mail under the ‘Message’ menu of T65. The subject of this e-mail has to provide the user’s password. The simple commands can also be encoded at subject too. If the commands will be more complex, these commands have to be embedded into the body of this operational e-mail.
2. This e-mail has to be sent to the public service account of e-MARE, emare@mail.twvs.tnc.edu.tw, by selecting the ‘send now’ menu item of ‘send and receive’ menu.
3. Few minutes later, the mobile user may select ‘send and receive’ menu item again. In most cases, the results or the acknowledgement message from the e-MARE server always are ready in the user’s mailbox.

The above steps are very simple and intuition. In most cases, if the user wants to operate the remote agents or applications of the server, the first step is to write an operational e-mail to initiate a list operation (such like &&SAP) to obtain the available services, and then write an operational e-mail to initiate some specified services or agents by specifying the listed service’s number. All mobile agents and services can be prepared in advance by the WWW interface provided by the Web server of e-MARE.

6.4 An introduction to web interface of e-MARE

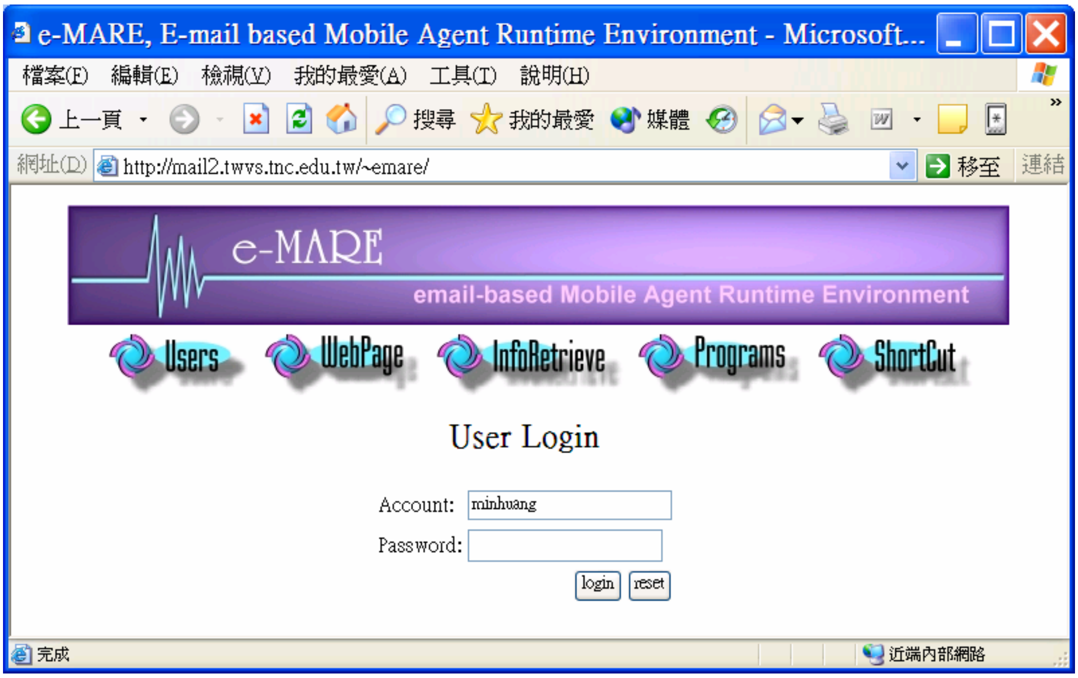
As the mentioned earlier, the e-MARE provides a web interface for the mobile user to maintain her personal preferences and install user-defined applications and mobile agents into the system. In this section, we will introduce the instructions of web interface of the e-MARE. The first page of e-MARE web site is shown in Figure 6.1.


e-MARE, E-mail based Mobile Agent Runtime Environment - Microsoft Internet E...
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http://mail2.twvs.tnc.edu.tw/~emare/
移至 連結
e-MARE
email-based Mobile Agent Runtime Environment
Users
WebPage
InfoRetrieve
Programs
ShortCut
Logout
近端內部網路

Name	Notify	Right	email	Phone #	Tag	Page Size	Managerment
minhuang	-	-	-	-	-	-	[Show links],[Search Engines]
jessica	-	-	-	-	-	-	[Show links],[Search Engines]
skynet	-	-	-	-	-	-	[Show links],[Search Engines]
blue	-	-	-	-	-	-	[Show links],[Search Engines]

Figure 6.1: The first page of e-MARE web site.

In the page which is shown in Figure 6.1, there are six main categories of functions which the user can manipulate. They are *User Management*, *Information Retrieval Database Management*, *Search Engine Database Management*, *Program Installation*, *Mobile Agent Management*, and *Application Management*. The *User Management* allows the system manager to add/modify/remove the user's information, and allows the users to modify their own preference. The basic preference of user at least including user's *name*, *right*, *e-mail address*, *mobile phone number*, *tag*, and the related information (i.e. information retrieval web page links, search engine links, etc.) which is associated to the users. By default, the user can not view anything except the user's name of the system until she login into the system. That is, at the first, the user who wants to access e-MARE system has to fill out her user name and user password in the web page. The process is shown at Figure 6.2.


e-MARE, E-mail based Mobile Agent Runtime Environment - Microsoft...
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http://mail2.twvs.tnc.edu.tw/~emare/
移至 連結
e-MARE
email-based Mobile Agent Runtime Environment
Users
WebPage
InfoRetrieve
Programs
ShortCut
完成
近端內部網路

User Login

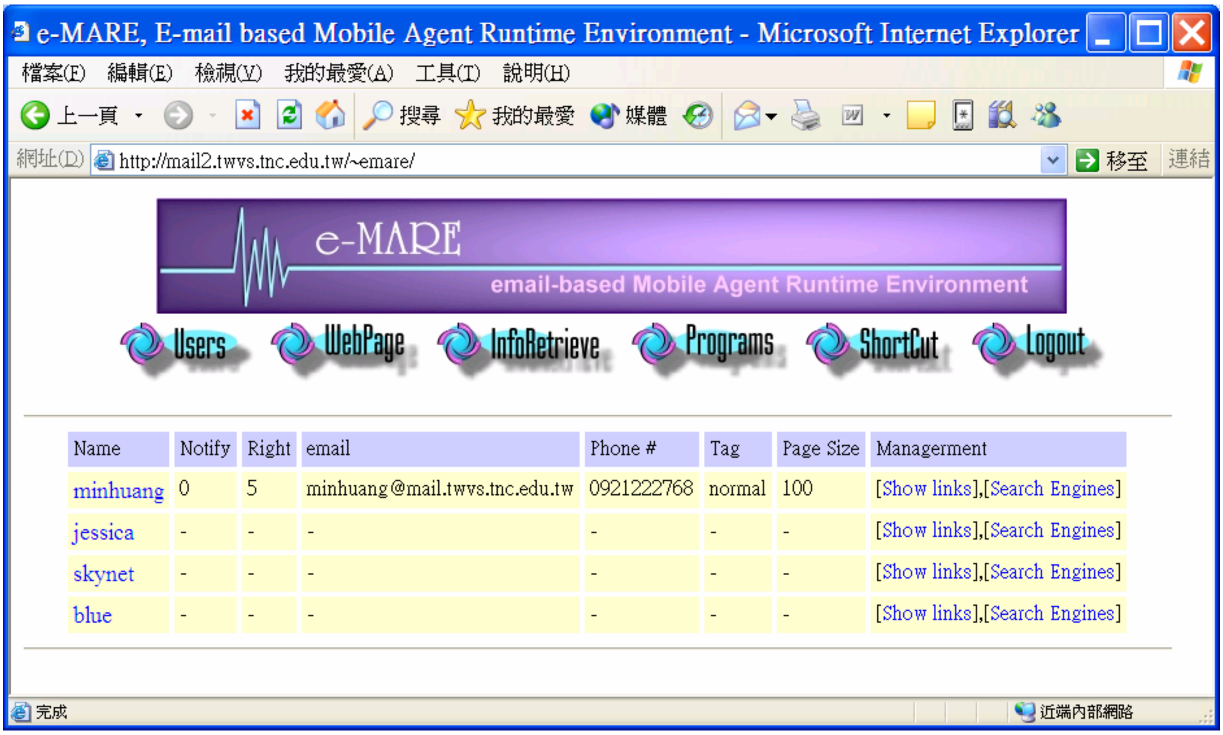
Account: minhuang

Password:

login reset

Figure 6.2: The user login page.

After the user has been logged into the system, the left-top corner of the web page will show the user’s login name, and all of that user’s information will be listed in the page too. It is shown as Figure 6.3. Meanwhile, the user may choose any category to manipulate her desired operation about the database of e-MARE. The system will display the information which is associated to the logged user when the user chooses any link on the web page. For example, through the **WebPage** link, the system will display all links of the user registered web pages. It is shown at Figure 6.4.

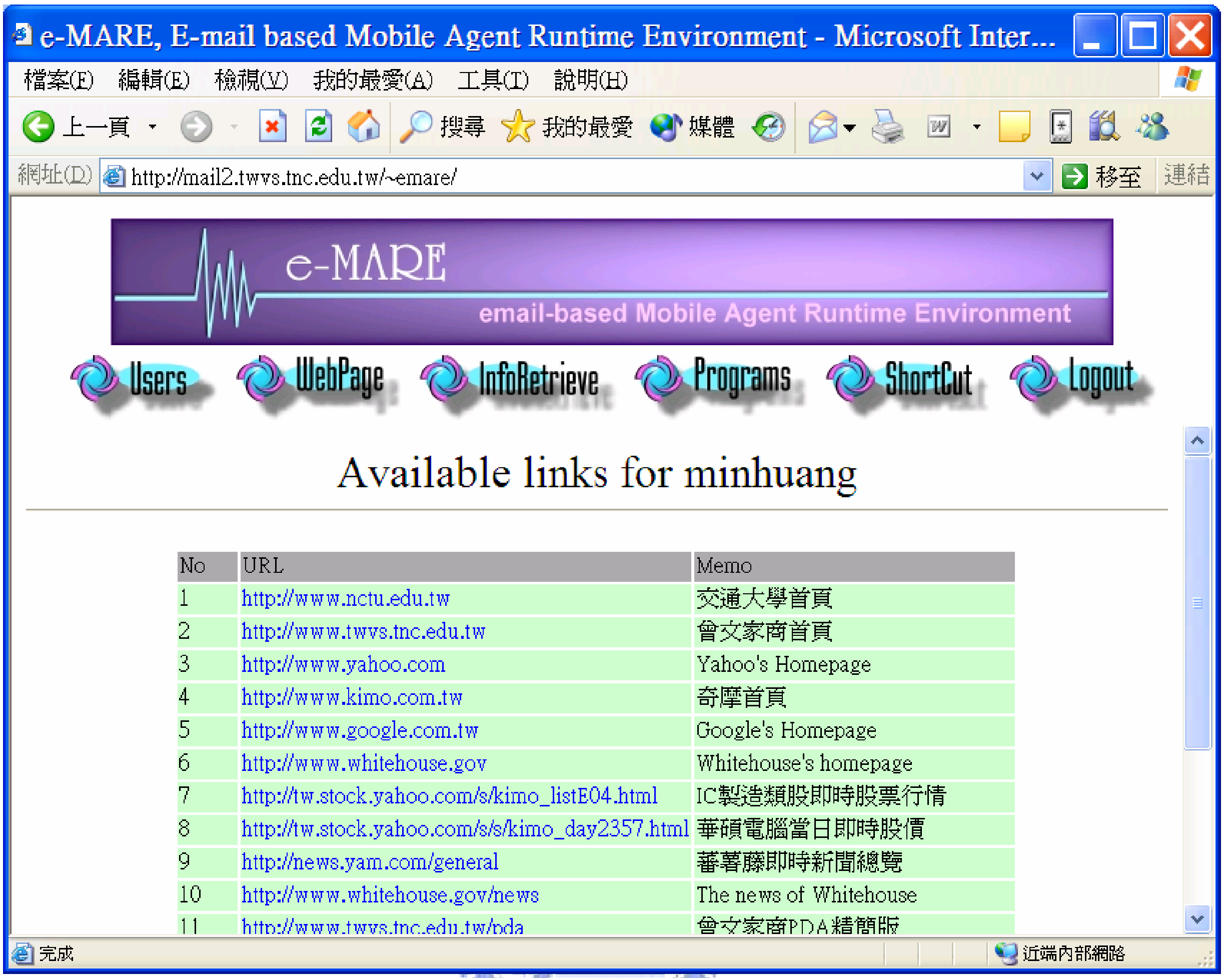

e-MARE, E-mail based Mobile Agent Runtime Environment - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http://mail2.twvs.tnc.edu.tw/~emare/
移至 連結
e-MARE
email-based Mobile Agent Runtime Environment
Users WebPage InfoRetrieve Programs ShortCut Logout
完成
近端內部網路

Name	Notify	Right	email	Phone #	Tag	Page Size	Management
minhuang	0	5	minhuang@mail.twvs.tnc.edu.tw	0921222768	normal	100	[Show links],[Search Engines]
jessica	-	-	-	-	-	-	[Show links],[Search Engines]
skynet	-	-	-	-	-	-	[Show links],[Search Engines]
blue	-	-	-	-	-	-	[Show links],[Search Engines]

Figure 6.3: The page after the user logged into the system.

In Figure 6.4, the system displays all registered URL of the user named ‘minhuang’. All links are numbered in a serial identical number. The number is the accordance for the user to specify for fetching the web page through her mobile devices. If the user specifies the ‘&&SAW’ commands, the list will be transmitted to user’s mailbox in pure ASCII text format. Then, the user may use the ‘&&AFW(*no*)’ command to fetch the contents of her desired web page in ASCII text format. The argument *no* is the item number which is shown in Figure 6.4. In most situations, this is the easiest way for the mobile user to fetch the web page though her mobile device. The e-MARE system will filter the contents of those listed web page, cut the unwanted images, information, advertisements, and other redundant hyper links. If the user has enough history information on the *User Profile* of the e-MARE’s database, the system also will rearrange the results according to the user’s history profile.

In theory, every kind of web pages can be putted into the database of the information retrieval of the e-MARE system. All data items in the database can be accessed through the user’s mobile devices. However, for those pages which do not frequently update the contents are not suitable for the mobile user to access. The frequently updated web page such as real-time news, stock quoting, or box score of play games are the best candidates. The later section will show some examples about how to access the information from those web sites.


e-MARE, E-mail based Mobile Agent Runtime Environment - Microsoft Inter...
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
搜尋
我的最愛
媒體
網址(D)
http://mail2.twvs.tnc.edu.tw/~emare/
移至
連結
e-MARE
email-based Mobile Agent Runtime Environment
Users
WebPage
InfoRetrieve
Programs
ShortCut
Logout
完成
近端內部網路

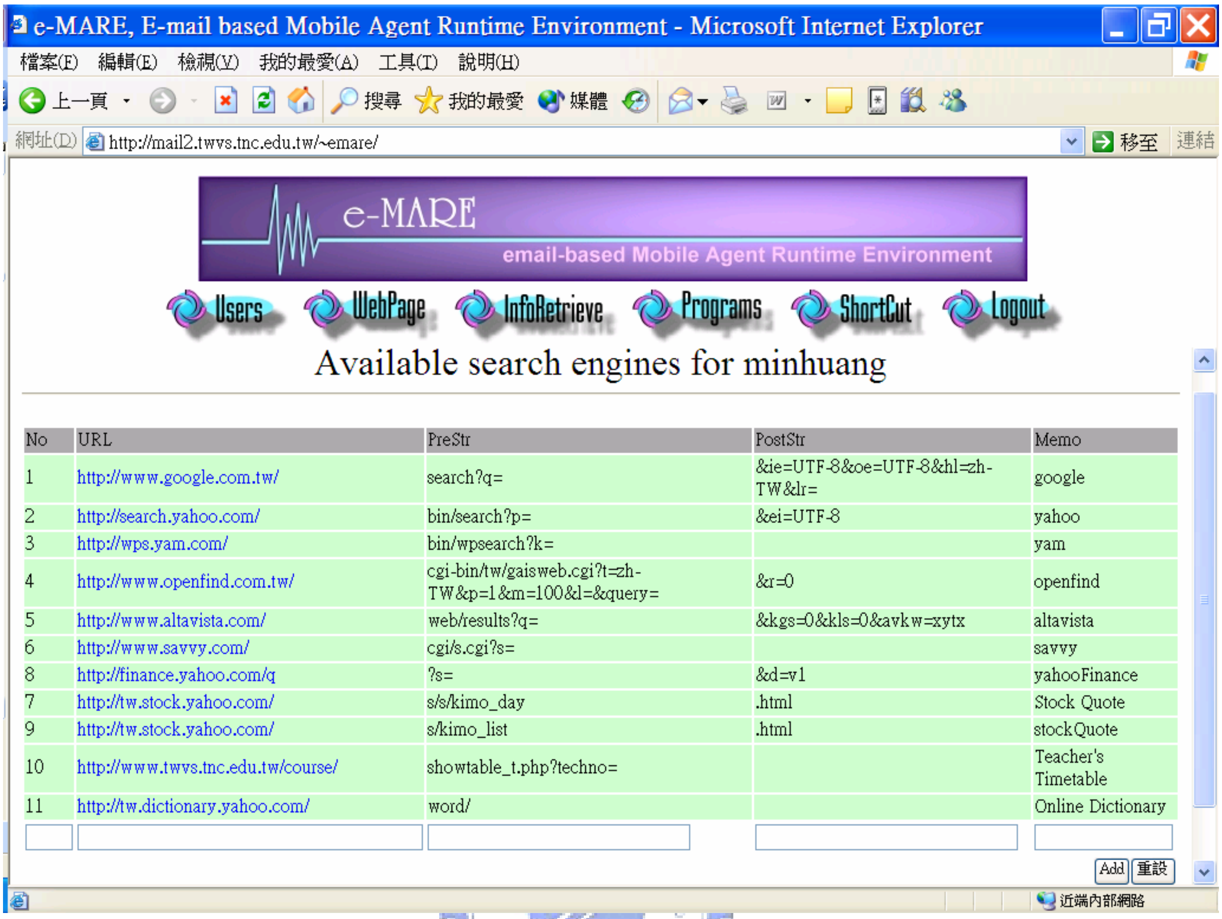
Available links for minhuang

No	URL	Memo
1	http://www.nctu.edu.tw	交通大學首頁
2	http://www.twvs.tnc.edu.tw	曾文家商首頁
3	http://www.yahoo.com	Yahoo's Homepage
4	http://www.kimo.com.tw	奇摩首頁
5	http://www.google.com.tw	Google's Homepage
6	http://www.whitehouse.gov	Whitehouse's homepage
7	http://tw.stock.yahoo.com/s/kimo_listE04.html	IC製造類股即時股票行情
8	http://tw.stock.yahoo.com/s/s/kimo_day2357.html	華碩電腦當日即時股價
9	http://news.yam.com/general	蕃薯藤即時新聞總覽
10	http://www.whitehouse.gov/news	The news of Whitehouse
11	http://www.twvs.tnc.edu.tw/pda	曾文家商PDA精簡版

Figure 6.4: The available links of the user named 'minhuang'.

As shown in Figure 6.4, the user may enter a new URL at the bottom of the available links page, and press the 'Add' button. Then, the entered URL will be inserted into the database of the system, and the new entered data item will be listed on the web pages right now. Of course, the newest inserted data item can be utilized by mobile users via her mobile device as soon as possible.

Entering some keywords and press the 'search' button to query the information on the Internet through the Search Engines is the easiest and the most popular activity on the desktop computer for the information collector. In the website of e-MARE, we also provide the interface for the user to settle her search engines. If the user has pressed the **InfoRetrieve** hyperlink of the main web page, the system will display all available search engines and information sources' URL. The screen snapshot is shown in Figure 6.5.


e-MARE, E-mail based Mobile Agent Runtime Environment - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D)
移至 連結
e-MARE
email-based Mobile Agent Runtime Environment
Users
WebPage
InfoRetrieve
Programs
ShortCut
Logout
Add
重設
近端內部網路

http://mail2.twvs.tnc.edu.tw/~emare/

Available search engines for minhuang

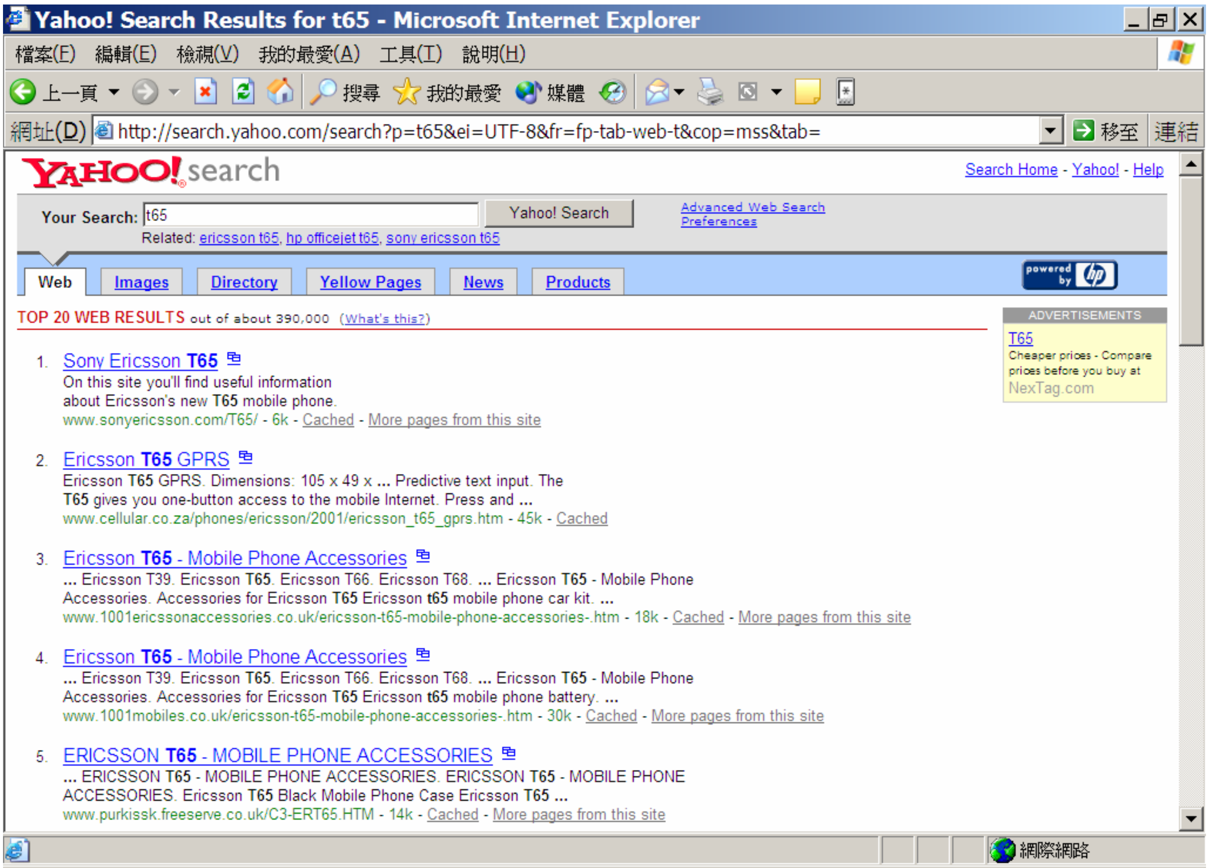
No	URL	PreStr	PostStr	Memo
1	http://www.google.com.tw/	search?q=	&ie=UTF-8&oe=UTF-8&hl=zh-TW&lr=	google
2	http://search.yahoo.com/	bin/search?p=	&ei=UTF-8	yahoo
3	http://wps.yam.com/	bin/wpsearch?k=		yam
4	http://www.openfind.com.tw/	cgi-bin/tw/gaisweb.cgi?t=zh-TW&p=1&m=100&l=&query=	&r=0	openfind
5	http://www.altavista.com/	web/results?q=	&kgs=0&kls=0&avkw=xytx	altavista
6	http://www.savvy.com/	cgi/s.cgi?s=		savvy
8	http://finance.yahoo.com/q	?s=	&d=v1	yahooFinance
7	http://tw.stock.yahoo.com/	s/s/kimo_day	.html	Stock Quote
9	http://tw.stock.yahoo.com/	s/kimo_list	.html	stockQuote
10	http://www.twvs.tnc.edu.tw/course/	showtable_t.php?techno=		Teacher's Timetable
11	http://tw.dictionary.yahoo.com/	word/		Online Dictionary

Figure 6.5: The list of available search engines on the web page.

Different the simple URL address of the **WebPage** category, the information on the **InfoRetrieve** hyperlink is more complex than **WebPage**. To enable the querying function of search engines through mobile device across the long-distance unstable network, the two extra querying field, *PreStr*, and *PostStr* is very important. A typical querying URL encoding command string for searching the keyword ‘T65’ in the search engine, http://www.yahoo.com, is shown as follows:

http://search.yahoo.com/search?p=t65&ei=UTF-8&fr=fp-tab-web-t&cop=mss&tab=

Entering the above querying string will show all related web pages which the contents includes the keyword ‘T65’ in the search engine’s indexed pages. The typical results returned from Yahoo are shown as Figure 6.6.



http://search.yahoo.com/search?p=t65&ei=UTF-8&fr=fp-tab-web-t&cop=mss&tab=

YAHOO! search

Your Search: t65

Related: ericsson t65, hp officejet t65, sony ericsson t65

TOP 20 WEB RESULTS out of about 390,000 (What's this?)

1. Sony Ericsson **T65**
On this site you'll find useful information about Ericsson's new **T65** mobile phone.
www.sonyericsson.com/T65/ - 6k - Cached - More pages from this site

2. Ericsson **T65** GPRS
Ericsson **T65** GPRS. Dimensions: 105 x 49 x **...** Predictive text input. The **T65** gives you one-button access to the mobile Internet. Press and **...**
www.cellular.co.za/phones/ericsson/2001/ericsson_t65_gprs.htm - 45k - Cached

3. Ericsson **T65** - Mobile Phone Accessories
... Ericsson T39. Ericsson **T65**. Ericsson T66. Ericsson T68. **...** Ericsson **T65** - Mobile Phone Accessories. Accessories for Ericsson **T65** Ericsson **t65** mobile phone car kit. **...**
www.1001ericssonaccessories.co.uk/ericsson-t65-mobile-phone-accessories-.htm - 18k - Cached - More pages from this site

4. Ericsson **T65** - Mobile Phone Accessories
... Ericsson T39. Ericsson **T65**. Ericsson T66. Ericsson T68. **...** Ericsson **T65** - Mobile Phone Accessories. Accessories for Ericsson **T65** Ericsson **t65** mobile phone battery. **...**
www.1001mobiles.co.uk/ericsson-t65-mobile-phone-accessories-.htm - 30k - Cached - More pages from this site

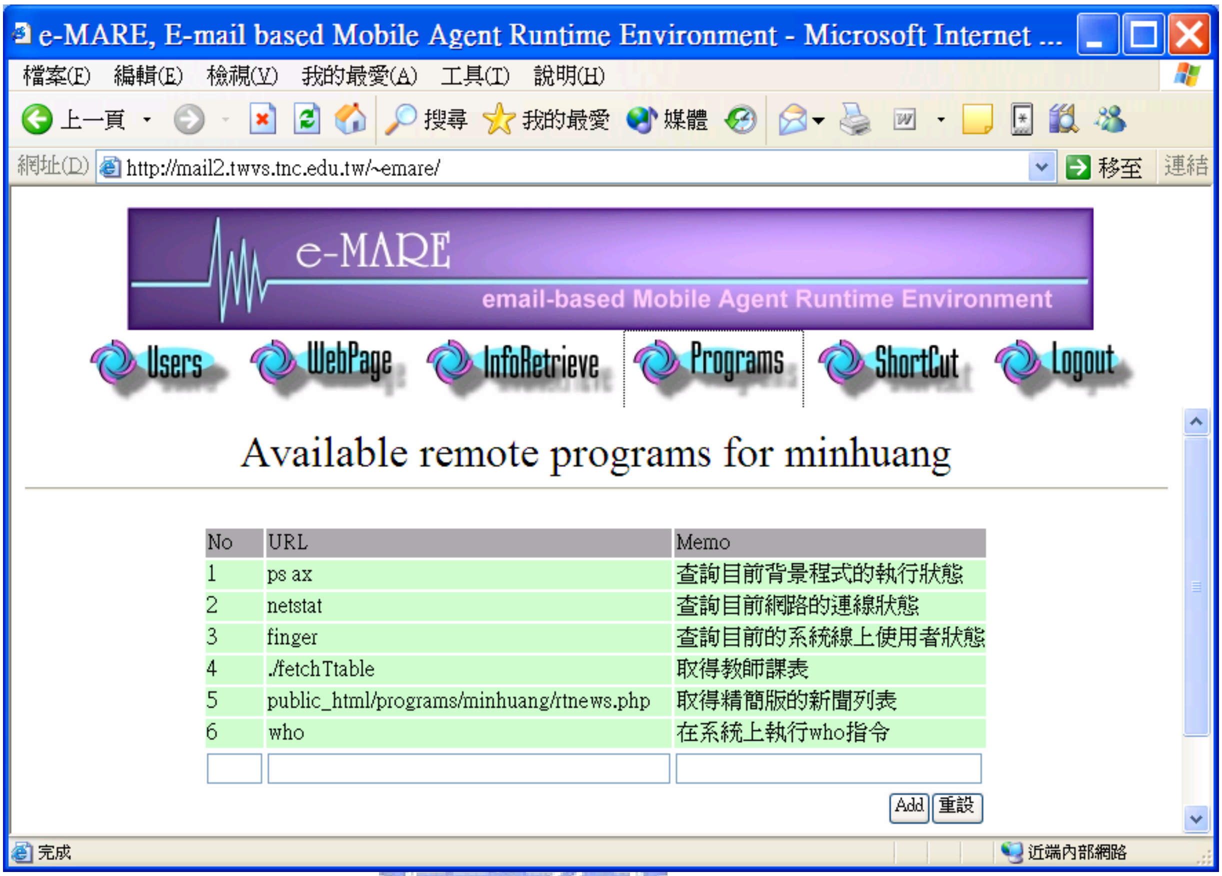
5. ERICSSON **T65** - MOBILE PHONE ACCESSORIES
... ERICSSON **T65** - MOBILE PHONE ACCESSORIES. ERICSSON **T65** - MOBILE PHONE ACCESSORIES. Ericsson **T65** Black Mobile Phone Case Ericsson **T65** **...**
www.purkissk.freeserve.co.uk/C3-ERT65.HTM - 14k - Cached - More pages from this site

T65
Cheaper prices - Compare prices before you buy at
NexTag.com

Figure 6.6: The querying results of entering the keyword 'T65' in Yahoo.

In this example, the URL address of the search engine is http://search.yahoo.com/, the *PreStr* of the commands is 'search?p=', and the *PostStr* is '&ei=UTF-8&fr=fp-tab-web-t&cop=mss&tab='. Actually, the *PreStr* and the *PostStr* are splitted by the keyword (i.e. 't65'). All of the three arguments (*The URL of search engine*, *PreStr*, and *PostStr*) are the basic elements for the user to query the information through the Internet. Thus, if the user properly specifies those arguments for specified search engine, she will queries her desired information through that search engine by e-MARE via her mobile device in a uniform way. In the figure which is shown in Figure 6.5, that page also provides the filed for the user to add her new search engine. To add a new search engine is very easy, and it just needs to fill out those three fields on that page, they are *The URL of search engine*, *PreStr*, and *PostStr*. Again, after the user has pressed the 'Add' button of that page, this new search engine interface will be putted into the database of e-MARE, and it can serve the user as soon as possible.

The same situation for the user to install the user's customize programs is shown at Figure 6.7.


e-MARE, E-mail based Mobile Agent Runtime Environment - Microsoft Internet ...
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
搜尋
我的最愛
媒體
網址(D)
http://mail2.twvs.tnc.edu.tw/~emare/
移至
連結
e-MARE
email-based Mobile Agent Runtime Environment
Users
WebPage
InfoRetrieve
Programs
ShortCut
Logout
Add
重設
完成
近端內部網路

Available remote programs for minhuang

No	URL	Memo
1	ps ax	查詢目前背景程式的執行狀態
2	netstat	查詢目前網路的連線狀態
3	finger	查詢目前的系統線上使用者狀態
4	./fetchTtable	取得教師課表
5	public_html/programs/minhuang/rtnews.php	取得精簡版的新聞列表
6	who	在系統上執行who指令

Figure 6.7: The available remote programs displaying page.

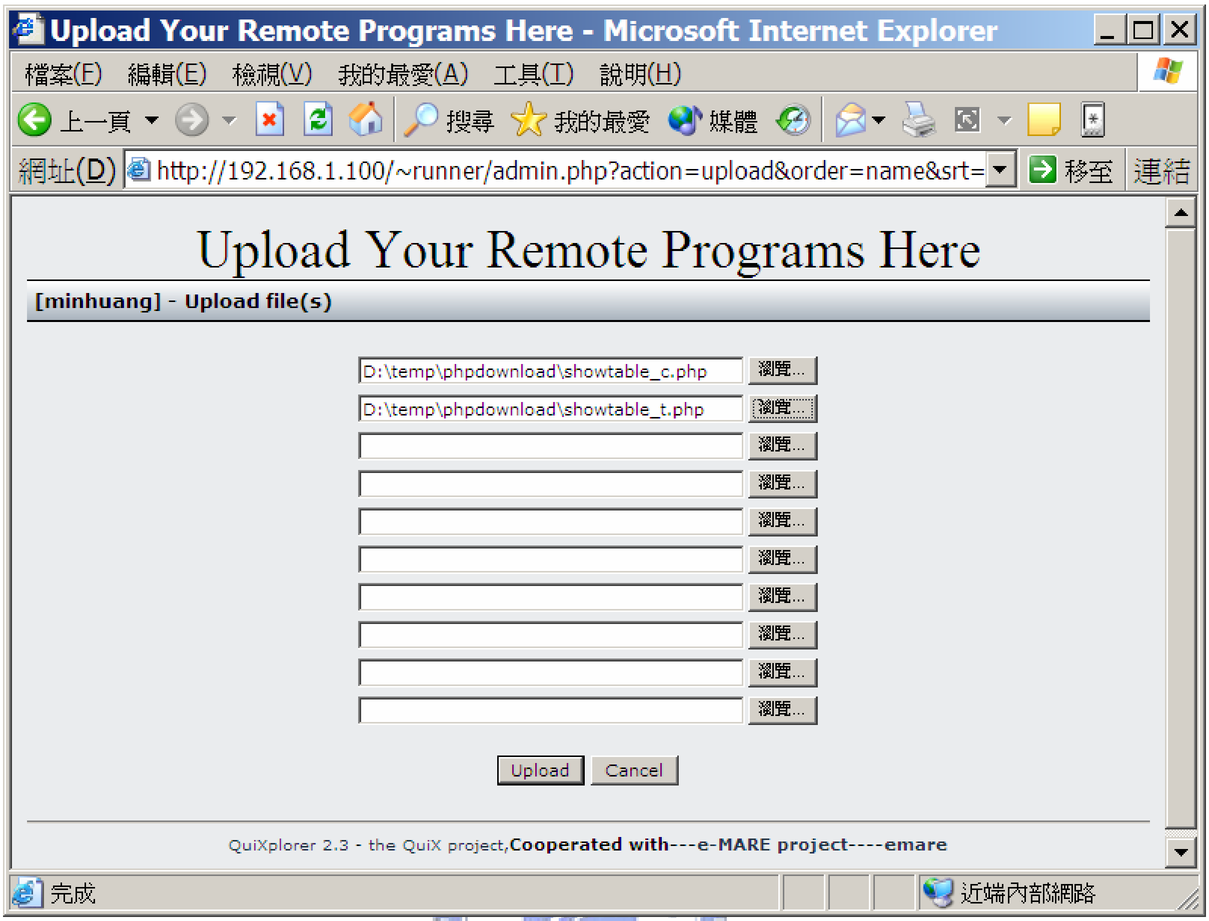
After the user has clicked the hyper link, **Programs**, the web site will show all installed programs which are associated with the user. As the Figure 6.7 shows, the programs can be a system command of the server (in our prototype, it is a Linux server), or the user defined program. No matter what the type of the program is, the installed program should satisfy the following requirements:

- Of course, it has to be properly executed in the server. The installed program can be written by any programming language. The most important key point is that if this command is not the built-in program of the server, the executable binary format has to be compatible with the system of the remote *Mobile Agent Execution Server*.

- It should be found by the server. That is, if the command is not the built-in commands of the server, the appropriate path has to be included in the prefix of the commands of the data item in the database. Of course, if the command needs some arguments to execute properly, these arguments have also to be attached at the following position of that target command. Hence, the commands with different arguments will occupy more than one different data items. They are all independently executed with the others.

- It should utilize the standard I/O as the data input/output mechanism. Thus, the system can collect the executing results through the standard I/O mechanism.
- If it needs some large amount data as the input, or it produces large amount data, the files should be placed in the home directory of the user's account, and specify the filename in an appropriate way by the built-in instructions of the e-MARE.

The installation process of the programs in the system can be divided into two major parts: *uploading the program into the Mobile Agent Execution Server*, *inserting the data item into the database*. As mentioned earlier, the target programs, applications, and mobile agents are all installed in the *Mobile Agent Execution Server*. In our prototype, it is a RedHat Linux server. That is, if the user wants to install her program into the e-MARE for later usage, first, it should enter the command with full path specification in the New Program filed of the web page which is shown in Figure 6.7. After the user has clicked the 'Add' button, the full path of the commands will be one of the data items of the available remote program database of e-MARE. If the command is the user-defined program, the user also has to upload that program into the appropriate directory of *Mobile Agent Execution Server*. In our implementation, by default, all programs will be placed in the *runner*'s home directory of the *Mobile Agent Execution Server*. That is, a user account named *runner* of that server is responsible for storing the user's custom programs, and executing one of those programs according to the user's desire. Our system also provides a web interface for the user to upload her programs into *runner*'s home directory of *Mobile Agent Execution Server*. The screen snapshot of the uploading programs web page is shown in Figure 6.8.


Upload Your Remote Programs Here - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
搜尋
我的最愛
媒體
網址(D)
移至
連結
瀏覽...
Upload
Cancel
完成
近端內部網路

http://192.168.1.100/~runner/admin.php?action=upload&order=name&srt=

Upload Your Remote Programs Here

[minhuang] - Upload file(s)

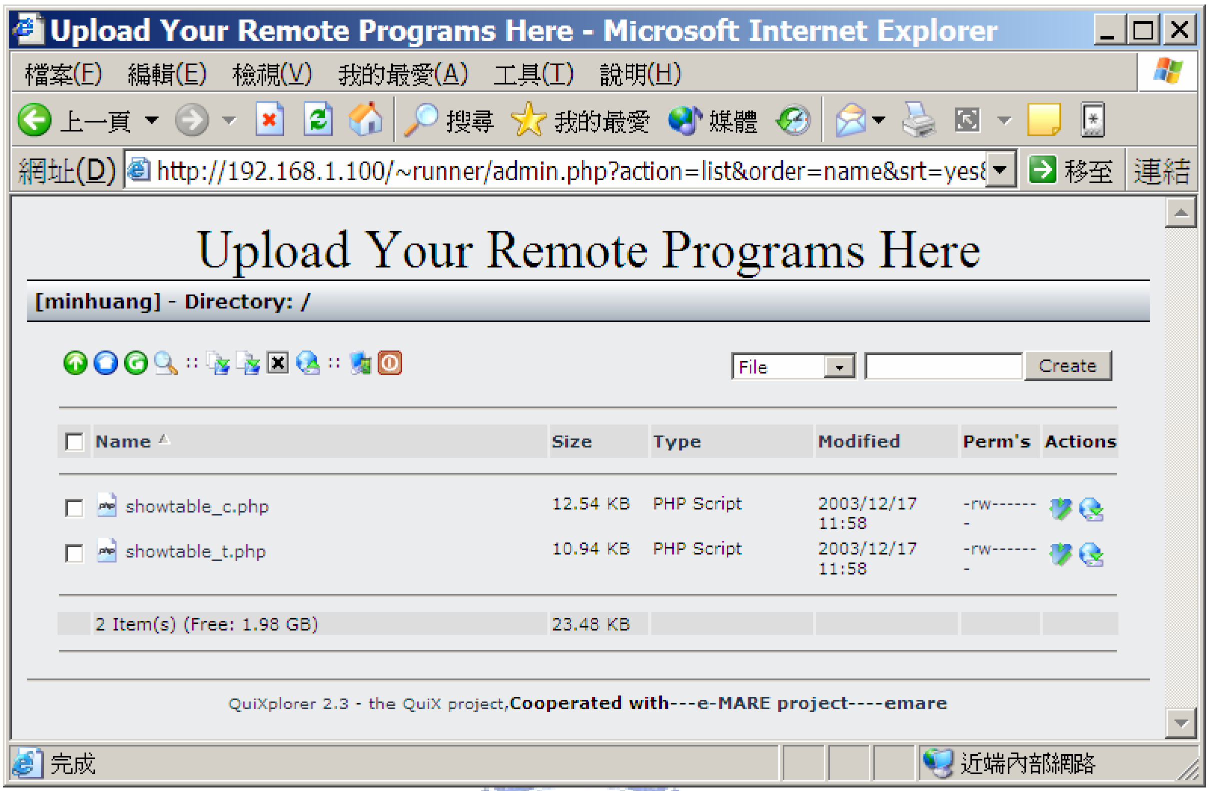
D:\temp\phpdownload\showtable_c.php

D:\temp\phpdownload\showtable_t.php

QuiXplorer 2.3 - the QuiX project,Cooperated with---e-MARE project----emare

Figure 6.8: Uploading the remote programs into runner's home directory.

As shown in Figure 6.8, the authorized user can specify the local programs in the web page, and then click ‘Upload’ button. The QuiXplorer program will upload those selected files to appropriate directory. In our implementation, each e-MARE user has her own sub-directory under the runner’s home directory. Moreover, the each e-MARE user will be added into the QuiXplorer’s administrator’s list. That is, she will own some rights to access (including upload, download, modify, and remove) her files on her own directory. Thanks to the help of QuiXplorer, after the user’s uploading process, the target programs will be listed in the screen, and the screen snapshot is shown in Figure 6.9. If the correspondent data items of those programs are correct, these programs will be found by the server, and be executed in the server as the user’s direction from the commands of her operational e-mail.



Upload Your Remote Programs Here - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http://192.168.1.100/~runner/admin.php?action=list&order=name&srt=yes
移至 連結
File
Create
完成
近端內部網路

Upload Your Remote Programs Here

[minhuang] - Directory: /

□	Name	Size	Type	Modified	Perm's	Actions
□	showtable_c.php	12.54 KB	PHP Script	2003/12/17 11:58	-rw-------	
□	showtable_t.php	10.94 KB	PHP Script	2003/12/17 11:58	-rw-------	
	2 Item(s) (Free: 1.98 GB)	23.48 KB				

QuiXplorer 2.3 - the QuiX project,Cooperated with---e-MARE project----emare

Figure 6.9: The screen snapshot of listed available programs for user 'minhuang'.

The process of mobile agents and applications installation are the same as program installation.

6.5 An introduction to WAP interface of e-MARE

Some low-end mobile devices, such as mobile cellular phones, don't provide the built-in e-mail client for the users to directly access his e-mail on the Internet. Moreover, to enter the complicate commands via such low-end mobile cellular phones is a hard work for many people. Thus, to ease the works for entering the commands via such mobile phones, e-MARE also provides a famous WAP interface. The URL address of the our proposed WAP interface is http://www.twvs.tnc.edu.tw/~wap. It means that if the WAP browser of the user's mobile phone has connected to that address, the main page of the WAP interface of e-MARE will be shown as Figure 6.10.

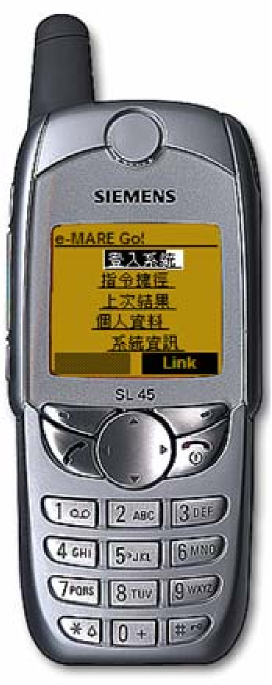
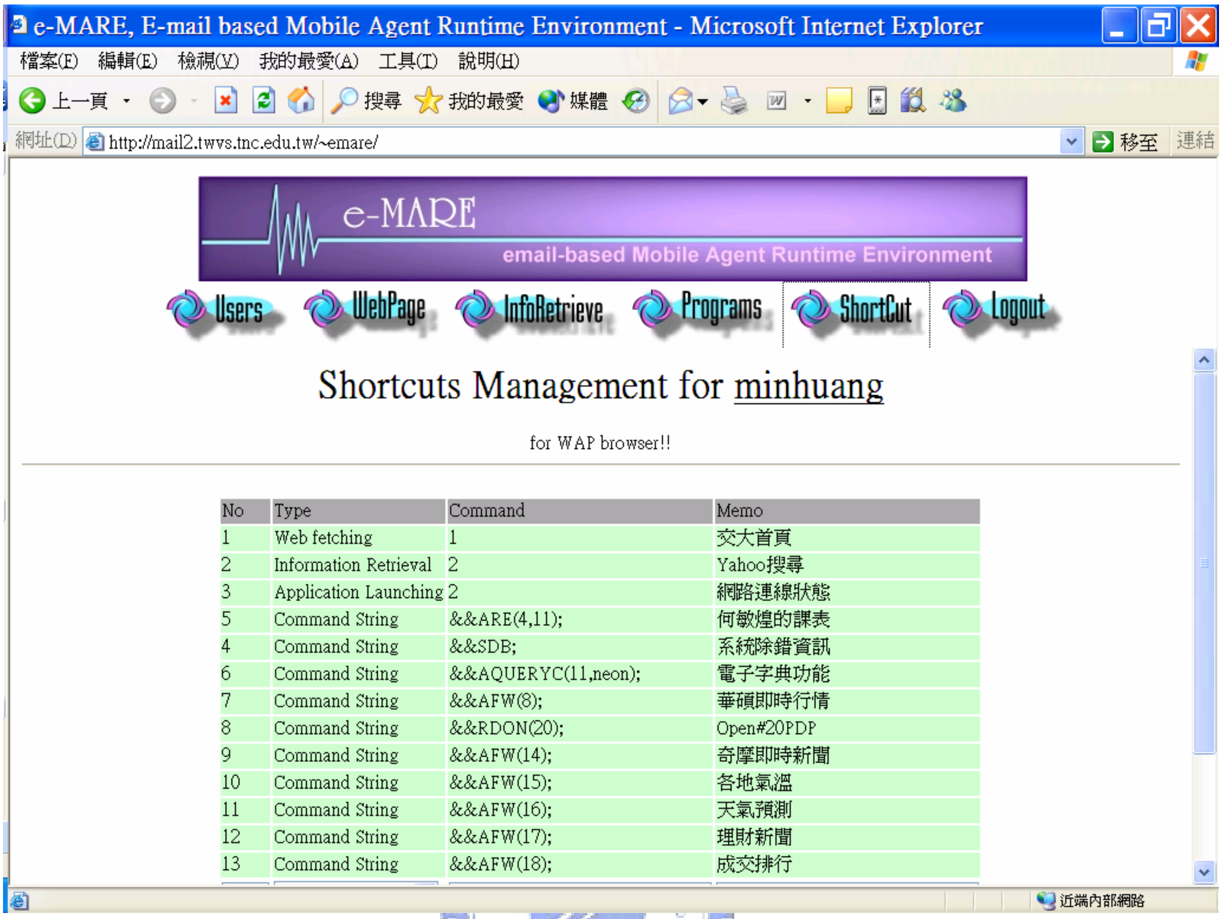


Figure 6.10: The main page of the WAP interface of e-MARE.

As shown in Figure 6.10, it is a WAP site named ‘e-MARE Go!’. At the first, the user may login to the system, and then, she will have the right to access the shortcut of the commands. Of course, the shortcut of commands can also be modified by WWW interface, which is shown in Figure 6.11.


e-MARE, E-mail based Mobile Agent Runtime Environment - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http://mail2.twvs.tnc.edu.tw/~emare/
移至 連結
e-MARE
email-based Mobile Agent Runtime Environment
Users
WebPage
InfoRetrieve
Programs
ShortCut
Logout
近端內部網路

Shortcuts Management for minhuang

for WAP browser!!

No	Type	Command	Memo
1	Web fetching	1	交大首頁
2	Information Retrieval	2	Yahoo搜尋
3	Application Launching	2	網路連線狀態
5	Command String	&&ARE(4,11);	何敏煌的課表
4	Command String	&&SDB;	系統除錯資訊
6	Command String	&&AQUERYC(11,neon);	電子字典功能
7	Command String	&&AFW(8);	華碩即時行情
8	Command String	&&RDON(20);	Open#20PDP
9	Command String	&&AFW(14);	奇摩即時新聞
10	Command String	&&AFW(15);	各地氣溫
11	Command String	&&AFW(16);	天氣預測
12	Command String	&&AFW(17);	理財新聞
13	Command String	&&AFW(18);	成交排行

Figure 6.11: The WWW interface for 'e-MARE Go!'.

As the Figure 6.11 shows, after the user has chosen the 'Shortcuts' on his mobile phone, the display of that phone will show the following menu, and the item lists of the menu will the same as the items which are shown in Figure 6.11. The shortcut 'of e-MARE Go!' is shown in Figure 6.12.

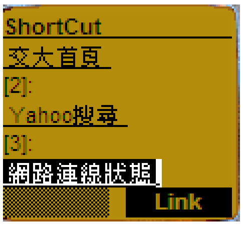
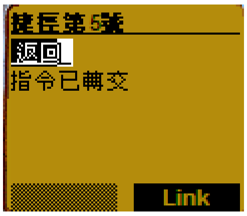


Figure 6.12: The shortcut of e-MARE Go.

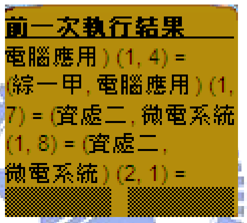
If the user chooses any one shortcut of that menu which is shown in Figure 6.12, 'e-MARE Go!' will execute that shortcut and return a notifying message on the display. The typical message is shown in Figure 6.13.



指令已轉交

Figure 6.13: The messages after user's selection for executing the 5th shortcut.

After the user has reviewed the message which is shown in Figure 6.13, it always means there is also a copy of results which were stored in the user's mailbox. In this situation, the user has two choices, the one is to read the results on the WAP environment of 'e-MARE Go!', or he also can read the results by any e-mail client via any kinds of mobile devices. The 'e-MARE Go!' also provides the choices for the user to read the one-time fetched results. In this example, after the user has chosen the 'last results' menu item, the results will be displayed on the screen of mobile phones, and it is shown in Figure 6.14.



電腦應用) (1, 4) =
(綜一甲, 電腦應用) (1,
7) = (資處二, 微電系統
(1, 8) = (資處二,
微電系統) (2, 1) =

Figure 6.14: The results after user's selection for executing the 5th shortcut.

To more enhance the usability of 'e-MARE Go!', it also provides two menu items for the user to enable or disable the SMS notification, the menu item is shown in Figure 6.15.

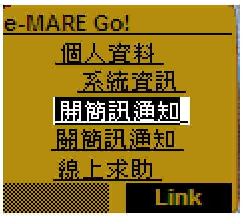


Figure 6.15: The menu item to enable or disable the SMS function.

6.6 Examples of e-MARE to access the information on the Internet

6.6.1 Example I: Querying the available search engines of e-MARE

At the first, a mobile user who wants to access the Internet through the e-MARE framework has to register to e-MARE in advance. In e-MARE, the id is the registered e-mail account. The user also has to set a nick name for this account for easily accessing the database. The e-MARE server provides a web site for the user to browse and edit the user's related information, and the figure which has already been shown in Figure 6.1.

The first web page of e-MARE will show all available registered users. Consider the protection of the user, detailed information of the user will be hided until the user has logged into the system. The registered user can login into the system, maintains her own information, and installs the user-defined applications into the *Remote Execution Server* through the web interface in Figure 6.1.

For a safety system, an appropriate authentication mechanism is very important. In our system, we let the user set her password in advance for later operation. If a user has not yet registered and set her password, she will have no right to access the system. In this example, we assume the user has already set her personal information in the database server of e-MARE framework.

Because querying for the system available search engines is the built-in function of e-MARE, there is no other previous task have to be done. The only thing has to be done for a registered mobile user is to compose an operational e-mail with the specified encoded subject, and then send it to emare@mail.twvs.tnc.edu.tw. The subject of the composed operational e-mail has to provide the user's password and the appropriate commands. The subject for querying the available search engines is shown as the follows:

&&PWD(secret);&&SASE;

The *PWD(secret)* command is the way for mobile user to provide the password of the user. If the user doesn't provide the password for authentication, the user will have no right to access the services of e-MARE. Because the example is very simple, hence it doesn't need to write the body of the operational e-mail in this example. In most cases, after the use has been sent the operational e-mail to the public service account, the results will be ready in the user's mailbox in a very short time. The sample results are shown as follows:

```
1,google,http://www.google.com.tw/
2,yahoo,http://search.yahoo.com/
3,yam,http://wps.yam.com/
4,openfind,http://www.openfind.com.tw/
5,altavista,http://www.altavista.com/
6,savvy,http://www.savvy.com/
7,Stock Quote,http://tw.stock.yahoo.com/
```

```
8,yahooFinance,http://finance.yahoo.com/q
9,stockQuote,http://tw.stock.yahoo.com/
10,Teacher's Timetable,http://www.twvs.tnc.edu.tw/course/
11,Online Dictionary,http://tw.dictionary.yahoo.com/

Service Time:0.182261943817139
```

The above search engines and information source lists can be added, removed, and modified via WWW interface by user in advance. The WWW interface has been introduced in Section 6.4. The e-MARE also provides some commands for mobile user to modify the lists via operational e-mail. Actually, not every item in the list is the whole complete search engine’s URL. Any information source which accepts any format of querying string can be added into the list. Of course, the format of querying string has to be recognized, and the *PreStr* and *PostStr* should be properly installed in the specified data item of the database of e-MARE server via the WWW interface, since the e-MARE system will wrap the keyword into an appropriate querying string according to the *PreStr* and *PostStr* setting, and send the querying string to that information source.

6.6.2 Example II: Querying keywords via several search engines

According to the results returned from the commands &&SASE as mentioned in Section 6.5.2, the user can query her desired information from the specified search engine by specify both the keyword and the listed number of search engine which is listed in *Example I*. For example, if the mobile user wants to query ‘Ericsson T65’ from Yahoo (http://www.yahoo.com), she may encode her subject of operational e-mail as follows:

```
&&PWD(secret);&&AQUERYC(2,Ericsson T65);
```

After the user has sent the above commands to emare@mail.twvs.tnc.edu.tw, few seconds later, she will obtain the following results:

```
Yahoo!
Search Home - Yahoo! - Help
Your Search:
Advanced Web Search
Preferences
Web            Images            Directory            Yellow 읍 ages            News            Products
WEB RESULTS out of about 265,000    (What's this?)
  1. Sony Ericsson T65 Open this result in new window
     On this site you'll find useful information
     about Ericsson's new T65 mobile phone.
     www.sonyericsson.com/T65/    - 6k    - Cached    - More pages from this
     site

  2. ERICSSON T65 - MOBILE PHONE ACCESSORIES
     Open this result in new window
     ... ERICSSON T65 - MOBILE PHONE ACCESSORIES. ERICSSON T65 - MOBILE
     PHONE
     ACCESSORIES. Ericsson T65 Black Mobile Phone Case Ericsson T65 ...
     www.purkissk.freeserve.co.uk/C3-ERT65.HTM    - 14k    - Cached    - More
```

pages from this site

3. Ericsson T65 GPRS Open this result in new window
Ericsson T65 GPRS. ... The T65 also supports features such as
background picture,
Mobile chat, calendar and games. More Ericsson Phones. Search For
More Info. ...
www.cellular.co.za/phones/ericsson/2001/ericsson_t65_gprs.htm - 45k
- Cached - More pages from this site

4. Free Ericsson T65 color wallpaper colour backgrounds logos
Open this result in new window
Free Ericsson T65 wallpapers backgrounds Select model: Alcatel One
Touch 535. ...
www.esato.com/logos/colour/index.php?t=2&pid=15 - 25k - Cached
- More pages from this site

5. Ericsson T65 Cosmic Blue - The Register
Open this result in new window
... Ericsson T65 Cosmic Blue Home Page > GSM Mobile Phones >
Ericsson
T65 Cosmic Blue. Ericsson T65 Cosmic Blue click for larger image,
...
www.expansys.com/regproduct.asp?code=ERIC_MOBT65 - 31k - Cached
- More pages from this site

6. IT AsiaOne - Phones - Creative chatterbox: The Ericsson T65
Open this result in new window
... Creative chatterbox: The Ericsson T65. ... Product: Ericsson
T65; Price: S$388 (2-year
plan) and S$438 (1-year plan); Availability: All Ericsson
authorised resellers ...
it.asia1.com.sg/reviews/phones/por001_20011108.html - 24k - Cached
- More pages from this site

7. Handy, PC & Co | Telekommunikation | MSN | FOCUS Online | Test:
... Open this result in new window - Translate this page
Handy, PC, Co, Top-Themen, News, Personal Computing,
Telekommunikation, Unterhaltungselektronik,
Netguide, Forum, FOCUS Online, Test: Ericsson T65. ...
focus.msn.de/D/DC/DCE/DCE97/dce97.htm - 48k - Cached - More
pages from this site

8. Acheter Coque portable Sony-Ericsson Coque pour Sony-Ericsson T65
... Open this result in new window - Translate this page
... Ericsson T65. Sony-Ericsson Coque pour Sony-Ericsson T65
Constructeur
: Sony-Ericsson - Genre : Coque portable Ecrivez votre critique
...
www.monsieurprix.com/listing/gen/J000026017.html - 18k - Cached
- More pages from this site

9. Telefon celular Ericsson Open this result in new window
... Telefon celular Ericsson T65 Ericsson T65Telefonul celular
Ericsson
T65 functioneaza in 2 benzi de frecventa: 1800 respectiv 900 Mhz.
...
www.idei-de-afaceri.go.ro/telefon_celular_ericsson.htm - 26k - Cached

10. Ericsson T65 - www.Portel.it Open this result in new window -
Translate this page
... home > schede, Ericsson T65, Caratteristiche Principali. ...
Leggi le recensioni
di questo cellulare: Ericsson T65, testato da F. Ponticelli.
Funzioni Vocali. ...
www.portel.it/telefonini/model.asp?model_num=166 - 30k - Cached
- More pages from this site

11. ITavisen.no | Ericsson T65 Open this result in new window
Ericsson T65, Av Espen Irwing Swang, fre 25. jan 2002 kl. 15:43.
T65
er Ericssons f 鶼 ste telefon uten antenne. Telefonen retter seg ...
www.itavisen.no/art/1298070.html - 46k - Cached - More pages
from this site

12. WMLClub - FAQS - Ericsson T65 Open this result in new window -
Translate this page
FAQS - Caracteristicas del Ericsson T65: Visualizacion,
tiempo de conversacion, tiempo en espera... ...
www.wmlclub.com/docs/faqs/ericssont65.htm - 10k - Cached - More
pages from this site

13. Ericsson T65 - ???????-?????? Open this result in new window
The summary for this Russian page contains characters that cannot
be correctly displayed in this language/character set.

```
        www.megafonmoscow.ru/phones/ericsson/    - 20k    - Cached
     14. Tiscali Open this result in new window - Translate this page
        ... delle notizie relative al mondo della telefonia e del Mobile
        Internet.
        Sfondi. Ericsson T39/R520/T65/T68/T60d. Seleziona la categoria.
        ...
        mobile.tiscali.it/ls/europicsCat.jsp?idsezione=69&idPhone=2    - 33k
          - Cached    - More pages from this site
     15. ??? ????? / ???????? ? ?????????? / ...
        Open this result in new window
        The summary for this Russian page contains characters that cannot
        be correctly displayed in this language/character set.
        mts.perm.ru/phones/se_t65.html    - 22k    - Cached
     16. Ericsson t65 : Kelkoo - B 醒 ta pris :
        Open this result in new window
        j 鄹 f 顛  priser och k 轔  via Kelkoo i de b 醒 ta butikerna i Sverige,Ericsson
        t65.
        R 酹 t produkt. ... S 闥 et tog 0.23 sekunder. Startsida > Resultat
        f 顛  Ericsson t65. ...
        se.kelkoo.com/b/a/ss_Ericsson_t65.html    - 48k    - Cached    - More
        pages from this site
     17. Sony ericsson t65 : Kelkoo - bedste priser :
        Open this result in new window
        sammenlign priser, k 鶉  hos Kelkoo i de bedste butikker i
        Danmark,Sony ericsson
        t65. Din s 鶤 emaskine til shopping. ... Start > S 鶤  efter Sony
        ericsson t65. ...
        dk.kelkoo.com/b/a/ss_Sony_ericsson_t65.html    - 57k    - Cached    - More
        pages from this site
     18. ericsson t65 : Kelkoo - beste pris :
        Open this result in new window
        sammenlign priser og kj 鶡  med Kelkoo hos de beste butikkene i
        Norge,ericsson
        t65. S 鳰. Finn. Kj 鶡. ... Hjem > S 鳰  etter ericsson t65. ...
        no.kelkoo.com/b/a/ss_ericsson_t65.html    - 53k    - Cached    - More
        pages from this site
     19. Ericsson T65 Mobile Phones reviews
        Open this result in new window
        Ericsson T65 in Mobile Phones / Mobile Phone Reviews reviews at
        Review Centre.
        Consumer reviews of products and services. 27 November 2003. ...
        ERICSSON T65. ...
        www.reviewcentre.com/reviews1082.html    - 59k    - Cached    - More
        pages from this site
     20. Mobil.cz: Ericsson T65 ? urcite si jej oblibite (recenze)
        Open this result in new window
        ... integrovanou antenou, podpora obrazkovych a hudebnich
        textovek, hry, obrazky na displeji,
        ale i kalendar, ukolovnik, GPRS a e-mail ? to vse je Ericsson T65.
        ...
www.mobil.cz/mobilni_komunikace/mobilni_telefony/abecedni_prehled_mt/sonyericsson/ericssont65recenze
        011213.html    - 101k    - Cached    - More pages from this site
   Results Page:
   1    2 3 4 5 6 7 8 9 10 Next
   Web            Images            Directory            Yellow 읍 ages            News            ?
   Advanced Web Search
   Preferences
   Copyright ?2003 Yahoo! Inc. All rights reserved. Privacy Policy -
   Terms of Service - Ad Feedback
   Search Technology provided by Google
Service Time:0.124495029449463
```

As the above results shows, it is the pure text version for searching that keyword 'Ericsson T65' via Yahoo. The service time is shown at the last line of the result and is about 0.12 seconds. Of course, the results are too long to display at many mobile devices at once. Therefore, we introduce the result-slicing mechanism to handle those very long results. However, for the convenience of explanation in the following examples, we assume all later examples show the results at once. The result-slicing mechanism will be demonstrated in another individual subsection.

Except querying the contents for some keywords through the specified search engines, moreover, the e-MARE also allows the user to fetch all links of that web page. To achieve that purpose, the subject of operational e-mail has to be composed as follows:

&&PWD(secret);&&AQUERYL(2,Ericsson T65);

The results of the above commands are shown as follows:

```
identification number: 2343813
1, http://rd.yahoo.com/M=224039.1984929.4456210.1922510/D=yahoo/S=2766679:HEAD/A=1522569/R=0/*http://search.yahoo.com
2, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WSH/*-http://search.yahoo.com
3, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WYH/*-http://www.yahoo.com
4, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WHL/*-http://help.yahoo.com/help/us/ysearch/
5,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WAST/*-http://search.yahoo.com/web/advanced?p=Ericsson+T65&ei=UTF
-8
6,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WPRT/*-http://search.yahoo.com/search/preferences?pref_done=http%3A//s
earch.yahoo.com/search%3Fp=Ericsson%2520T65&pref_cancel=http%3A//search.yahoo.com/search%3Fp=Ericsson%2BT65%26ei=UTF-8
7,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WNIT/*-http://images.search.yahoo.com/search/images?p=Ericsson+T65&e
i=UTF-8
8,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WNDT/*-http://search.yahoo.com/search/dir?p=Ericsson+T65&ei=UTF-8
9,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WNYT/*-http://yp.search.yahoo.com/search/ypredirect?p=Ericsson+T65&ei
=UTF-8
10,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WNNT/*-http://search.news.yahoo.com/search/news?p=Ericsson+T65&ei=
UTF-8
11,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WNPT/*-http://search.shopping.yahoo.com/search?p=Ericsson+T65&ei=UT
F-8&cop=mss&__yltc=s:2766679,d:14489115,sec:srchtab,slk:products,k:QUERY.p_rd_safe
12, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WHS/*-http://help.yahoo.com/help/us/ysearch/basics/basics-03.html
13, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=1/H=0/*-http://www.sonyericsson.com/T65/
14, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=1/NW=1/H=0/*-http://www.sonyericsson.com/T65/
15,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=1/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&
url=k1pjZxDOy_8J:www.sonyericsson.com/T65/
16,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=1/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&v
st=0&vs=www.sonyericsson.com
17, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=2/H=0/*-http://www.purkissk.freeserve.co.uk/C3-ERT65.HTM
18,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=2/NW=1/H=0/*-http://www.purkissk.freeserve.co.uk/C3-ERT65.H
TM
19,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=2/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&
url=1_CPFogGIm4J:www.purkissk.freeserve.co.uk/C3-ERT65.HTM
20,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=2/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&v
st=0&vs=www.purkissk.freeserve.co.uk
21,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=3/H=0/*-http://www.cellular.co.za/phones/ericsson/2001/ericsson_t
65_gprs.htm
22,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=3/NW=1/H=0/*-http://www.cellular.co.za/phones/ericsson/2001/eri
csson_t65_gprs.htm
23,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=3/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&
url=LsCY5ee1pcYJ:www.cellular.co.za/phones/ericsson/2001/ericsson_t65_gprs.htm
24,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=3/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&v
st=0&vs=www.cellular.co.za
```

25,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=4/H=0/*-http://www.esato.com/logos/colour/index.php?t=2&pid=15

26,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=4/NW=1/H=0/*-http://www.esato.com/logos/colour/index.php?t=2&pid=15

27,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=4/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=RuH4HG6A15gJ:www.esato.com/logos/colour/index.php%3Ft=2%26pid=15

28,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=4/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&vst=0&vs=www.esato.com

29,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=5/H=0/*-http://www.expansys.com/regproduct.asp?code=ERIC_MOBT65

30,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=5/NW=1/H=0/*-http://www.expansys.com/regproduct.asp?code=ERIC_MOBT65

31,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=5/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=2KzKLOP9N90J:www.expansys.com/regproduct.asp%3Fcode=ERIC_MOBT65

32,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=5/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&vst=0&vs=www.expansys.com

33,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=6/H=0/*-http://it.asia1.com.sg/reviews/phones/por001_20011108.html

34,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=6/NW=1/H=0/*-http://it.asia1.com.sg/reviews/phones/por001_20011108.html

35,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=6/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=prUPzpas94EJ:it.asia1.com.sg/reviews/phones/por001_20011108.html

36,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=6/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&vst=0&vs=it.asia1.com.sg

37, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=7/H=0/*-http://focus.msn.de/D/DC/DCE/DCE97/dce97.htm

38,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=7/NW=1/H=0/*-http://focus.msn.de/D/DC/DCE/DCE97/dce97.htm

39,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS6/R=7/*-http://tools.search.yahoo.com/language/translation/translatedPage.php?tt=url&text=http%3A//focus.msn.de/D/DC/DCE/DCE97/dce97.htm&lp=de_en

40,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=7/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=gEc08aADTmIJ:focus.msn.de/D/DC/DCE/DCE97/dce97.htm

41,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=7/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&vst=0&vs=focus.msn.de

42,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=8/H=0/*-http://www.monsieurprix.com/listing/gen/J000026017.html

43,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=8/NW=1/H=0/*-http://www.monsieurprix.com/listing/gen/J000026017.html

44,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS6/R=8/*-http://tools.search.yahoo.com/language/translation/translatedPage.php?tt=url&text=http%3A//www.monsieurprix.com/listing/gen/J000026017.html&lp=fr_en

45,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=8/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=RmOkP4eTFaUJ:www.monsieurprix.com/listing/gen/J000026017.html

46,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=8/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&vst=0&vs=www.monsieurprix.com

47, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=9/H=0/*-http://www.itavisen.no/art/1298070.html

48, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=9/NW=1/H=0/*-http://www.itavisen.no/art/1298070.html

49,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=9/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=Q9_vRr3pC9UJ:www.itavisen.no/art/1298070.html

50,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=9/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&vst=0&vs=www.itavisen.no

51, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=10/H=0/*-http://www.wmlclub.com/docs/faqs/ericssont65.htm

52,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=10/NW=1/H=0/*-http://www.wmlclub.com/docs/faqs/ericssont65.ht

m
53,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS6/R=10/*-http://tools.search.yahoo.com/language/translation/translatedPage.php?tt=url&text=http%3A//www.wmlclub.com/docs/faqs/ericssont65.htm&lp=es_en
54,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=10/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=9IIPAJPbXEAJ:www.wmlclub.com/docs/faqs/ericssont65.htm
55,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=10/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&vst=0&vs=www.wmlclub.com
56,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=11/H=0/*-http://www.portel.it/telefonini/model.asp?model_num=166
57,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=11/NW=1/H=0/*-http://www.portel.it/telefonini/model.asp?model_num=166
58,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS6/R=11/*-http://tools.search.yahoo.com/language/translation/translatedPage.php?tt=url&text=http%3A//www.portel.it/telefonini/model.asp%3Fmodel_num=166&lp=it_en
59,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=11/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=hGY9YL_7GsIJ:www.portel.it/telefonini/model.asp%3Fmodel_num=166
60,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=11/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&vst=0&vs=www.portel.it
61, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=12/H=0/*-http://www.megafonmoscow.ru/phones/ericsson/
62,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=12/NW=1/H=0/*-http://www.megafonmoscow.ru/phones/ericsson/
63,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=12/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=D0RfK98bNvMJ:www.megafonmoscow.ru/phones/ericsson/
64,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=13/H=0/*-http://www.ericsson.fi/infocenter/news/articles/t65.shtml
65,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=13/NW=1/H=0/*-http://www.ericsson.fi/infocenter/news/articles/t65.shtml
66,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=13/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=-btpe-NJ8uAJ:www.ericsson.fi/infocenter/news/articles/t65.shtml
67,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=13/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&vst=0&vs=www.ericsson.fi
68,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=14/H=0/*-http://mobile.tiscali.it/ls/europicsCat.jsp?idsezione=69&idPhone=2
69,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=14/NW=1/H=0/*-http://mobile.tiscali.it/ls/europicsCat.jsp?idsezione=69&idPhone=2
70,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS6/R=14/*-http://tools.search.yahoo.com/language/translation/translatedPage.php?tt=url&text=http%3A//mobile.tiscali.it/ls/europicsCat.jsp%3Fidsezione=69%26idPhone=2&lp=it_en
71,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=14/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=GGKilIeOrmoJ:mobile.tiscali.it/ls/europicsCat.jsp%3Fidsezione=69%26idPhone=2
72,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=14/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&vst=0&vs=mobile.tiscali.it
73, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=15/H=0/*-http://mts.perm.ru/phones/se_t65.html
74, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=15/NW=1/H=0/*-http://mts.perm.ru/phones/se_t65.html
75,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=15/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=BR6_OrnRXawJ:mts.perm.ru/phones/se_t65.html
76, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=16/H=0/*-http://se.kelkoo.com/b/a/ss_Ericsson_t65.html
77, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=16/NW=1/H=0/*-http://se.kelkoo.com/b/a/ss_Ericsson_t65.html
78,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=16/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=pQvvS8piey8J:se.kelkoo.com/b/a/ss_Ericsson_t65.html
79,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=16/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&vst=0&vs=se.kelkoo.com
80, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=17/H=0/*-http://dk.kelkoo.com/b/a/ss_Sony_ericsson_t65.html
81,
http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=17/NW=1/H=0/*-http://dk.kelkoo.com/b/a/ss_Sony_ericsson_t65.html

82, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=17/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=anBmMtTP7AAJ:dk.kelkoo.com/b/a/ss_Sony_ericsson_t65.html

83, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=17/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&vst=0&vs=dk.kelkoo.com

84, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=18/H=0/*-http://no.kelkoo.com/b/a/ss_ericsson_t65.html

85, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=18/NW=1/H=0/*-http://no.kelkoo.com/b/a/ss_ericsson_t65.html

86, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=18/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=SasOO9ymuDAJ:no.kelkoo.com/b/a/ss_ericsson_t65.html

87, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=18/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&vst=0&vs=no.kelkoo.com

88, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=19/H=0/*-http://www.reviewcentre.com/reviews1082.html

89, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=19/NW=1/H=0/*-http://www.reviewcentre.com/reviews1082.html

90, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=19/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=DEkrlS0AyugJ:www.reviewcentre.com/reviews1082.html

91, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=19/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&vst=0&vs=www.reviewcentre.com

92, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=20/H=0/*-http://www.mobil.cz/mobilni_komunikace/mobilni_telefony/abecedni_prehled_mt/sonyericsson/ericssont65recenze011213.html

93, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS1/R=20/NW=1/H=0/*-http://www.mobil.cz/mobilni_komunikace/mobilni_telefony/abecedni_prehled_mt/sonyericsson/ericssont65recenze011213.html

94, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS5/R=20/*-http://cp.yahoo.net/search/cache?p=Ericsson+T65&ei=UTF-8&url=e36lHSU2cuYJ:www.mobil.cz/mobilni_komunikace/mobilni_telefony/abecedni_prehled_mt/sonyericsson/ericssont65recenze011213.html

95, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WS3/R=20/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&vst=0&vs=www.mobil.cz

96, http://us.rd.yahoo.com/search/navbar/pages/2/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&b=21

97, http://us.rd.yahoo.com/search/navbar/pages/3/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&b=41

98, http://us.rd.yahoo.com/search/navbar/pages/4/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&b=61

99, http://us.rd.yahoo.com/search/navbar/pages/5/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&b=81

100, http://us.rd.yahoo.com/search/navbar/pages/6/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&b=101

101, http://us.rd.yahoo.com/search/navbar/pages/7/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&b=121

102, http://us.rd.yahoo.com/search/navbar/pages/8/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&b=141

103, http://us.rd.yahoo.com/search/navbar/pages/9/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&b=161

104, http://us.rd.yahoo.com/search/navbar/pages/10/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&b=181

105, http://us.rd.yahoo.com/search/navbar/nexts/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&b=21

106, http://us.rd.yahoo.com/search/navbar/nexts/*-http://search.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&b=21

107, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WNIB/*-http://images.search.yahoo.com/search/images?p=Ericsson+T65&ei=UTF-8

108, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WNDB/*-http://search.yahoo.com/search/dir?p=Ericsson+T65&ei=UTF-8

109, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WNYB/*-http://yp.search.yahoo.com/search/ypredirect?p=Ericsson+T65&ei=UTF-8

110, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WNNB/*-http://search.news.yahoo.com/search/news?p=Ericsson+T65&ei=UTF-8

111, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WNPB/*-http://search.shopping.yahoo.com/search?p=Ericsson+T65&ei=UTF-8&cop=mss&__yltc=s:2766679,d:14489115,sec:srchtab,slk:products,k:QUERY.p_rd_safe%20%3EProducts%3C/a%3E%3C/b%3E%20%20%20%3C/pre%3E%0A%3C/div%3E%0A%3Ctable%20cellpadding=5%20cellspacing=0%20border=0%20width=100%%20bgcolor=e0e0e0%3E%3Cinput%20type=hidden%20name=x%20value=wrb%3E%3Ctr%3E%0A%3Ctd%20nowrap%20width=10%%3E%3Cb%3EYour%20Search:%3C/b%3E%20%3Cinput%20size=42%20name=va%20value=

112, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WASB/*-http://search.yahoo.com/web/advanced?p=Ericsson+T65&ei=UTF-8

113, http://drs.yahoo.com/S=2766679/K=Ericsson+T65/v=2/SID=e/l=WPRB/*-http://search.yahoo.com/search/preferences?pref_done=http%3A//search.yahoo.com/search%3Fp=Ericsson%2520T65&pref_cancel=http%3A//search.yahoo.com/search%3Fp=Ericsson%2BT65%26ei=UTF-8

114, http://rd.yahoo.com/M=224039.1984930.4456222.1922511/D=yahoo/S=2766679:FOOT/A=1524869/R=1/*http://privacy.yahoo.com/privacy/us/ysearch/

115,

```
http://rd.yahoo.com/M=224039.1984930.4456222.1922511/D=yahoo/S=2766679:FOOT/A=1524869/R=2/*http://docs.yahoo.com/info/terms/
116, javascript:ADFlaunch()
Service Time:1.33884716033936
```

Different to the &&AQUERYC command only shows the text version of webpage contents, these links are also be numbered, and the pages of these links can be fetched again by commands '&&AFWID(*no,identification number*)' by both specifying those numbers and the identification which is given in the first line of the result. To achieve the functionality, the system will store these results into the result table of database of e-MARE. When the user executes the &&AFWID command, the system will refer the data item of that database to see if the link is valid or not. If the link is valid, it will fetch the document of that page, and send it to the mailbox of the user. Of course, the links which are stored in the database do not occupy the resources forever. Actually, the system will automatically remove those out-of-date data periodically.

This example shows the possibility for the user to extract diverse information on the Internet (such as stock quote, dictionary word querying, game box scores, and real-time news) by specifying the appropriate querying string. If the user is familiar with the encoding format of querying string for any specified search engine, the link of that search engine and the encoding method of querying strings can be added, removed, and modified through the WWW interface of the web site of e-MARE. This is a great flexible way to expand the information retrieval ability of the mobile devices.

6.6.3 Example III: Fetching any web pages on the Internet

Not all users want to query the information through the search engine by entering the keyword. Actually, in many cases, most of the users only want to fetch some specified web page to review some real time information on those web sites (i.e. http://news.yam.com/general to review the real-time news in Taiwan, http://tw.stock.yahoo.com/s/tse.html to quote the stock prices from TAIEX, and http://www.twvs.tnc.edu.tw/cgi-bin/bbs/list to browse the title of posts of school's discussing board). The major common characteristic of those mentioned web sites is the contents will be automatically updated. In general, every time when the user is browsing one of those web pages, their contents will be different. That is, for the user, the address of URL is all the same, but the contents are always different. Therefore, if the user wants to view those mentioned pages by e-MARE via her mobile device, the system may just simply querying those web site in the same URL every time. This work can be done by &&AFW(*no*)

command. Of course, at the first, the user also has to request the available built-in web page links. This work can be done by command, &&SAW, and the results of &&SAW is shown as the follows:

```
1,交通大學首頁,http://www.nctu.edu.tw
2,曾文家商首頁,http://www.twvs.tnc.edu.tw
3,Yahoo's Homepage,http://www.yahoo.com
4,奇摩首頁,http://www.kimo.com.tw
5,Google's Homepage,http://www.google.com.tw
6,Whitehouse's homepage,http://www.whitehouse.gov
7,IC 製造類股即時股票行情,http://tw.stock.yahoo.com/s/kimo_listE04.html
8,華碩電腦當日即時股價,http://tw.stock.yahoo.com/s/s/kimo_day2357.html
9,蕃薯藤即時新聞總覽,http://news.yam.com/general
10,The news of Whitehouse,http://www.whitehouse.gov/news
11,曾文家商 PDA 精簡版,http://www.twvs.tnc.edu.tw/pda
12,中時電子報,http://www.chinatimes.com
13,NBA offical site,http://www.nba.com

Service Time:0.0338478088378906
```

If the we are interested in the contents of the webpage of USA White House, according to the list shown at above, we compose &&AFW(6) as the subject of the operational e-mail, the results will soon be ready in the user's mailbox, and it is shown as follows:

```
    Skip to ContentText OnlyGo to Search
    Welcome to the White House PresidentNewsVice PresidentHistory & ToursFirst LadyMrs. Cheney
    Welcome to the White House GovernmentKids OnlyEspanolContactPrivacy PolicySiteMapSearch
    Welcome to the White House Receive Email Updates
    In Focus
    Medicare
    Iraq
    National Security
    Economic Security
    Homeland Security
    More Issues
    En Espa 隳 l
    News
    Current News
    Video
    Press Briefings
    Proclamations
    Executive Orders
    Radio Addresses
    Appointments
    Nominations
    Application
    Offices
    Global Communications
    USA Freedom Corps
    Faith-Based & Community
    OMB
    More Offices
    Major Speeches
    Saddam Capture
```

UN Address
National Address
Iraqi Freedom
National Address
State of the Union
Photos

Arriving for the State Dinner President George W. Bush, Philippine President Gloria Macapagal-Arroyo, her husband Jose Miguel Arroyo and Laura Bush greet the press from the North Portico of the White House Monday, May 19, 2003.

Photo Essays
December 17, 2003 | Last Updated 7:52 p.m. (EST)
President Commemorates 100th Anniversary of Wright Brothers Flight
President Bush on Wednesday spoke from the Wright Brothers National
Memorial at the First Flight Centennial Celebration. full story
----
E-Gov Chief Discusses Electronic Government on Ask the White House
Karen EvansE-Gov Chief Karen Evans discussed her vision for
E-Government on the first anniversary of the signing of the E-Gov Act
of 2002. Read the full transcript. full story

Commerce Secretary Don Evans Reads "Cowboy Night Before Christmas"
Don
Evans Secretary of Commerce Don Evans participated in the second
night of holiday bedtime stories by reading "Cowboy Night Before
Christmas." full story

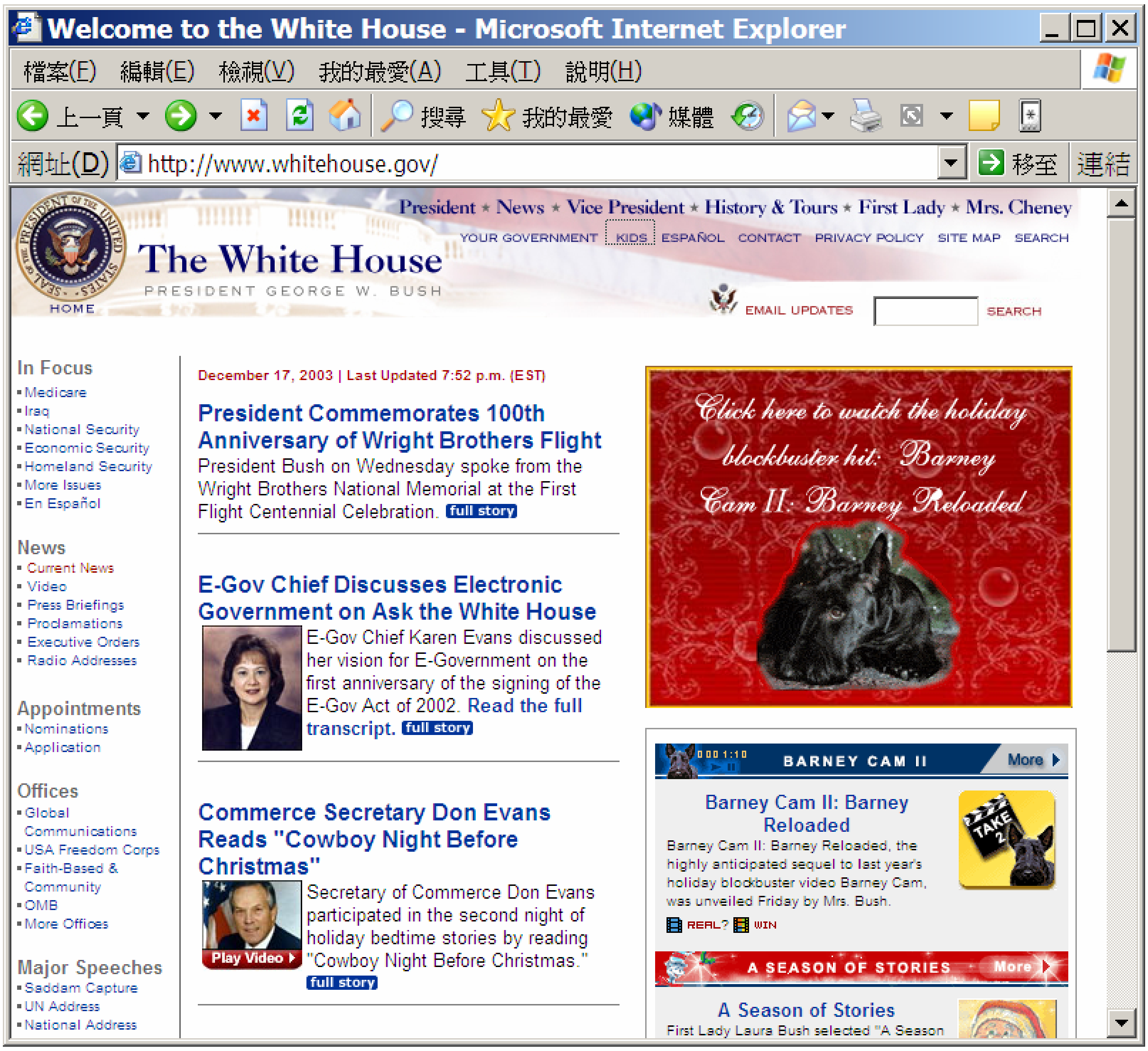
FTC Chair Discusses Anti-Spam Bill on Ask the White House
FTC Chairman Tim Muris Federal Trade Commission Chairman Tim Muris
appeared on "Ask the White House" Tuesday to discuss the anti-SPAM
bill. Read the transcript.
Fact sheetFact Sheet: Anti-Spam Law
full story

President Bush Signs American Dream Downpayment Act of 2003
President Bush on Tuesday signed into law the American Dream
Downpayment Act of 2003, which will help approximately 40,000 families
a year with their down payment and closing costs, and further
strengthen America's housing market. full story en Espa 隳 l
Fact sheetIn Focus: Home Ownership

More White House News
Barney Cam II: Barney Reloaded
Barney Cam II
Barney Cam II: Barney Reloaded Barney Cam II: Barney Reloaded, the
highly anticipated sequel to last year's holiday blockbuster video
Barney Cam, was unveiled Friday by Mrs. Bush.
Real Media Format - Click Here to view video
Windows Media Format - Click here to view video
Barney Cam 2
A Season of Stories
A Season of StoriesFirst Lady Laura Bush selected "A Season of
Stories" combining the wonder of the season with the magic of classic
children's stories that have shaped American culture.
Ready to Read - Just Click Here
Ask the White House
E-Gov Chief Discusses Electronic GovernmentE-Gov Chief Karen Evans
discussed her vision for E-Government on the first anniversary of the
signing of the E-Gov Act of 2002. Read the full transcript..
Ask the White House

```
    Economic Security
    Tax Relief Helps Economic Growth New job figures released on December
    5, 2003 and other recent indicators show America's economy is gaining
    strength, and the President's jobs and growth plan is working.
    President |   Vice President |   First Lady |   Mrs. Cheney |   News &
    Policies
    History & Tours |   Kids |   Your Government |   Appointments |   Jobs |
    Contact |   Text only
    Accessibility |   Search |   Privacy Policy |   Help
Service Time:1.45368385314941
```

To compare the result with the original web page, we show the original web page in Figure 6.16.



December 17, 2003 | Last Updated 7:52 p.m. (EST)

President Commemorates 100th Anniversary of Wright Brothers Flight

President Bush on Wednesday spoke from the Wright Brothers National Memorial at the First Flight Centennial Celebration.

E-Gov Chief Discusses Electronic Government on Ask the White House

E-Gov Chief Karen Evans discussed her vision for E-Government on the first anniversary of the signing of the E-Gov Act of 2002. Read the full transcript.

Commerce Secretary Don Evans Reads "Cowboy Night Before Christmas"

Secretary of Commerce Don Evans participated in the second night of holiday bedtime stories by reading "Cowboy Night Before Christmas."

Barney Cam II: Barney Reloaded

Barney Cam II: Barney Reloaded, the highly anticipated sequel to last year's holiday blockbuster video Barney Cam, was unveiled Friday by Mrs. Bush.

A Season of Stories

First Lady Laura Bush selected "A Season

Figure 6.16: The webpage of http://www.whitehouse.gov.

As mentioned earlier, all information on the Internet can not be covered in a single search engine. To retrieve the comprehensive information on the Internet or even local area network, the user has to refer more than one information sources. There are many researches

[2][5][19][43][45] have been dedicated to related topic. No matter what the proposed system of research is, most of the systems provide the unified WWW interface for the user to retrieve the information. In this situation, the integrated multi-search system may be treated as a higher level mediator for many different information sources, and it also provides the higher level querying commands such as SQL-like querying commands in reference [45]. However, to fetch these integrated information retrieval systems, the accessing method of e-MARE is still the same except the keyword will be leveraged into the higher level SQL-like querying command.

6.6.4 Example IV: Executing a program in execution server of e-MARE

The e-MARE can not only allow the user to access the web pages on the Internet but also allow the user to execute the program remotely. The user can install some programs via the WWW interface in advance, and then execute these programs by specifying the desire commands. The WWW interface has been introduced in Section 6.4. At the first, the user can list the available programs of e-MARE by specifying the following commands:

```
&&PWD(secret);&&SAP;
```

The results will be returned as follows:

```
1,ps ax
2,netstat
3,finger
4,./fetchTtable
5,public_html/programs/minhuang/rtnews.php
6,who
7,ls
8,pwd
9,public_html/programs/minhuang/lookupw.php

Service Time:0.0328488349914551
```

Again, the user can encode the following commands to execute the second program on the remote execution server:

```
&&PWD(secret);&&ARE(2);
```

The results for executing the 'netstat' command in the execution server are shown as follows:

```
The command executed result is shown as follows:
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address            Foreign Address          State
tcp        0      0 mail.twvs.tnc.edu:40115 mail.twvs.tnc.ed:telnet ESTABLISHED
tcp        0      0 mail.twvs.tnc.ed:telnet mail.twvs.tnc.edu:40115 ESTABLISHED
tcp        0      0 mail.twvs.tnc.edu.:smtp msr86.hinet.net:52275   TIME_WAIT
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node Path
unix  9      [ ]         DGRAM                    1135   /dev/log
unix  3      [ ]         STREAM     CONNECTED     1919   /tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTED     1918
unix  3      [ ]         STREAM     CONNECTED     1911   /tmp/.font-unix/fs7100
unix  3      [ ]         STREAM     CONNECTED     1910
unix  3      [ ]         STREAM     CONNECTED     1913   /tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTED     1901
unix  2      [ ]         DGRAM                    1873
unix  2      [ ]         DGRAM                    1698
unix  2      [ ]         DGRAM                    1603
unix  2      [ ]         DGRAM                    1525
unix  2      [ ]         DGRAM                    1450
unix  2      [ ]         DGRAM                    1201
unix  2      [ ]         DGRAM                    1147
[runner@mail runner]
```

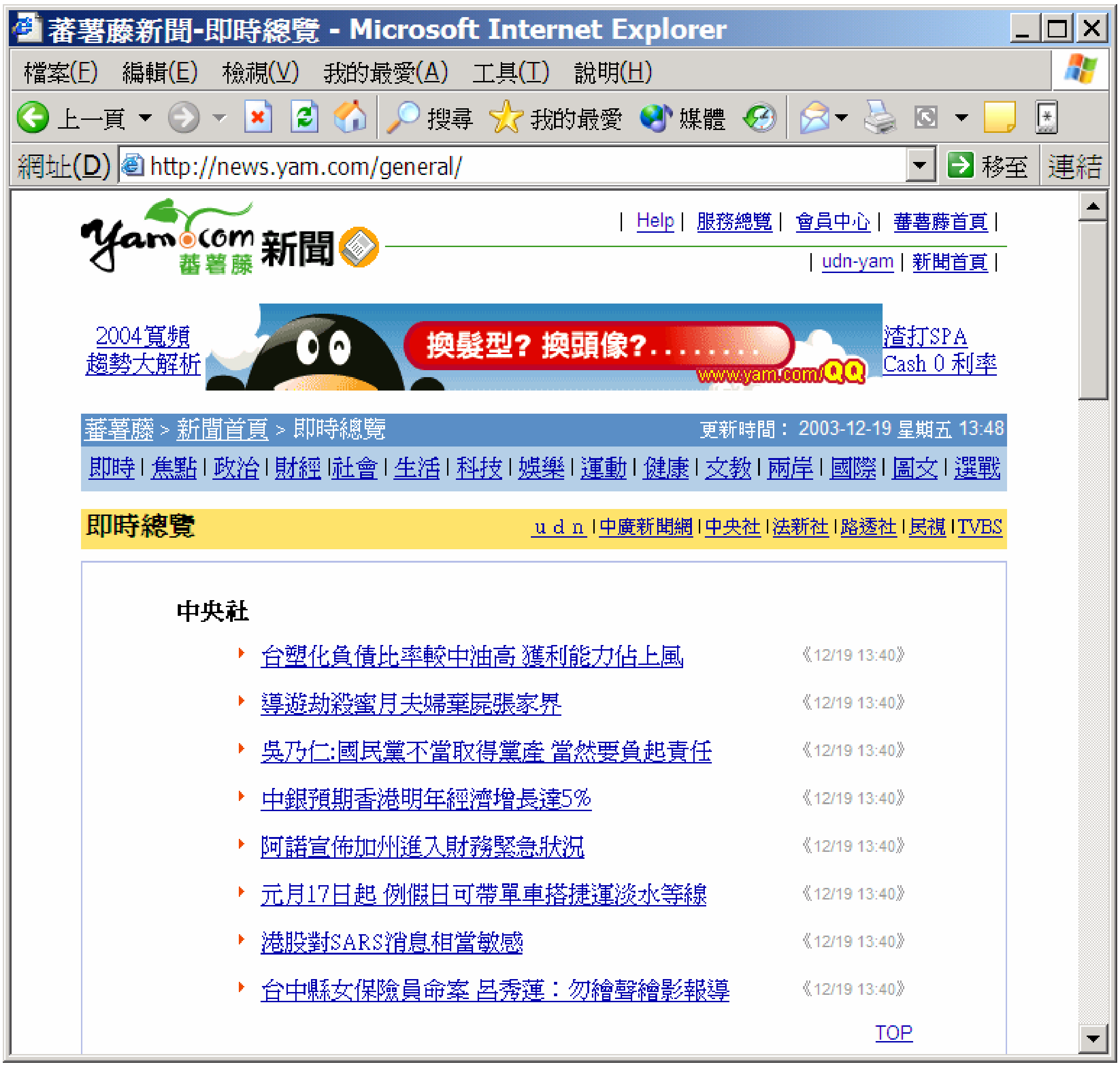
The above examples are utilizing the built-in shell commands of the server to achieve the user’s purpose. In theory, every shell programs and shell commands of the *Mobile Agent Execution Server* (Which is a RedHat Linux server) can be found and executed by e-MARE properly. In our prototype system, we utilize the following code fragments to achieve the execution purpose (issueAgent.pl):

```
my $telnet = Net::Telnet->new(RHOST);
$telnet->login(RUSER,RPASS);
my @lines = $telnet->cmd($rcmds);
$result .= "The command executed result is shown as follows:\n";
foreach (@lines)
{
  $result .= $_;
}
$result .= "\n";
```

The codes of fragment are a simple version of issueAgent.pl. It mainly utilizes Net::Telnet module to login into the *Mobile Agent Execution Server*, and sends the commands (which is encoded in $rcmds) to the server, then collects the results through the standard I/O mechanism. It is just a gateway for the user to send and execute her desired commands remotely via her mobile devices. The main works for simply information retrieval are all completed in the users’ customize program.

If the user wants to write some customize program to filter the information on the Internet in her own way, she may write her own program in any programming language, and upload it to the *Mobile Agent Execution Server*, and compose an operational e-mail to call the program, and letting that program to do the delegated works remotely.

For example, suppose the user wants to browse the real-time news on the http://news.yam.com/general, the popular way is to activate a web browser and enter the desired URL.　The documents of http://news.yam.com/general/ are shown in Figure 6.17.


蕃薯藤新聞-即時總覽 - Microsoft Internet Explorer
檔案(F)　編輯(E)　檢視(V)　我的最愛(A)　工具(T)　說明(H)
上一頁　搜尋　我的最愛　媒體
網址(D)　移至　連結

http://news.yam.com/general/

| Help | 服務總覽 | 會員中心 | 蕃薯藤首頁 |

| udn-yam | 新聞首頁 |

2004寬頻趨勢大解析

換髮型？換頭像？........ www.yam.com/QQ

渣打SPA Cash 0 利率

蕃薯藤 > 新聞首頁 > 即時總覽　更新時間：2003-12-19 星期五 13:48

即時 | 焦點 | 政治 | 財經 | 社會 | 生活 | 科技 | 娛樂 | 運動 | 健康 | 文教 | 兩岸 | 國際 | 圖文 | 選戰

即時總覽　u d n | 中廣新聞網 | 中央社 | 法新社 | 路透社 | 民視 | TVBS

中央社

- 台塑化負債比率較中油高 獲利能力佔上風　《12/19 13:40》
- 導遊劫殺蜜月夫婦棄屍張家界　《12/19 13:40》
- 吳乃仁:國民黨不當取得黨產 當然要負起責任　《12/19 13:40》
- 中銀預期香港明年經濟增長達5%　《12/19 13:40》
- 阿諾宣佈加州進入財務緊急狀況　《12/19 13:40》
- 元月17日起 例假日可帶單車搭捷運淡水等線　《12/19 13:40》
- 港股對SARS消息相當敏感　《12/19 13:40》
- 台中縣女保險員命案 呂秀蓮：勿繪聲繪影報導　《12/19 13:40》

TOP

Figure 6.17: The screen snapshot of http://news.yam.com/general.

In fact, the built-in command ‘&&AFW’ can partly fulfill the user’s purpose.　However, the command ‘&&AFW’ can only simply filter unwanted HTML tags for general purpose.　It can not custom the specified web documents to fulfill the user’s desire according to the characteristic of each kind of web page.　In this case, utilizing the &&AFW commands to fetch http://news.yam.com/general web page will return the results which are shown as the follows:

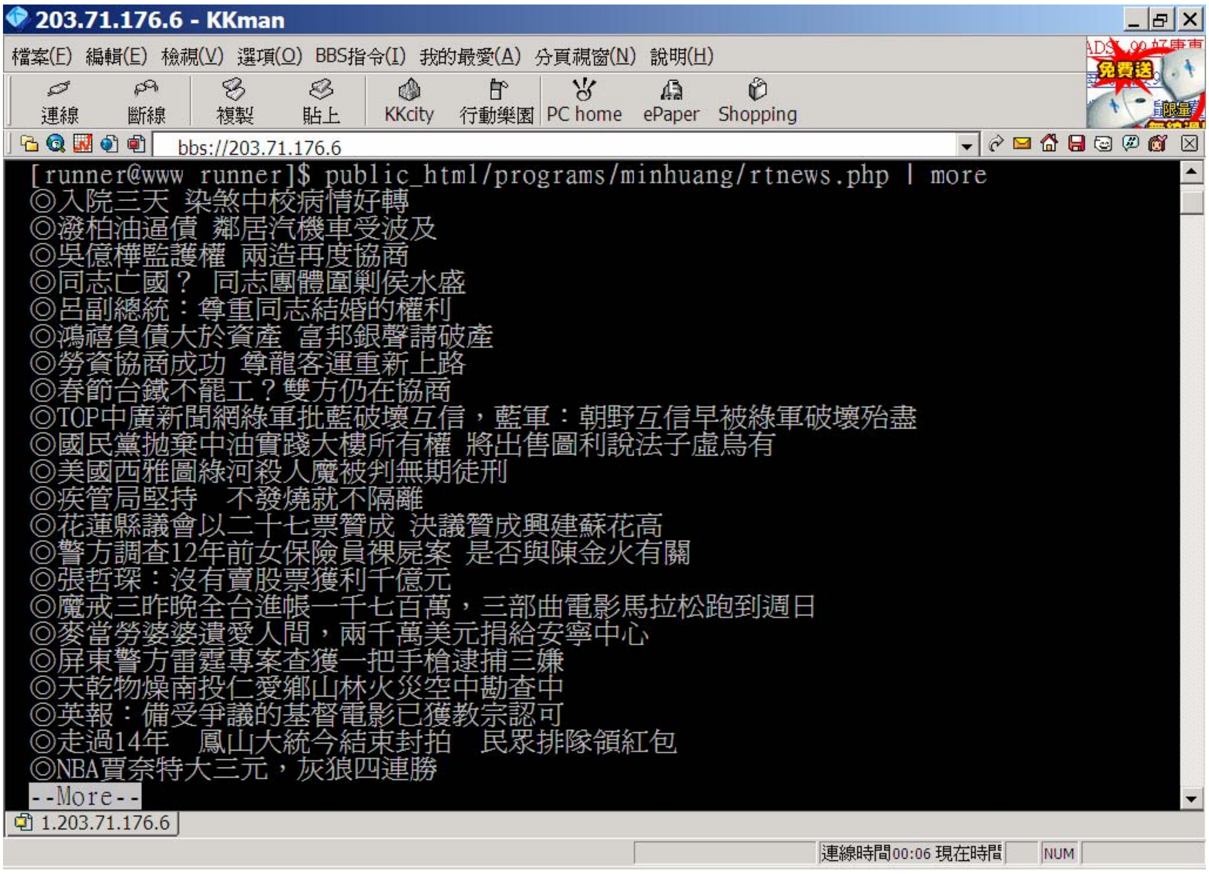
```
    yam.com

    ｜ Help｜ 服務總覽｜ 會　中腄 U 蕃薯藤熄｜
    ｜udn-yam｜新聞熄｜
    蕃薯藤 > 新聞熄 > 即時總覽
    更新時間： 2003-12-18 星期四 00:33
    即時總覽
    ｕｄｎ | 中廣新聞網 | 中央社 | 法新社 | 路透社 | 民視 | TVBS
    民視
    女安毒犯拒捕 持榔頭襲警
    《12/18 00:16》
    醉漢酒駕撞人 扭送警局仍發酒瘋
    《12/18 00:16》
    與染煞中校接觸 新加坡 70 人隔離
    《12/18 00:16》
    台灣驚爆染煞病患 WHO 關切
    《12/18 00:16》
    輕艷徽鷴 p 臨大敵 嚴控台灣旅客
    《12/18 00:16》
    長青劇 "親惜ф較"邁向介 g 年
    《12/18 00:16》
    斗六國中教育組長 被控不當體罰
    《12/18 00:16》
    供智障者工作？ 詐騙新招超惡劣
    《12/18 00:16》
    李雲寧 獲頒蒂 w 終生成就獎
    《12/18 00:16》
    TOP
    --------------------------------------------------------------------
    中央社
    職棒領隊會議決議封殺代訓球  國外發展
    《12/17 23:40》
    TOP
    --------------------------------------------------------------------
    TVBS
    女嫌襲警  榔頭牙齒齊下 3 人受傷
    《12/17 23:45》
    TOP
    --------------------------------------------------------------------
    下一目錄
    --------------------------------------------------------------------
Service Time:498015
```

Obviously, the above results have many unwanted information for the user. In this example, the user may utilize PHP programming language to write a real-time news fetching program (rtnews.php) as follows:

```
#!/usr/bin/php -q
<?php
   $fp = @fopen('http://news.yam.com/general', 'r');
   while($line = @fgets($fp, 1024))
   {
      $nline = ereg_replace("([\t \n])+", "\\1", $line);
      $nline = ereg_replace(" 《.*》 ", "\n◎", $nline);
      $contents .= trim($nline);
   }
```

```
    $contents = ereg_replace("蕃薯.*民視", "◎", $contents);
    $contents .="\n";
        $new_str =    strip_tags($contents);
        echo $new_str;
?>
```

The above will retrieve the document of the address ‘http://news.yam.com/general’, filter the unwanted tags, and reformat the display style for the documents. The sample results are shown in Figure 6.18.



```
[runner@www runner]$ public_html/programs/minhuang/rtnews.php | more
◎入院三天 染煞中校病情好轉
◎潑柏油逼債 鄰居汽機車受波及
◎吳億樺監護權 兩造再度協商
◎同志亡國？ 同志團體圍剿侯水盛
◎呂副總統：尊重同志結婚的權利
◎鴻禧負債大於資產 富邦銀聲請破產
◎勞資協商成功 尊龍客運重新上路
◎春節台鐵不罷工？雙方仍在協商
◎TOP中廣新聞網綠軍批藍破壞互信，藍軍：朝野互信早被綠軍破壞殆盡
◎國民黨拋棄中油實踐大樓所有權 將出售圖利說法子虛烏有
◎美國西雅圖綠河殺人魔被判無期徒刑
◎疾管局堅持  不發燒就不隔離
◎花蓮縣議會以二十七票贊成 決議贊成興建蘇花高
◎警方調查12年前女保險員裸屍案 是否與陳金火有關
◎張哲琛：沒有賣股票獲利千億元
◎魔戒三昨晚全台進帳一千七百萬，三部曲電影馬拉松跑到週日
◎麥當勞婆婆遺愛人間，兩千萬美元捐給安寧中心
◎屏東警方雷霆專案查獲一把手槍逮捕三嫌
◎天乾物燥南投仁愛鄉山林火災空中勘查中
◎英報：備受爭議的基督電影已獲教宗認可
◎走過14年  鳳山大統今結束封拍  民眾排隊領紅包
◎NBA賈奈特大三元，灰狼四連勝
--More--
```

Figure 6.18: The results after executing the rtnews.php.

As shown in both Figure 6.17 and Figure 6.18, the original document in Figure 6.17 is rich in several images, and formats. However, through the ‘rtnews.php’ programs, most of the unwanted tags and formats will be cleaned. The filtered result is concise, and it is encoded in ASCII standard text. Hence it is very suitable for reading by mobile user via the resource-limited mobile devices.

Again, although the program code is simple and clean, but the tag strip and web documents fetching functions are both time-consuming works. In the most cases, they always need a powerful CPU with lots of resources. It is impossible for many resource-constraint mobile devices. However, through the remote program executing mechanism of e-MARE, the news on the Internet now can be gotten by mobile devices in a very easy and simple way.

For example, we may compose the following command in e-mail client to launch ‘rtnews.php’ via mobile device:

```
&&PWD(secret);&&ARE(5);
```

The results will soon be ready in user’s mailbox, and it is shown as the follows:

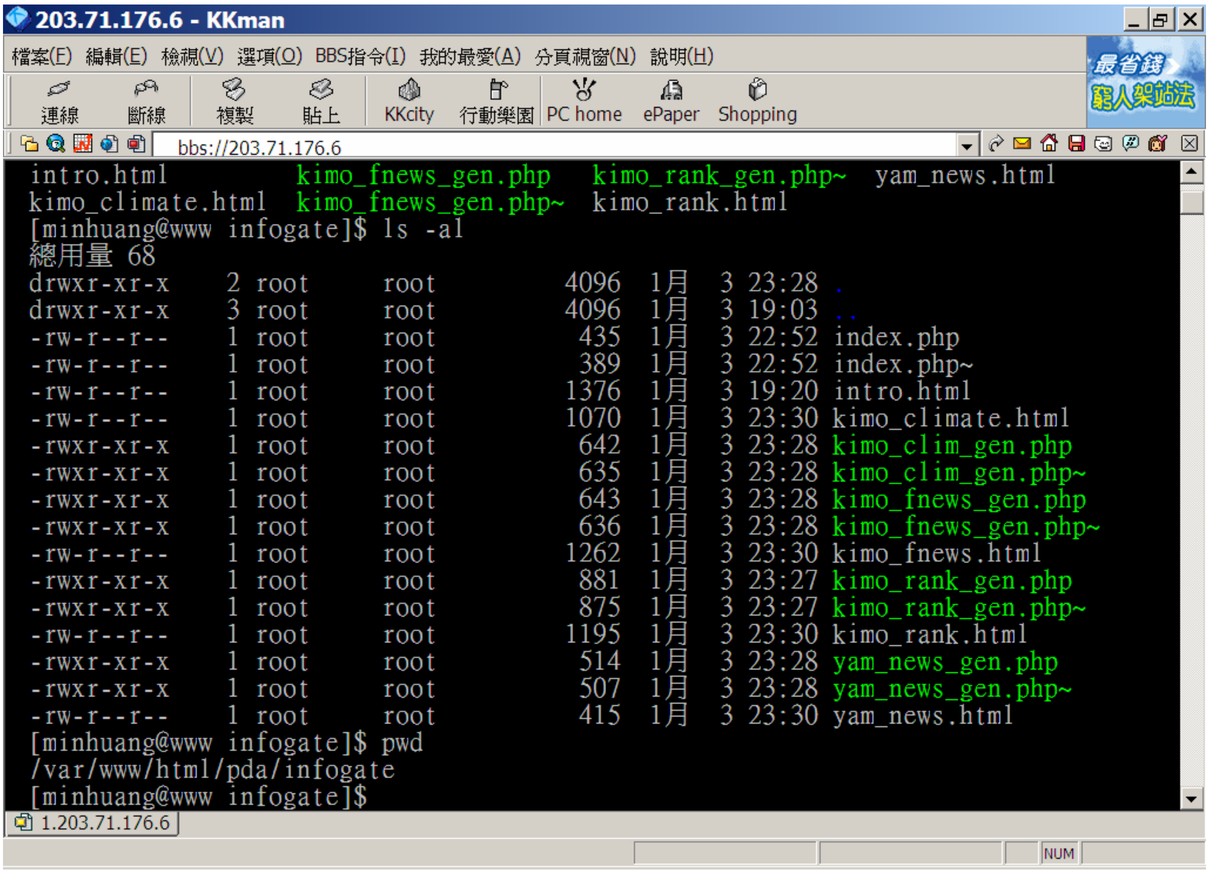
```
The command executed result is shown as follows:
◎入院三天  染煞中校病情好轉
◎潑柏油逼債  鄰居汽機車受波及
◎吳億樺監護權  兩造再度協商
◎同志亡國？  同志團體圍剿侯水盛
◎呂副總統：尊重同志結婚的權利
◎鴻禧負債大於資產  富邦銀聲請破產
◎勞資協商成功  尊龍客運重新上路
◎春節台鐵不罷工？雙方仍在協商
◎TOP 中廣新聞網綠軍批藍破壞互信，藍軍：朝野互信早被綠軍破壞殆盡
◎國民黨拋棄中油實踐大樓所有權  將出售圖利說法子虛烏有
◎美國西雅圖綠河殺人魔被判無期徒刑
◎疾管局堅持    不發燒就不隔離
◎花蓮縣議會以二十七票贊成  決議贊成興建蘇花高
◎警方調查 12 年前女保險員裸屍案  是否與陳金火有關
◎張哲琛：沒有賣股票獲利千億元
◎魔戒三昨晚全台進帳一千七百萬，三部曲電影馬拉松跑到週日
◎麥當勞婆婆遺愛人間，兩千萬美元捐給安寧中心
◎屏東警方雷霆專案查獲一把手槍逮捕三嫌
◎天乾物燥南投仁愛鄉山林火災空中勘查中
◎英報：備受爭議的基督電影已獲教宗認可
◎走過 14 年    鳳山大統今結束封拍    民眾排隊領紅包
◎NBA 賈奈特大三元，灰狼四連勝
◎需自主健康管理者，至少有一名同機旅客失聯
◎強烈冷氣團南下，溫度入夜後一路走低：7~9 度
◎香港發燒病人接受 SARS 測試呈陰性反應
◎TOP 路透社[醫學新知]持續受衣原體感染的女性罹患子宮頸癌危險較高--研究
◎自行車-美國名將阿姆斯特朗可能將再參加兩次環法大賽
◎[醫學新知]長期接受抗愛滋藥物治療會增加罹患心臟病的危險
◎TOP 下一目錄
[runner@www runner]
Service Time:653129
```

Clearly, the results generated by customized program is much better than the one that generated by &&AFW command directly fetched from that specified web page. It also means that the framework has the great flexibility for the user to design and utilize their customized application on the remote host via his resource-constraint mobile device.

Alternately, to speed up whole process for querying the news or information on the

Internet, there is also a feasible way by the automatically regular program launcher, 'cron' daemon, under UNIX-based operating system. That is, in UNIX-based *Mobile Agent Runtime Server*, the user may create cron table to regularly execute the user's customized information retrieval application. The application should fetch the documents from the target information sources, analyze the contents of those documents, and generate those results into a simple version HTML file. The HTML file is then stored at the home directory of user's web site in a structural format. Finally, the user may directly use &&AFW command of e-MARE to fetch the arranged information from the other information source on the Internet.

For example, we install some programs for generating the extracted information on the specified home directory of the operating system, which is shown in Figure 6.19.



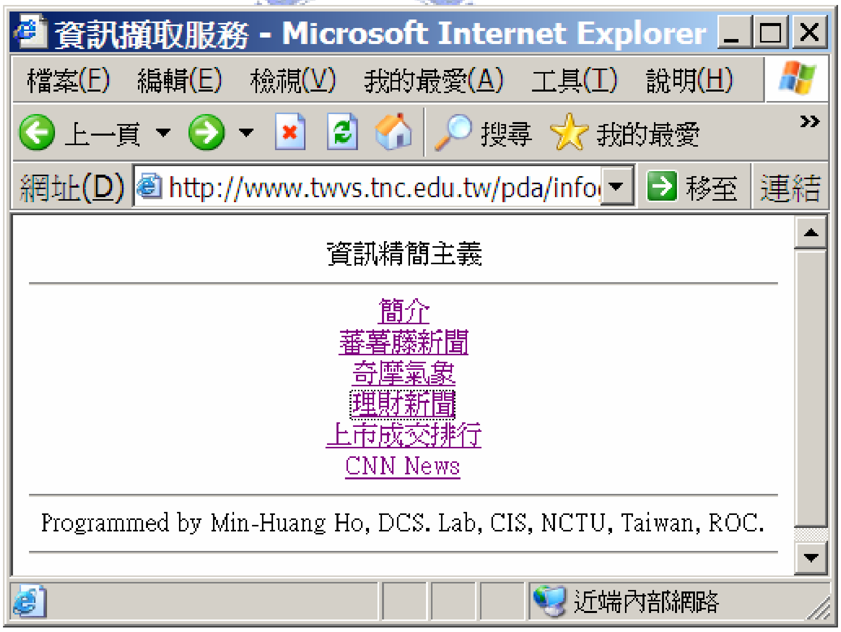
```
intro.html         kimo_fnews_gen.php   kimo_rank_gen.php~   yam_news.html
kimo_climate.html  kimo_fnews_gen.php~  kimo_rank.html
[minhuang@www infogate]$ ls -al
總用量 68
drwxr-xr-x    2 root     root         4096  1月   3 23:28 .
drwxr-xr-x    3 root     root         4096  1月   3 19:03 ..
-rw-r--r--    1 root     root          435  1月   3 22:52 index.php
-rw-r--r--    1 root     root          389  1月   3 22:52 index.php~
-rw-r--r--    1 root     root         1376  1月   3 19:20 intro.html
-rw-r--r--    1 root     root         1070  1月   3 23:30 kimo_climate.html
-rwxr-xr-x    1 root     root          642  1月   3 23:28 kimo_clim_gen.php
-rwxr-xr-x    1 root     root          635  1月   3 23:28 kimo_clim_gen.php~
-rwxr-xr-x    1 root     root          643  1月   3 23:28 kimo_fnews_gen.php
-rwxr-xr-x    1 root     root          636  1月   3 23:28 kimo_fnews_gen.php~
-rw-r--r--    1 root     root         1262  1月   3 23:30 kimo_fnews.html
-rwxr-xr-x    1 root     root          881  1月   3 23:27 kimo_rank_gen.php
-rwxr-xr-x    1 root     root          875  1月   3 23:27 kimo_rank_gen.php~
-rw-r--r--    1 root     root         1195  1月   3 23:30 kimo_rank.html
-rwxr-xr-x    1 root     root          514  1月   3 23:28 yam_news_gen.php
-rwxr-xr-x    1 root     root          507  1月   3 23:28 yam_news_gen.php~
-rw-r--r--    1 root     root          415  1月   3 23:30 yam_news.html
[minhuang@www infogate]$ pwd
/var/www/html/pda/infogate
[minhuang@www infogate]$
```

Figure 6.19: Installing some programs at the home directory on the operating system.

As Figure 6.19 shows, 'yam_news_gen.php' is the program for generating the extracted information, and the 'yam_news.html' is the extracted information file. As described earlier, the file, 'yam_news.html', basically is a concise text version of target web documents. In the file 'index.php', we may directly link the concise text version at our homepage. The file of 'index.php' is shown as follows:

```
<html>
<head>
<title>
資訊擷取服務
</title>
</head>
<body>
<center>
資訊精簡主義
<hr>
<a href='intro.html'>簡介</a><br>
<a href='yam_news.html'>蕃薯藤新聞</a><br>
<a href='kimo_climate.html'>奇摩氣象</a><br>
<a href='kimo_fnews.html'>理財新聞</a><br>
<a href='kimo_rank.html'>上市成交排行</a><br>
<a href='cnn_news.html'>CNN News</a><br>
<hr>
Programmed by Min-Huang Ho, DCS. Lab, CIS, NCTU, Taiwan, ROC.
<hr>
</center>
</body>
</html>
```

The screen for the above 'index.php' is shown in Figure 6.20.



http://www.twvs.tnc.edu.tw/pda/info

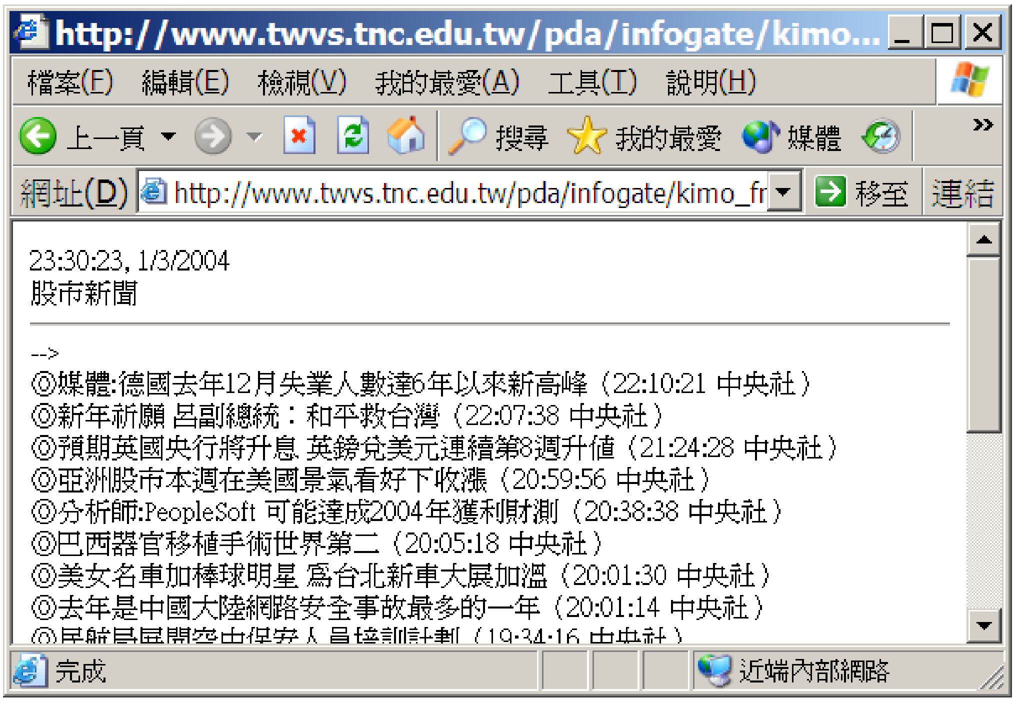
資訊精簡主義

簡介
蕃薯藤新聞
奇摩氣象
理財新聞
上市成交排行
CNN News

Programmed by Min-Huang Ho, DCS. Lab, CIS, NCTU, Taiwan, ROC.

Figure 6.20: The screen of index.php.

As Figure 6.20 shows, the user may browse the listed information sources by click the hyperlink on the screen. If the user has clicked the finance news of listed item, the screen will be shown as Figure 6.21.



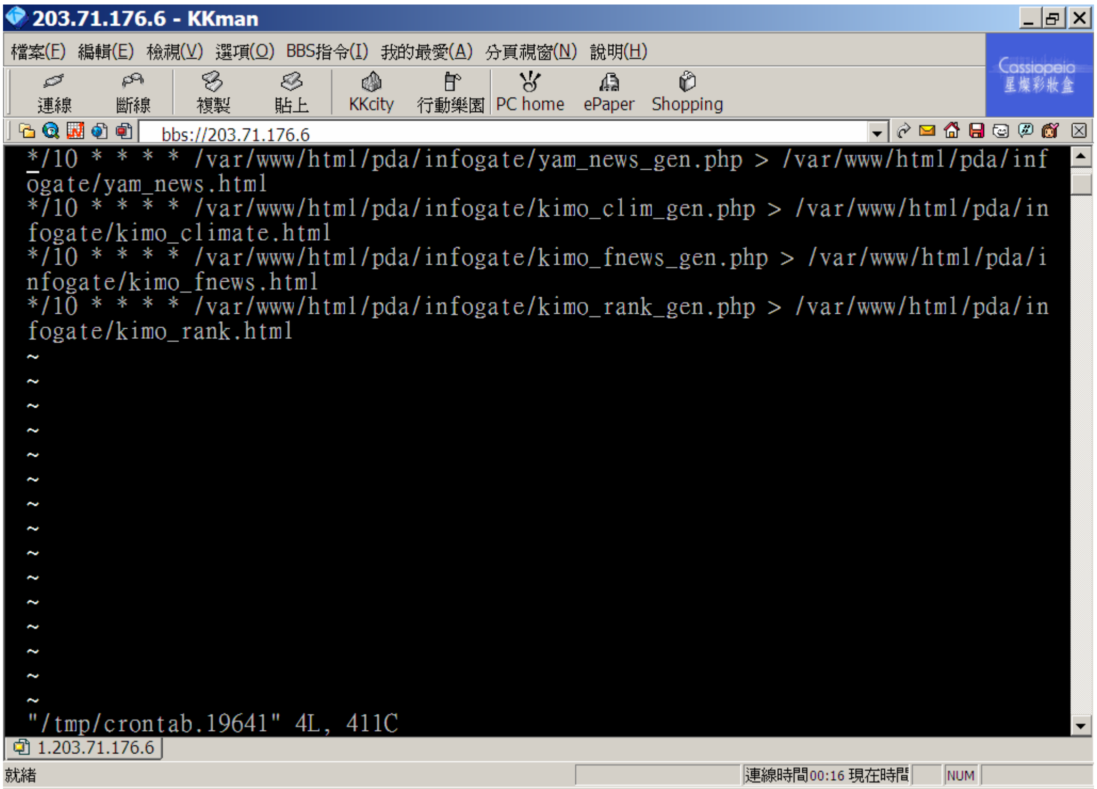
http://www.twvs.tnc.edu.tw/pda/infogate/kimo...
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
搜尋
我的最愛
媒體
網址(D)
http://www.twvs.tnc.edu.tw/pda/infogate/kimo_fr
移至
連結
完成
近端內部網路

23:30:23, 1/3/2004
股市新聞

-->
◎媒體:德國去年12月失業人數達6年以來新高峰（22:10:21 中央社）
◎新年祈願 呂副總統：和平救台灣（22:07:38 中央社）
◎預期英國央行將升息 英鎊兌美元連續第8週升值（21:24:28 中央社）
◎亞洲股市本週在美國景氣看好下收漲（20:59:56 中央社）
◎分析師:PeopleSoft 可能達成2004年獲利財測（20:38:38 中央社）
◎巴西器官移植手術世界第二（20:05:18 中央社）
◎美女名車加棒球明星 爲台北新車大展加溫（20:01:30 中央社）
◎去年是中國大陸網路安全事故最多的一年（20:01:14 中央社）
◎民航局展開空中保安人員培訓計劃（19:34:16 中央社）

Figure 6.21: The concise version of the finance news on the Internet.

Of course, to regularly generate these concise text version documents of information sources, to edit the cron table is very important. For example, the table may be looked the same as Figure 6.22. In this example, we install four programs for regular execution. After the table was installed into the operating system, the cron daemon will handle these installed programs to generate the appropriate text files, and finally, these text files can be fetched through WWW protocol. Because the extracting works are all done by server, the mobile device will not to worry about the executing environment.



```
*/10 * * * * /var/www/html/pda/infogate/yam_news_gen.php > /var/www/html/pda/inf
ogate/yam_news.html
*/10 * * * * /var/www/html/pda/infogate/kimo_clim_gen.php > /var/www/html/pda/in
fogate/kimo_climate.html
*/10 * * * * /var/www/html/pda/infogate/kimo_fnews_gen.php > /var/www/html/pda/i
nfogate/kimo_fnews.html
*/10 * * * * /var/www/html/pda/infogate/kimo_rank_gen.php > /var/www/html/pda/in
fogate/kimo_rank.html
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/tmp/crontab.19641" 4L, 411C
```

Figure 6.22: The cron table of our example.

6.6.5 Example V: Parallel and distributed execution programs in e-MARE

In the e-MARE framework, we provide a simple mechanism that allows the mobile user to execute several programs in parallel. Through the parallel execution mechanism, the user may specify the number of parallelism by the '&&APE(*no*)' command, and then, the following commands will be executed in parallel. For example, if the mobile user has composed the following subject of operational e-mail:

&&PWD(secret);&&APE(3);&&AFW(1);&&AQUERYC(5,Ericsson T65);&&ARE(2);

Thus, the commands '&&AFW(1)', '&&AQUERYC(5,Ericsson T65)', and '&&ARE(2)' will be executed by different worker agent. Each worker agent is responsible for one specified command. The worker agent can be at the same server with main mail server, or it can be at the different server too. All worker agents are independent, thus the results will be replied to mobile user respectively. That is, in this example, the user will receive three different replies from different worker agent. However, in some situations, the user may want to write a program to analysis all fetched documents, then extract and combine only the desired fields. In such situation, the user may compose another following subject of operational e-mail:

&&PWD(secret);&&APES(3);&&ARE(9);&&AQUERYC(5,Ericsson T65);&&AFW(2);

The &&APES will do the above mentioned works for the user. In this example, the results of last two commands (&&AQUERYC(5,Ericsson T65), and &&AFW(2)) will be redirected to the first command (&&ARE(9)). That is, the program specified by the first command will get the chance to analysis the results of the following two commands. This will let the mobile user obtain the chance to write a program for filtering the results, and then sends a more precise or concise result to the user.

Again, the worker agent can be installed via WWW interface of e-MARE. Of course, the system developer may also manually install the system into any e-mail ready server. The related information of installed worker agent also has to be registered into the database server of e-MARE. Hereafter, the &&APE commands can be applied to specify the operational e-mail executing the subsequent commands in parallel way. The number of parallelism and the place of worker agent can be referred from the database of e-MARE.

Through the above mentioned mechanism, the system developer or mobile user may install many information retrieval applications or programs into several servers under different networks, and write a program to structurally combine the fetched results which are returned from different server. It means that the user may achieve the distributed computation by mobile device under our e-MARE framework.

Although the runtime of each worker agent is independent, the worker agent also can communicate each other. However, due to the system and platform of worker agent can be very diverse, to directly communicate between two worker agents is very difficult. Thus, we implement a blackboard messaging mechanism to solve the situation. The blackboard messaging mechanism was installed in an independent database server of e-MARE, and the information of the database can be accessed by *BBCoordinator* and *MBCoordinator*. That is, if a worker agent wants contact to the other worker agent, it has to through *BBCoordinator* or *MBCoordinator.* This mechanism has been described in Section 5.5.5.

6.6.6 Example VI: Executing mobile agents in EAMRE

The previous examples are all encoded their commands in the subject of operational e-mail. The body of e-mail is needless to fill any content, but for some more complex tasks, the user may want to write some mobile agents to execute or contact to the available mobile agent runtime environment on the Internet. In e-MARE, if the mobile user wants to write a program in operational e-mail, the codes of the program has to be embedded in the body of e-mail, and be embraced with the symbol ‘ &&<’ and ‘>&&’. Currently, our framework can process three types of language, Perl, PHP, and shell script. The user has to specify the language type for processing by &&LANG(*type*) command. The *type* can be set as PERL, PHP, or SHELL. That is, if the subject of incoming operational e-mail is encoded as follows:

&&PWD(secret);&&LANG(PERL);

ManagerAgent will try to extract all of the statements between the symbol ‘&&<’ and ‘>&&’. All of the statements will be packed into a file, and be uploaded to the runner home directory of execution server. *IssueAgent* will execute the program remotely through TELNET protocol. All of the execution results have to be outputted to standard I/O. They will be redirected into the *ResultWrapper*. Of course, for some complex situation, the results also can be generated into a file. The *ResultWrapper* will also download that file, and return it to user as an attachment of e-mail. This mechanism is one of the best ways to write a mobile agent for accessing the traditional mobile agent runtime environment system which is available on the network.

1896

6.7 Discussions about transferring bytes and time comparisons

For the resource-limited mobile devices, the transferring bytes and execution time from the server for the user’s information retrieving operation is a very important consideration. In this section, we compare the transferring bytes and execution time between two different methods, the e-MARE framework and the traditional web browsing method. We design five distinct typical querying operations to compare the differences between e-MARE framework and traditional web browsing method. The five experiments are described in Table 6.2.

Table 6.2: The five distinct typical querying operations.

Typical querying operations	Detail description
E1: &&SASE;	Requesting the available search engines or information sources in the e-MARE's database.
E2:&&AQUERYC(7,2303);	Quoting the stock value from a free real-time stock quote web site (http://stock.kimo.com.tw) according to the results of E1.
E3:&&AQUERYL(5,T65);	Querying the keyword 'T65' to the famous search engine 'AltaVista' (http://www.altavista.com) according to the results of E1.
E4:&&SAP;	Listing all of the available remote programs pre-installed in the Execution Server.
E5:&&ARE(4,30);	Executing the fourth program of the Execution Server according to the results of E4, and passing the argument '30' to that program.

Experiment E1 is the most basic operation for a mobile user to query the useful information from the Internet by the e-MARE framework. The purpose of E1 is to list all the available search engines or information resources in the e-MARE's database. Each available search engines and information sources was settled to a distinct number, which can be specified in the following further querying commands, such like command '&&AQUERYC' and '&&AQUERYL'. In fact, the data item in the database of e-MARE can not be merely a simple URL but also a complex format of querying string. Any information source which is supporting HTTP protocol can be easily accessed via this mechanism.

Experiment E2 is a typical example for retrieving the information from the information source. In E2, we specify the 7th information source as the target for information retrieval. Because we has settled some appropriate arguments for 7th information resource for quoting a real-time stock information of Taiwan Stock Exchange (TAIEX) in advance, we can just easily provide the stock id number in the &&AQUERYC command, and the real-time stock information will be quoted and be sent to user's mailbox.

Providing the ability for a user to retrieve more complete information through the mobile device is also an important purpose of e-MARE. To full utilize the ready-made search engines on the Internet is the easiest way to achieve this purpose. Experiment E3 shows a feasible solution. We encode the querying string of a famous search engine 'AltaVista (http://www.altavista.com)' into 5th item of the e-MARE's database. Thus, the command of E3, &&AQUERYL(5,T65), will send the keyword, 'T65', to AltaVista. The e-MARE will filter all of the links of the querying results from AltaVista, and sent them back to the user's mailbox. Each link will be specified a distinct identification item number by e-MARE for further handling. Consequently, the mobile user may intensively surf all the information on the Internet which is provided by search engines through this mechanism.

The users also can execute some information retrieval application programs on the remote Execution Server. Experiment E4 shows how to list the available programs of e-MARE. The command, '&&SAP', will return an e-mail reporting all of the available remote programs of the Execution Server. Some information sources provide the Application Program Interface (API) for user to custom her own information retrieval application. In such case, the user may design an information retrieval program, and upload it into the Execution Server through the web interface of e-MARE framework. Then, in experiment E5, we can use command '&&ARE' to execute that program remotely. The meaning of E5 is to execute 4th program provided by Execution Server with the argument, 30. In this experiment, the 4th program is a school timetable querying program for a specified teacher. The passing argument is a teacher's id number of a senior high school. Different to the information which is filtered from the web page, the information provided by program can be prepared in a well-formed format. All the measurements for these five experiments are shown in the Table 6.3.

Table 6.3: File size and time consuming comparison.

Experiment Description	File size in e-MARE	Response Time (micro-second)	File size in Web
E1: &&SASE;	Depends on the item numbers of the search engines list.	81,684	The same as e-MARE.
E2:&&AQUERYC(7,2303);	2K Bytes	353,557	91.7K Bytes
E3:&&AQUERYL(5,T65);	9K Bytes	530,693	39.6K Bytes
E4:&&SAP;	Depends on the item numbers of the available programs list	131,324	N/A
E5:&&ARE(4,11);	1K Bytes	504,765	9.01K Bytes

As shown in Table 6.3, we found that all of the five operations have executed their works in a very short time, and in the most cases, the results will have been prepared in the user's mailbox before the user's next time fetching operation. Due to all of the results returned from the e-MARE server is basic standard ASCII format, the transferring bytes are dramatically reduced. We believe that the mechanism is very suitable for mobile devices to retrieve the information from the Internet.

Chapter 7

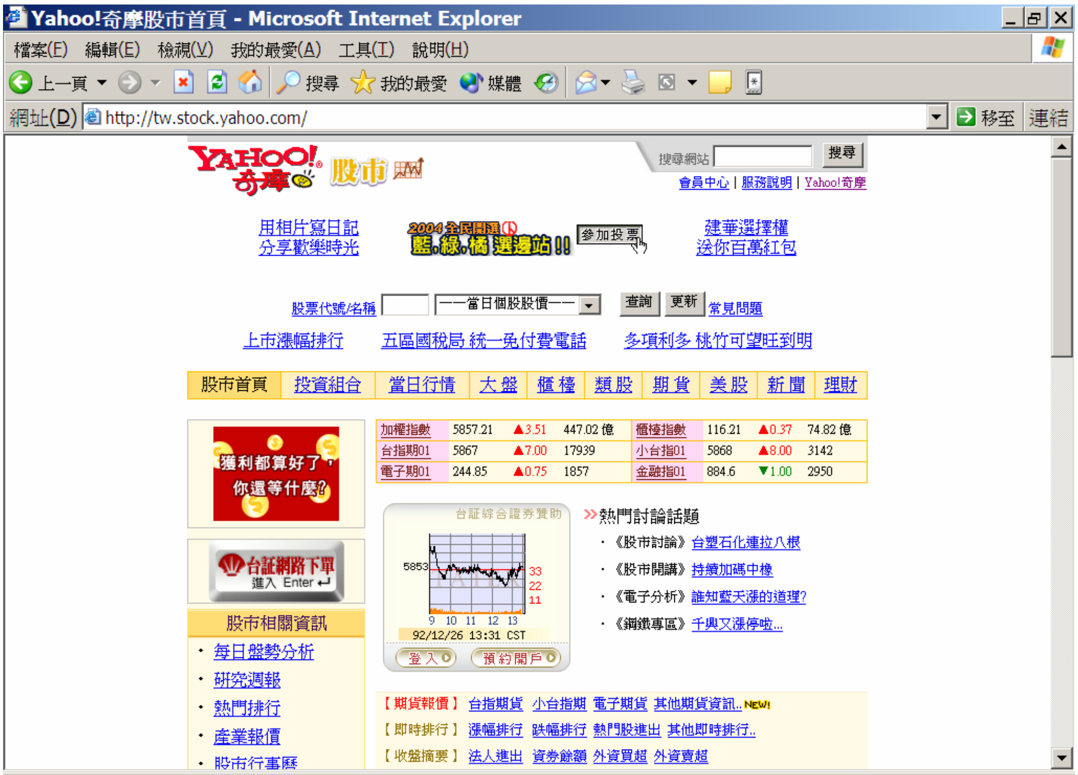
Applications

In this chapter, we will demonstrate some practical and useful applications which were applied the e-MARE framework. The outcomes from these applications show that e-MARE framework can improve information accessing ability and efficiency of the mobile users dramatically. It is very suitable and useful for many wireless mobile computing based information-oriented and information-concentrated projects.

7.1 Information retrieval on the Internet

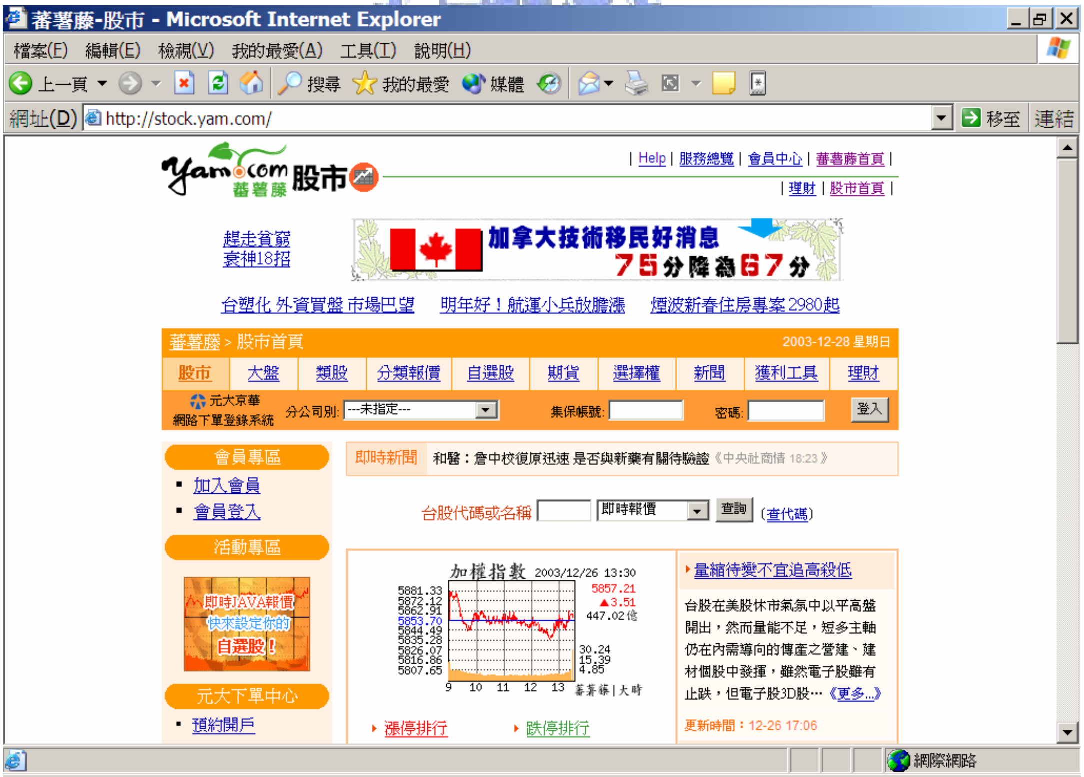
As the description in Chapter 6, the e-MARE allows the mobile user easily query the information on the Internet via her mobile device. In this section, we will go through the real whole information retrieval process in detail.

Suppose we are interested in quoting the real time stock value at anywhere in anytime. In general case, there are lots of free stock quote web site on the Internet. One of the most popular examples in Taiwan about TAIEX market is http://stock.kimo.com.tw and http://stock.yam.com. Figure 7.1 shows the example screen of http://stock.kimo.com.tw and Figure 7.2 show the example screen of http://stock.yam.com. In those web sites, the user may choose her interesting stock, and browse the real-time value and related information about that stock in the web browser on desktop or notebook PC. Most of these kinds of web sites provide many categories for the user to customize their browsing preferences. That is, for a stock quoting user, she may choose to browse all stock values of the same sort company or just choose one specific company.


Yahoo!奇摩股市首頁 - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http://tw.stock.yahoo.com/
移至 連結
YAHOO! 奇摩 股市
搜尋網站 搜尋
會員中心 | 服務說明 | Yahoo!奇摩
用相片寫日記 分享歡樂時光
2004全民開選 藍,綠,橘 選邊站!! 參加投票
建華選擇權 送你百萬紅包
股票代號/名稱 ——當日個股股價—— 查詢 更新 常見問題
上市漲幅排行 五區國稅局 統一免付費電話 多項利多 桃竹可望旺到明
股市首頁 投資組合 當日行情 大盤 櫃檯 類股 期貨 美股 新聞 理財
獲利都算好了，你還等什麼?
台証網路下單 進入 Enter
股市相關資訊
每日盤勢分析
研究週報
熱門排行
產業報價
股市行事曆
台証綜合證券贊助
5853
33
22
11
9 10 11 12 13
92/12/26 13:31 CST
登入 預約開戶
熱門討論話題
《股市討論》台塑石化連拉八根
《股市開講》持續加碼中橡
《電子分析》誰知藍天漲的道理?
《鋼鐵專區》千興又漲停啦...
【期貨報價】台指期貨 小台指期 電子期貨 其他期貨資訊.. NEW!
【即時排行】漲幅排行 跌幅排行 熱門股進出 其他即時排行..
【收盤摘要】法人進出 資券餘額 外資買超 外資賣超

加權指數	5857.21	▲3.51	447.02 億	櫃檯指數	116.21	▲0.37	74.82 億
台指期01	5867	▲7.00	17939	小台指01	5868	▲8.00	3142
電子期01	244.85	▲0.75	1857	金融指01	884.6	▼1.00	2950

Figure 7.1: http://stock.kimo.com.tw.


蕃薯藤-股市 - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http://stock.yam.com/
移至 連結
Yam.com 蕃薯藤 股市
| Help | 服務總覽 | 會員中心 | 蕃薯藤首頁 |
| 理財 | 股市首頁 |
趕走貧窮 衰神18招
加拿大技術移民好消息 75分降為67分
台塑化 外資買盤 市場巴望 明年好！航運小兵放膽漲 煙波新春住房專案2980起
蕃薯藤 > 股市首頁 2003-12-28 星期日
股市 大盤 類股 分類報價 自選股 期貨 選擇權 新聞 獲利工具 理財
元大京華 網路下單登錄系統 分公司別: ---未指定--- 集保帳號: 密碼: 登入
會員專區
加入會員
會員登入
活動專區
即時JAVA報價 快來設定你的自選股！
元大下單中心
預約開戶
台股代碼或名稱 即時報價 查詢 (查代碼)
加權指數 2003/12/26 13:30
5881.33
5872.12
5862.91
5853.70
5844.49
5835.28
5826.07
5816.86
5807.65
5857.21
▲3.51
447.02億
30.24
15.39
4.85
9 10 11 12 13
蕃薯藤|大時
漲停排行 跌停排行
網際網路

即時新聞 和醫：詹中校復原迅速 是否與新藥有關待驗證《中央社商情 18:23》

量縮待變不宜追高殺低

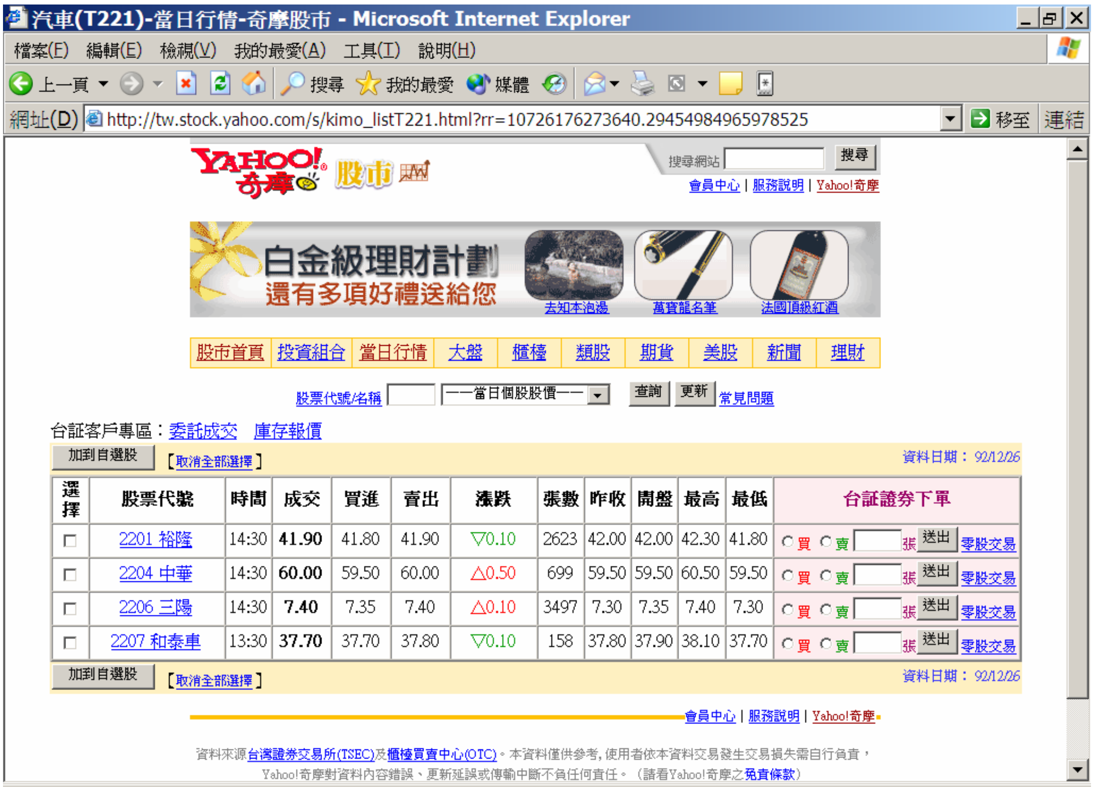
台股在美股休市氣氛中以平高盤開出，然而量能不足，短多主軸仍在內需導向的傳產之營建、建材個股中發揮，雖然電子股雖有止跌，但電子股3D股…《更多...》

更新時間：12-26 17:06

Figure 7.2: http://stock.yam.com.

If the user chooses to show all the stock values of the same type company on the web browser, for example the company about the automobile, the screen will be shown as Figure

7.3.



汽車(T221)-當日行情-奇摩股市 - Microsoft Internet Explorer

網址(D) http://tw.stock.yahoo.com/s/kimo_listT221.html?rr=10726176273640.29454984965978525

檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
移至 連結
YAHOO! 奇摩 股市
搜尋網站 搜尋
會員中心 | 服務說明 | Yahoo!奇摩
白金級理財計劃
還有多項好禮送給您
去知本泡湯 萬寶龍名筆 法國頂級紅酒
股市首頁 投資組合 當日行情 大盤 櫃檯 類股 期貨 美股 新聞 理財
股票代號/名稱 ——當日個股股價—— 查詢 更新 常見問題

台証客戶專區：委託成交　庫存報價

加到自選股 【取消全部選擇】 資料日期：92/12/26

選擇	股票代號	時間	成交	買進	賣出	漲跌	張數	昨收	開盤	最高	最低	台証證券下單
□	2201 裕隆	14:30	**41.90**	41.80	41.90	▽0.10	2623	42.00	42.00	42.30	41.80	○買 ○賣 張 送出 零股交易
□	2204 中華	14:30	**60.00**	59.50	60.00	△0.50	699	59.50	59.50	60.50	59.50	○買 ○賣 張 送出 零股交易
□	2206 三陽	14:30	**7.40**	7.35	7.40	△0.10	3497	7.30	7.35	7.40	7.30	○買 ○賣 張 送出 零股交易
□	2207 和泰車	13:30	**37.70**	37.70	37.80	▽0.10	158	37.80	37.90	38.10	37.70	○買 ○賣 張 送出 零股交易

加到自選股 【取消全部選擇】 資料日期：92/12/26

會員中心 | 服務說明 | Yahoo!奇摩

資料來源台灣證券交易所(TSEC)及櫃檯買賣中心(OTC)。本資料僅供參考，使用者依本資料交易發生交易損失需自行負責，
Yahoo!奇摩對資料內容錯誤、更新延誤或傳輸中斷不負任何責任。（請看Yahoo!奇摩之免責條款）

Figure 7.3: All stock list of the company about automobile.

In this example, we may easily to find that the querying string is shown as follows:

http://tw.stock.yahoo.com/s/kimo_listT221.html?rr=10726177510120.8847874848407946

Obviously, the following ‘rr=10726177510120.8847874848407946’ is the identification for advertisement, and is not irrelevant to the content of this time of search. Thus, the major querying string is shown as follows:

http://tw.stock.yahoo.com/s/kimo_listT221.html

Again, comparing the above querying string to the example for querying the stock value of first classification of electronic companies which is shown in the following:

http://tw.stock.yahoo.com/s/kimo_listT231.html

In this case, we may recognize that the URL address of the stock quote information will be http://tw.stock.yahoo.com, and ‘s/kimo_list’ will be the pre-string of the querying format, and then the ‘.html’ will be the post-string of the querying format. Furthermore, obviously, the T221 or T231 should be the classification number of the stock value of TAIEX market on the web site. Thus, for the mobile user, she may set these arguments (URL, pre-string, post-string) to the database of e-MARE in advance by WWW interface. Hereafter, she can compose the following subject of operational e-mail to query the specified classification of company stock value through her mobile device:

&&PWD(secret);&&AQUERYC(9,T221);

The result of the querying results is shown as the follows:

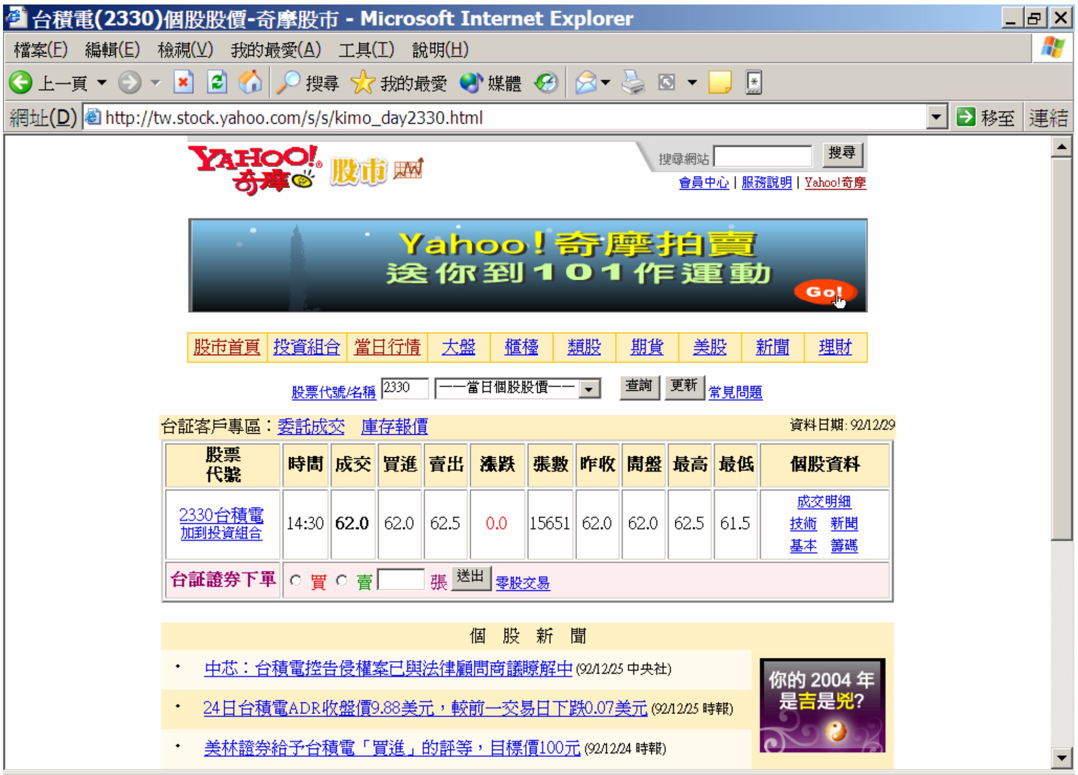
```
Yahoo!奇摩股市
搜尋網站
會갈中腄냉 A 務說明ㅣYahoo!奇摩
股市熄
投資組合
當日行情
大盤
櫃檯
類股
期貨
美股
新聞
理財
股票代號/名稱      常見問題
【取消全部選擇】
資料日期： 92/12/29
選擇
股票代號
時間
成交
買進
賣出
漲跌
張數
昨收
開盤
最高
最低
台証證券下單
2201 裕隆
14:30
41.40
41.30
41.40
▽0.40
5000
41.80
41.80
42.10
41.30
買 賣 張 零股交易
2204 中華
14:30
59.50
59.00
59.50
▽0.50
1363
60.00
59.50
60.00
59.00
買 賣 張 零股交易
2206 三陽
14:30
7.50
```

```
    7.50
    7.55
    △0.10
    12278
    7.40
    7.40
    7.60
    7.35
    買 賣 張 零股交易
    2207 和泰車
    13:30
    37.80
    37.80
    37.90
    △0.10
    243
    37.70
    37.70
    38.00
    37.60
    買 賣 張 零股交易
    【取消全部選擇】
    資料日期： 92/12/29
Service Time:504906
```

In the same example, the above situation is that when the user wants to browse many companies at a time. However, in many cases, there are lots of mobile users will only to query one or little stock value in the market. At the same web site, if the user has entered the stock id number 2330 into the web site, and press the 'query' button, the URL will be encoded as follows:

http://tw.stock.yahoo.com/s/s/kimo_day2330.html

Again, we may conclude that the URL address is always 'http://tw.stock.yahoo.com/', and the pre-string is 's/s/kimo_day', the post-string is '.html'. The screen after the user has entered the stock id is shown in Figure 7.4.


台積電(2330)個股股價-奇摩股市 - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
搜尋
我的最愛
媒體
網址(D)
移至
連結
YAHOO! 奇摩 股市
搜尋網站
搜尋
會員中心 | 服務說明 | Yahoo!奇摩
Yahoo!奇摩拍賣
送你到101作運動
Go!
股市首頁 投資組合 當日行情 大盤 櫃檯 類股 期貨 美股 新聞 理財
查詢
更新
常見問題
送出
你的 2004 年 是吉是兇?

http://tw.stock.yahoo.com/s/s/kimo_day2330.html

股票代號/名稱 2330 ——當日個股股價——

台証客戶專區：委託成交 庫存報價　　資料日期: 92/12/29

股票 代號	時間	成交	買進	賣出	漲跌	張數	昨收	開盤	最高	最低	個股資料
2330台積電 加到投資組合	14:30	**62.0**	62.0	62.5	0.0	15651	62.0	62.0	62.5	61.5	成交明細 技術 新聞 基本 籌碼
台証證券下單	○ 買 ○ 賣 □ 張 送出 零股交易										

個股新聞

- 中芯：台積電控告侵權案已與法律顧問商議瞭解中 (92/12/25 中央社)
- 24日台積電ADR收盤價9.88美元，較前一交易日下跌0.07美元 (92/12/25 時報)
- 美林證券給予台積電「買進」的評等，目標價100元 (92/12/24 時報)

Figure 7.4: The screen of querying the stock id 2330.

To quote the specific stock price via mobile device by e-MARE framework should compose the following command:

```
&&PWD(secret);&&AQUERYC(7,2330);
```

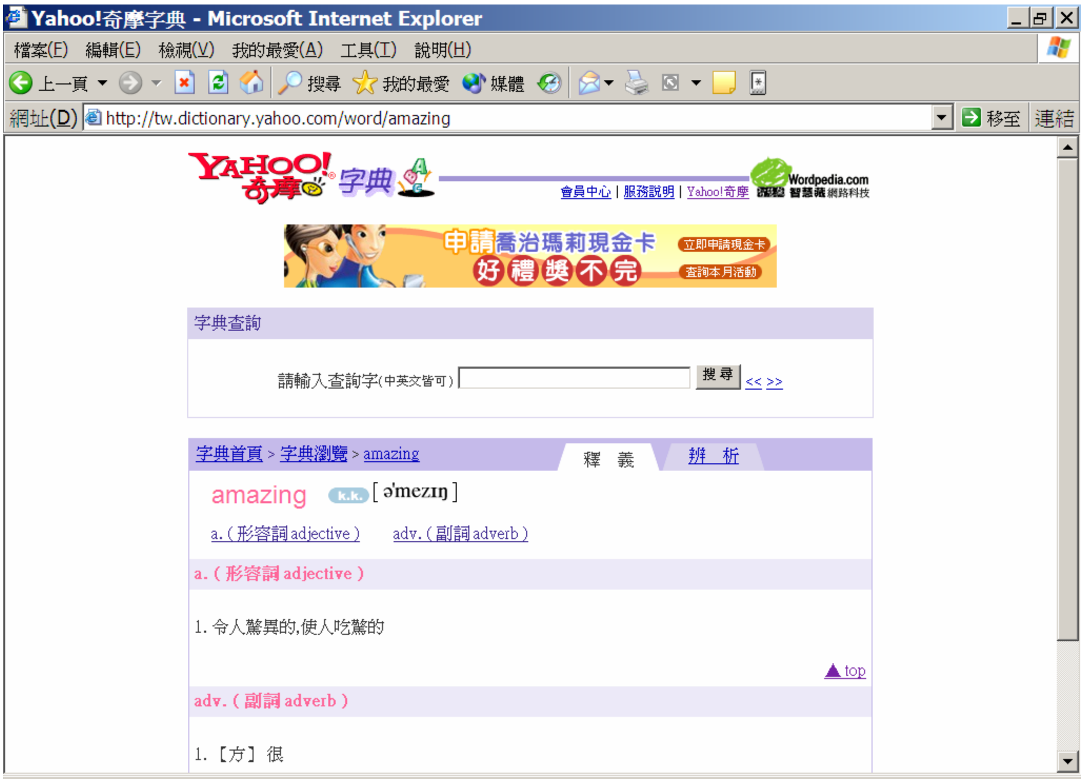
After the user has sent her operational e-mail, few minutes later, the results will be already in the user’s mailbox, and it is also shown as follows:

```
Yahoo!奇摩股市
搜尋網站
會갈中腄냈 A 務說明 l Yahoo!奇摩
股市熄
投資組合
當日行情
大盤
櫃檯
類股
期貨
美股
新聞
理財
股票代號/名稱　　常見問題
資料日期: 92/12/29
股票
代號
時間
成交
買進
賣出
漲跌
```

```
    張數
    昨收
    開盤
    最高
    最低
    莯捃禤 Æ
    2330 台積電
    加到投資組合
    14:30
    62.0
    62.0
    62.5
    0.0
    15651
    62.0
    62.0
    62.5
    61.5
    成交明細
    技術　新聞
    基本　籌碼
    台証證券下單
    買 賣 張 零股交易
    荂@股　新　聞
    中芯：台積電控告侵權案已與法律顧問商議瞭解中 (92/12/25 中央社)
    24 日台積電 ADR 收盤價 9.88 美元，較前一交易日下跌 0.07 美元 (92/12/25 時報)
    美林證券給予台積電「買進」的評等，目標價 100 元 (92/12/24 時報)
    23 日台積電 ADR 收盤價 9.95 美元，較前一交易日下跌 0.04 美元 (92/12/24 時報)
    外資奐 s 歸隊，台積電股價抗跌 (92/12/23 時報)
    中芯上市前夕，台積電告中芯侵權 (92/12/23 時報)
    更多歷史新聞
Service Time:514830
```

Although the above results have many unwanted information, but those documents of the various information sources can be read on the user's mobile device now, still less those unwanted information can also be filtered by some elaborate codes of programming language, such as the statement 's//' statement in Perl.

Except quoting the stock on the Internet, looking up new words on the Internet dictionary via mobile device is also a very easy work at e-MARE framework. In general, if a user has entered the work 'amazing' into the dictionary site (http://tw.dictionary.yahoo.com/) on the Internet, the screen will the same as Figure 7.5.



Yahoo!奇摩字典 - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http://tw.dictionary.yahoo.com/word/amazing
移至 連結
會員中心 | 服務說明 | Yahoo!奇摩
Wordpedia.com 智慧藏網路科技
申請喬治瑪莉現金卡 好禮獎不完
立即申請現金卡
查詢本月活動
搜 尋
<< >>
釋 義
辨 析
top

字典查詢

請輸入查詢字(中英文皆可)

字典首頁 > 字典瀏覽 > amazing

amazing [ə'mezɪŋ]

a. (形容詞 adjective) adv. (副詞 adverb)

a. (形容詞 adjective)

1. 令人驚異的,使人吃驚的

adv. (副詞 adverb)

1. 【方】 很

Figure 7.5: The screen of http://tw.dictionary.yahoo.com/.

Meanwhile, the URL was encoded as the follows:

http://tw.dictionary.yahoo.com/word/amazing

Obviously, the encoded string can be separated into the URL part 'http://tw.dictionary.yahoo.com/, and the pre-string part, 'word'. In this example, there is no post-string at all. The result through e-MARE for looking for the word 'amazing' is shown as follows:

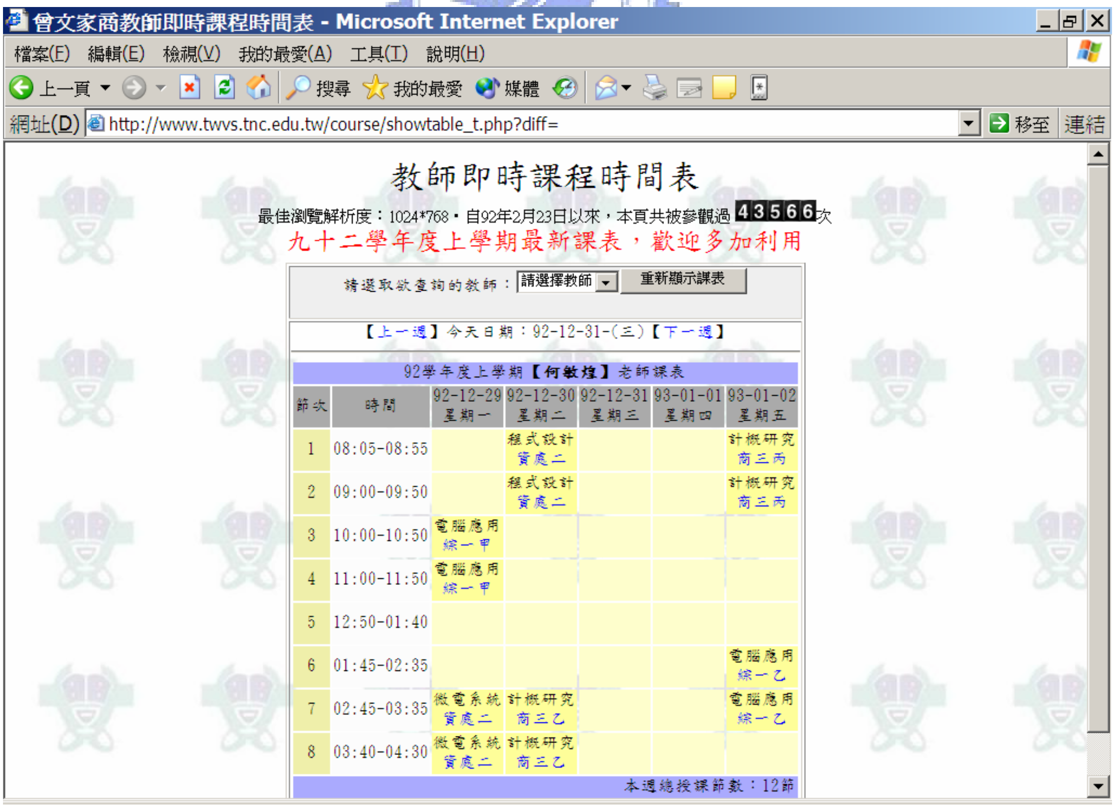
```
The fetching result for http://tw.dictionary.yahoo.com/word/amazing is shown as follows:
    Yahoo!奇摩字典
    會갈中腫냀 A 務說明 l Yahoo!奇摩
    字典查詢
    請輸入查詢字(中^文皆可) << >>
    字典熄 > 字典瀏覽 > amazing
    釋  義
    辨  析
      amazing
      a. ( 形容詞 adjective )   adv. ( 副詞 adverb )
    a. ( 形容詞 adjective )
    1. 令人驚異的,使人吃驚的
    ▲ top
    adv. ( 副詞 adverb )
    1. 【方】 很
    ▲ top
    會갈中腫냀 A 務說明 l Yahoo!奇摩
    雅虎國際資訊  版權所有 ?2003 Yahoo! Taiwan Inc. All Rights Reserved.
    隱私權政策 - 服務條款
Service Time:590536
```

Thus, utilizing e-MARE framework, through mobile device under unstable and expensive Internet connection, the user also may easily look for the new English word from the Internet.

7.2 Information retrieval from various information sources

Not only the information on the Internet are suitable for e-MARE framework, the other information resources are but also suitable for accessing through e-MARE framework. Different to the mechanism described in Section 7.1, the functions for fetching the information from the information resources other than Internet can not be used directly by the built-in commands, &&AQUERYC, &&AQUERYL, and &&AFW. Conversely, the user has to design her information retrieval application on the remote host, and activate those applications by the command ‘&&ARE’ of e-MARE framework. Thus, the user can fetch any information on the Internet or of the local area network. This section will describe such works in detail.

Suppose there is a web site which provides the class timetables for the student and teacher of high school. In general, the content of timetable is only readable for WWW browser, and it is shown in Figure 7.6.


曾文家商教師即時課程時間表 - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
搜尋
我的最愛
媒體
網址(D)
移至
連結

http://www.twvs.tnc.edu.tw/course/showtable_t.php?diff=

教師即時課程時間表

最佳瀏覽解析度：1024*768。自92年2月23日以來，本頁共被參觀過 43566次

九十二學年度上學期最新課表，歡迎多加利用

請選取欲查詢的教師：請選擇教師 ▾ 重新顯示課表

【上一週】今天日期：92-12-31-(三)【下一週】

92學年度上學期【何敏煌】老師課表

節次	時間	92-12-29 星期一	92-12-30 星期二	92-12-31 星期三	93-01-01 星期四	93-01-02 星期五
1	08:05-08:55		程式設計 資處二			計概研究 商三丙
2	09:00-09:50		程式設計 資處二			計概研究 商三丙
3	10:00-10:50	電腦應用 綜一甲				
4	11:00-11:50	電腦應用 綜一甲				
5	12:50-01:40					
6	01:45-02:35					電腦應用 綜一乙
7	02:45-03:35	微電系統 資處二	計概研究 商三乙			電腦應用 綜一乙
8	03:40-04:30	微電系統 資處二	計概研究 商三乙			

本週總授課節數：12節

Figure 7.6: An example of teacher’s timetable of high school.

As shown in Figure 7.6, the layout of timetable is a table, and it is not easy to display perfectly on limited display size of mobile device. Moreover, to perfectly transform the screen layout to pure text-version is very difficult. Fortunately, to this information source, there is more than one solution to fetch such timetable into the display of mobile device. If the user is only allowed to access the web interface, the user should write an application to analyze the layout of timetable, and transform it to mobile device readable layout. On the other side, if the user is allowed to directly access the original database of all timetables, she may write an application directly accessing the database, and fetching the desired data from the database. In the following example, we assume the latter case.

Suppose, the user has the right to access the database of course timetables. To view the teacher's or class's timetable via mobile device, at the first, the user has to write a program for fetching the related data from the course database (in this case, the database management is MySQL executed under Linux operating system). The Perl program, 'fetchTable', for fetching the related table is shown as follows:

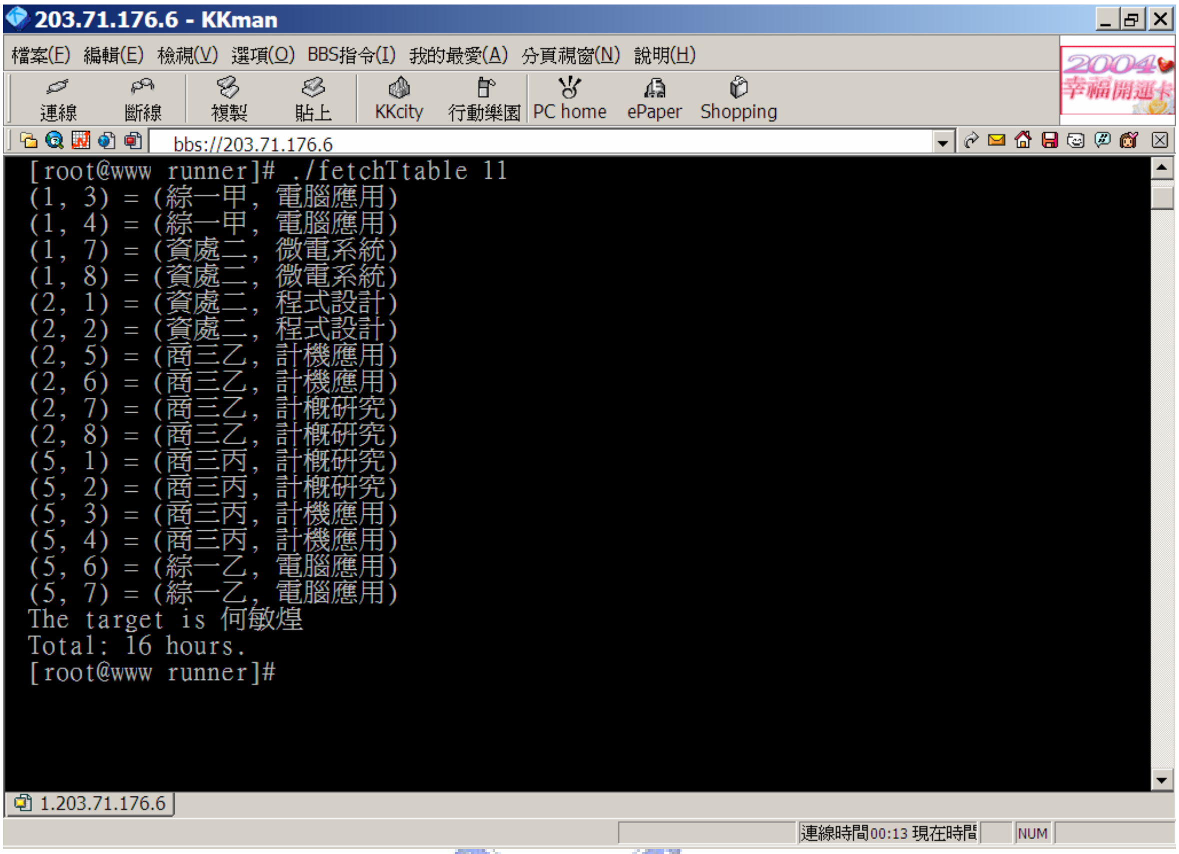
```
#!/usr/bin/perl
use DBI;

my $dsn          = 'DBI:mysql:92ACourse:203.71.176.6';
my $user         = '****';
my $password     = '****';
my $dbh;
my $sth;
my @ary;
my $techNo;
my $tlist;
my $clist;
my $slist;
my $ttable_c;
my $ttable_s;
my $rs;
my $qstr;

$techNo = $ARGV[0];

$dbh = DBI->connect($dsn, $user, $password, {RaiseError => 1});
$qstr = "select * from tlist order by techno";
$sth = $dbh->prepare($qstr);
$sth->execute();
while($rs = $sth->fetchrow_hashref())
{
   $tno = $$rs{'techno'};
   $tna = $$rs{'name'};
   if (length($tna)>0)
   {
      $tlist{"$tno"} = $tna;
   }
}
```

```
$qstr = "select * from clist order by crno";
$sth = $dbh->prepare($qstr);
$sth->execute();
while($rs = $sth->fetchrow_hashref())
{
   $crno = $$rs{'crno'};
   $crna = $$rs{'crname'};
   if (length($crna)>0)
   {
      $clist{"$crno"} = $crna;
   }
}

$qstr = "select * from slist order by subno";
$sth = $dbh->prepare($qstr);
$sth->execute();
while($rs = $sth->fetchrow_hashref())
{
   $subno = $$rs{'subno'};
   $subna = $$rs{'fullname'};
   if (length($subna)>0)
   {
      $slist{"$subno"} = $subna;
   }
}

$qstr = "select * from alltable where fulldate>='20030901' and fulldate<='20030905' " .
            "and techno=" . $techNo . " order by week, section";
$sth = $dbh->prepare($qstr);
$sth->execute();
$i=0;
while($rs = $sth->fetchrow_hashref())
{
   $i++;
   $tweek = $$rs{'week'};
   $tsection = $$rs{'section'};
   $tindex = $tweek . $tsection;
   $tcrno = $$rs{'crno'};
   $tsubno = $$rs{'subno'};

   $ttable_c{"$tindex"} = $tcrno;
   $ttable_s{"$tindex"} = $tsubno;
   $cc = $clist{$tcrno};
   $ss = $slist{$tsubno};
   print "($tweek, $tsection) = ($cc, $ss)\n";
}

$target = $tlist{$techNo};
print "The target is $target\n";
print "Total: $i hours.\n";
```

In ‘fetchTable’ program, it accepts the number of teacher in Tseng-Wen vocational high school (http://www.twvs.tnc.edu.tw), and generates the weekly course timetable for that specified teacher. For example, if we order the following command at the operating environment of *Mobile Agent Execution Server*:

./fetchTable 11

The program will generate the results as shown in Figure 7.7.



```
[root@www runner]# ./fetchTtable 11
(1, 3) = (綜一甲, 電腦應用)
(1, 4) = (綜一甲, 電腦應用)
(1, 7) = (資處二, 微電系統)
(1, 8) = (資處二, 微電系統)
(2, 1) = (資處二, 程式設計)
(2, 2) = (資處二, 程式設計)
(2, 5) = (商三乙, 計機應用)
(2, 6) = (商三乙, 計機應用)
(2, 7) = (商三乙, 計概研究)
(2, 8) = (商三乙, 計概研究)
(5, 1) = (商三丙, 計概研究)
(5, 2) = (商三丙, 計概研究)
(5, 3) = (商三丙, 計機應用)
(5, 4) = (商三丙, 計機應用)
(5, 6) = (綜一乙, 電腦應用)
(5, 7) = (綜一乙, 電腦應用)
The target is 何敏煌
Total: 16 hours.
[root@www runner]#
```

Figure 7.7: The executing results of command ‘fetchTable 11’.

Through the service of e-MARE, after the user has sent the operational e-mail subjected ‘&&ARE(4,11)’, the result will soon be at the user’s mailbox, and it is shown as follows:

```
The command executed result is shown as follows:
(1, 3) = (綜一甲, 電腦應用)
(1, 4) = (綜一甲, 電腦應用)
(1, 7) = (資處二, 微電系統)
(1, 8) = (資處二, 微電系統)
(2, 1) = (資處二, 程式設計)
(2, 2) = (資處二, 程式設計)
(2, 5) = (商三乙, 計機應用)
(2, 6) = (商三乙, 計機應用)
(2, 7) = (商三乙, 計概研究)
(2, 8) = (商三乙, 計概研究)
(5, 1) = (商三丙, 計概研究)
(5, 2) = (商三丙, 計概研究)
(5, 3) = (商三丙, 計機應用)
(5, 4) = (商三丙, 計機應用)
(5, 6) = (綜一乙, 電腦應用)
(5, 7) = (綜一乙, 電腦應用)
```

```
The target is 何敏煌
Total: 16 hours.
[runner@www runner]
the command is ./fetchTtable 11.
Service Time:841108
```

Applying the technology mentioned above, the mobile device now can be utilized to obtain all the information in the database management system though the appropriate designed program and e-MARE framework.

The other information sources which are applied in another protocol are just the same technology as the above approach. That is, in another aspect, although some information sources do not provide the HTTP protocol (such as Z39.50) and web service to serve the remote user, they always still connected on the network. Furthermore, many of them are connected to the Internet, and serve their client in some specific network port. By default, the commands of e-MARE for fetching the web documents are through the standard HTTP port, which is numbered 80. For those information sources do not provide the HTTP port, the built-in commands of e-MARE can be applied for information retrieval of mobile users. In such case, to design or to install the specific information application into the *Mobile Agent Execution Server* of e-MARE is a feasible alternative solution.

7.3 Remote displaying via e-MARE framework

To be a mobile information carrier, the most important defect is the size of screen. The limited size screen of mobile device can not show large enough information at once. The user often scrolls the screen while she is reading documents, especially for the document that is not designed for mobile devices. Fortunately, because the e-MARE framework can initialize the remote application by operational e-mail through mobile device, the user can expand the mobile device's display ability by implementing remote displaying, printing, and remote FAX. Following will explain the detail implementation.

Remote displaying technology is a very easy but useful application for the limited size mobile device to expand its displaying ability. The basic idea and architecture is shown in Figure 7.8. As shown in Figure 7.8, from the viewpoint of the user, she may compose an operational e-mail, which is combined with the displaying commands in the subject of that e-mail. Those commands will control the contents of *Public Display Point (PDP)* through the e-MARE server. In general, the related information about the contents of PDP and the controller is stored in the blackboard server. The e-MARE server will access that server to find the most suitable information for displaying at PDP. That is, the user may modify the information of the blackboard system by e-MARE’s commands, and the PDP will generate the web contents according to the contents of the data table of the blackboard system.

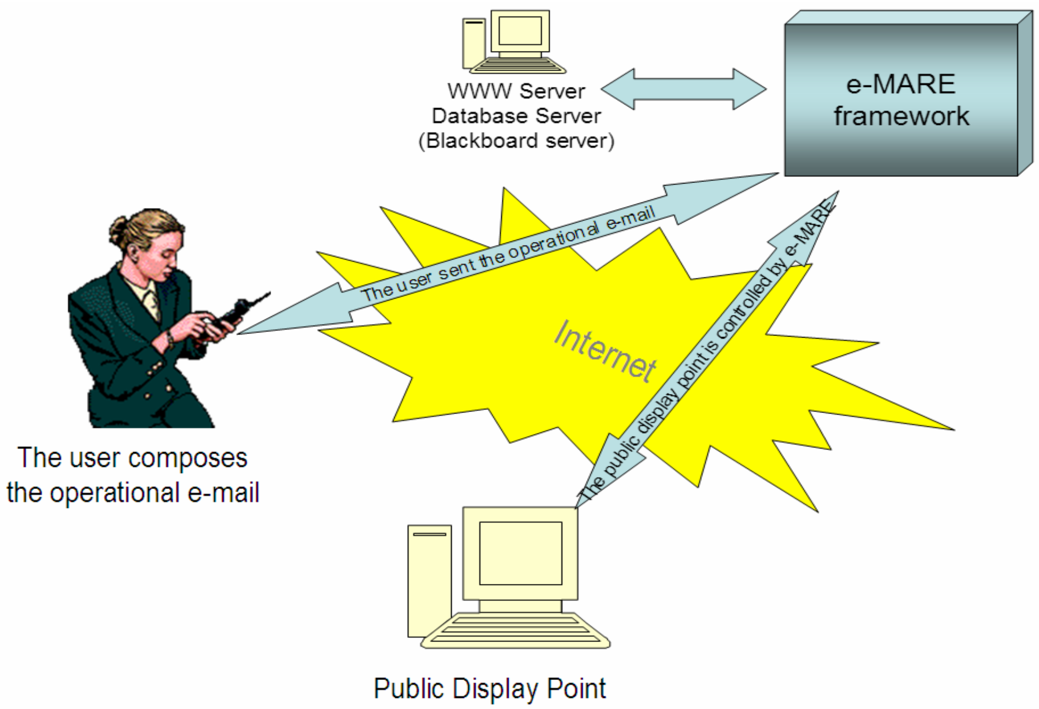
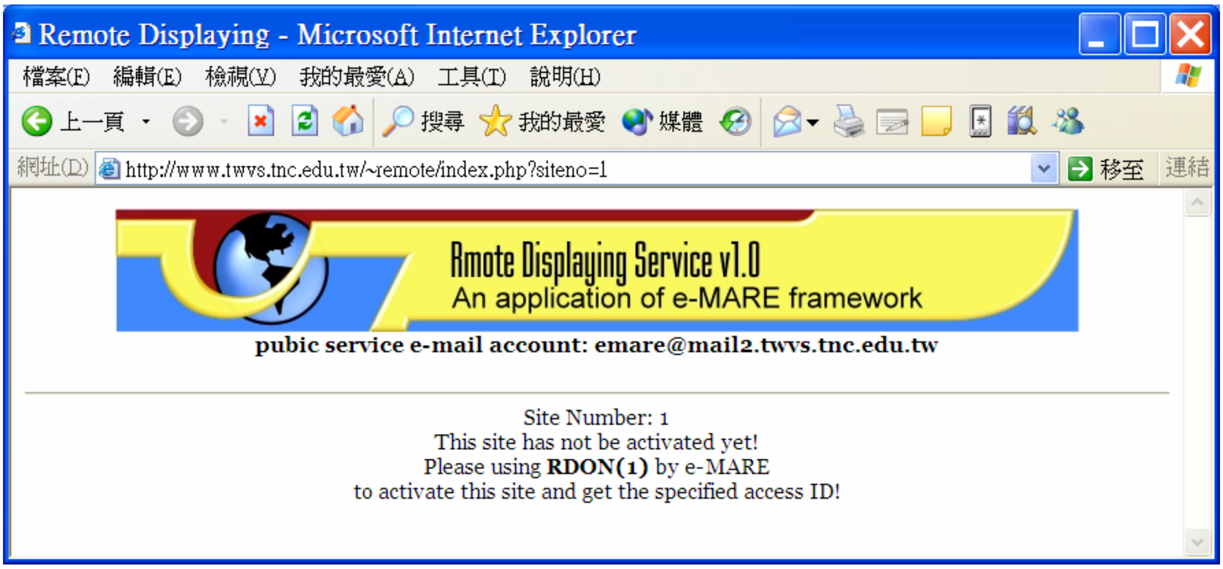

WWW Server
Database Server
(Blackboard server)
e-MARE framework
The user sent the operational e-mail
Internet
The public display point is controlled by e-MARE
The user composes the operational e-mail
Public Display Point

Figure 7.8: The basic architecture of remote displaying technology.

As the earlier explanation and from the viewpoint of the server, PDP is the client of service, and it is merely a web browser which is browsing a specified web page of the server web site. The web page of server web site is controlled by a manager program, and the program will adjust the contents of web page according to the specified data item in e-MARE blackboard system. Because the specified data items are controlled by mobile user via e-MARE, the user may indirectly control the PDP via her mobile device.

For example, in this section, we construct a dedicated web site for remote displaying application which is addressed at http://www.twvs.tnc.edu.tw/~remote. To activate a PDP by that URL address has to be encoded with the site number, siteno. The homepage of http://www.twvs.tnc.edu.tw/~remote/index.php?siteno=1 is shown in Figure 7.9.

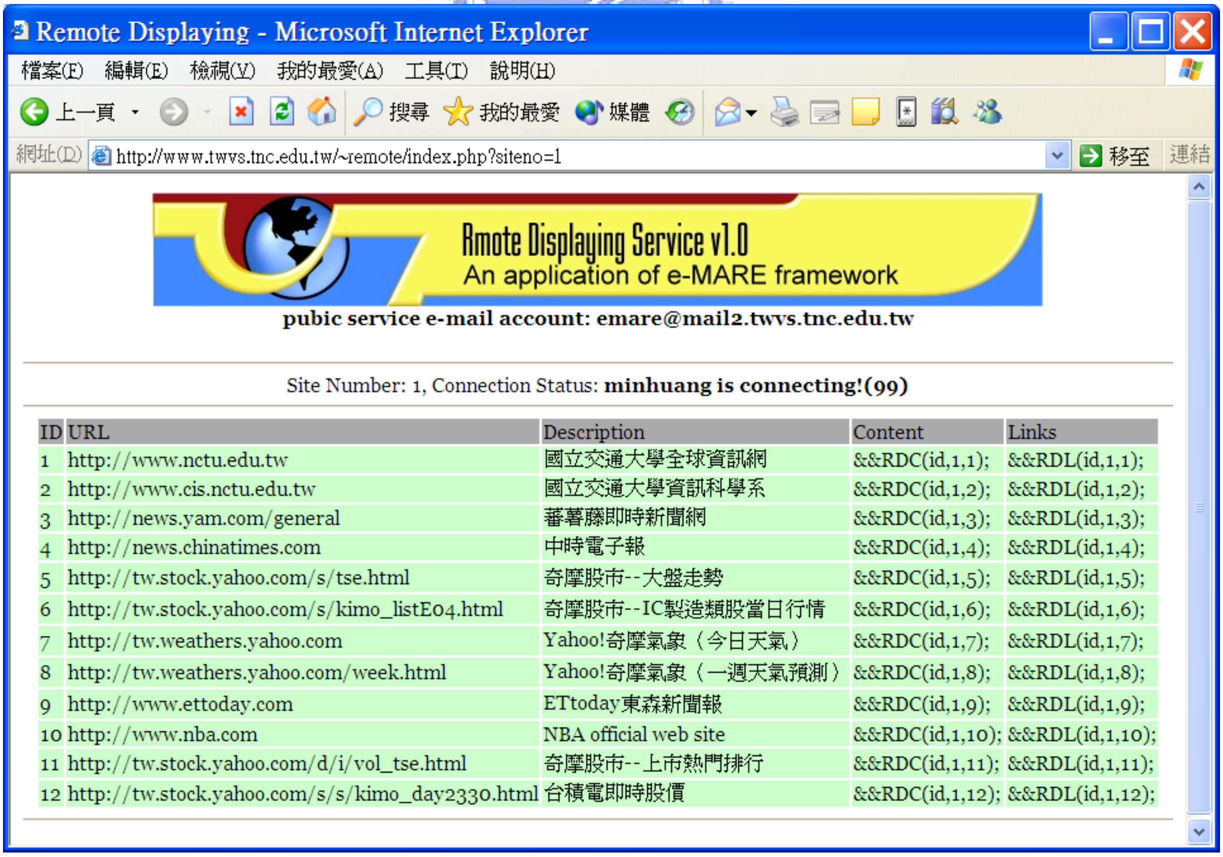

Remote Displaying - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http://www.twvs.tnc.edu.tw/~remote/index.php?siteno=1
移至 連結
Rmote Displaying Service v1.0
An application of e-MARE framework

pubic service e-mail account: emare@mail2.twvs.tnc.edu.tw

Site Number: 1
This site has not be activated yet!
Please using **RDON(1)** by e-MARE
to activate this site and get the specified access ID!

Figure 7.9: The main webpage of http://www.twvs.tnc.edu.tw/remote.

As shown in Figure 7.9, because this is our first time to access the PDP #1, there are some messages shown in the web page remaining us to construct this PDP by RDON(1) command through e-MARE. After the user's activated this PDP by RDON(*no*), the homepage of this PDP will soon be changed as shown in Figure 7.10.


Remote Displaying - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http://www.twvs.tnc.edu.tw/~remote/index.php?siteno=1
移至 連結
Rmote Displaying Service v1.0
An application of e-MARE framework

pubic service e-mail account: emare@mail2.twvs.tnc.edu.tw

Site Number: 1, Connection Status: **minhuang is connecting!(99)**

ID	URL	Description	Content	Links
1	http://www.nctu.edu.tw	國立交通大學全球資訊網	&&RDC(id,1,1);	&&RDL(id,1,1);
2	http://www.cis.nctu.edu.tw	國立交通大學資訊科學系	&&RDC(id,1,2);	&&RDL(id,1,2);
3	http://news.yam.com/general	蕃薯藤即時新聞網	&&RDC(id,1,3);	&&RDL(id,1,3);
4	http://news.chinatimes.com	中時電子報	&&RDC(id,1,4);	&&RDL(id,1,4);
5	http://tw.stock.yahoo.com/s/tse.html	奇摩股市--大盤走勢	&&RDC(id,1,5);	&&RDL(id,1,5);
6	http://tw.stock.yahoo.com/s/kimo_listE04.html	奇摩股市--IC製造類股當日行情	&&RDC(id,1,6);	&&RDL(id,1,6);
7	http://tw.weathers.yahoo.com	Yahoo!奇摩氣象（今日天氣）	&&RDC(id,1,7);	&&RDL(id,1,7);
8	http://tw.weathers.yahoo.com/week.html	Yahoo!奇摩氣象（一週天氣預測）	&&RDC(id,1,8);	&&RDL(id,1,8);
9	http://www.ettoday.com	ETtoday東森新聞報	&&RDC(id,1,9);	&&RDL(id,1,9);
10	http://www.nba.com	NBA official web site	&&RDC(id,1,10);	&&RDL(id,1,10);
11	http://tw.stock.yahoo.com/d/i/vol_tse.html	奇摩股市--上市熱門排行	&&RDC(id,1,11);	&&RDL(id,1,11);
12	http://tw.stock.yahoo.com/s/s/kimo_day2330.html	台積電即時股價	&&RDC(id,1,12);	&&RDL(id,1,12);

Figure 7.10: The screen snapshot after the system has received the command, &&RDON(1).

As shown in Figure 7.10, each web page of remote displaying has its own serial number which is called *Site Number*. In this example, its *Site Number* is 1. This number is specified by the PDP while the browser has been connected to http://www.twvs.tnc.edu.tw/~remote. If the user wants to browse the contents of listed web site, she may just issue the &&RDC(id,1,*no*) command to the service account of e-MARE, if the user wants to view all links of the specified URL, the &&RDL is instead of &&RDC. For instance, if the user wants to browse the web page of http://news.yam.com/general, the user has to issue the command, &&RDC(id,1,3), in this PDP. Conversely, if the user wants to browse all links of the web page of http://tw.stock.yahoo.com/s/kimo_listE04.html and further dig those links on that page more depth, the command, &&RDL(id,1,3) is much preferred than &&RDC(id,1,3).

Meanwhile, the remote site serial number is started at 1, and the 0 is the page browsing commands. The mean of these commands are shown in Table 7.1.

Table 7.1: The mean of &&RDC(id, 0, n) commands.

Commands	Description
&&RDC(id,0,0);	Return to the homepage of http://www.twvs.tnc.edu.tw/~remote.
&&RDC(id,0,1);	Scroll the screen to the top of the webpage.
&&RDC(id,0,2);	Scroll Page Up.
&&RDC(id,0,3);	Scroll Page Down
&&RDC(id,0,4);	Scroll up one line.
&&RDC(id,0,5);	Scroll down one line.

Each *PDP* has a unique site number. If a computer want to be a PDP, the only thing has to do is just to connect the specified URL address with the unique site number, which is shown as follows:

http://www.twvs.tnc.edu.tw/~remote/index.php?siteno=1

Because the HTML document of the mentioned URL is passive and has not be applied any other active client web technology (such as Java Applet or ActiveX), the page has to refresh itself regularly for updating the user's requests. In our implementation, the document will refresh itself every 60 seconds. Each time while the page is refreshing, the program of web server will checks the actSite table of the database in the blackboard server to see if the status is changed or not. Then, the appropriate user-defined HTML document will be generated for the specified number of remote displaying site.

To activate the remote displaying function for the mobile user through e-MARE, the user has to initiate the command &&RDON(*siteno*). After *managerAgent* of e-MARE has received the operational e-mail with that command, it will look up the user's information from the user profile database, and check the user's accessing right. If the user has the right to access the remote site, and the specified remote site has not registered yet, that specified site will be assigned to the mobile user, and give a time quota for the user' accessing process. This value (the default value is 100) will be stored at the TimeToLive field of actSite table. Each time while the web page is being refreshed, the value of TimeToLive will be subtracted to 1. If finally the value of TimeToLive be subtracted to zero, the user's related information will be wiped out from the actSite table of blackboard system. However, if the user's has been to alternate any information in actSite table, the value of TimeToLive will be restored into default value. It means the user will enforce to be disconnected if she has not accessing her information on the data table for 20 minutes.

To achieve the earlier functionalities, the manager of system will maintain a table, called actSite, in blackboard system's database, which is shown in Table 7.2.

Table 7.2: The table structure of actSite.

no	Siteno	Status	TimeToLive	showLinks	user	Id
Integer	Integer	Integer	Integer	Integer	String	Integer

As shown in Table 7.2, the field 'no' is the serial number of all site number, and the 'siteno' is for identifying these data fields is for what site. The field 'status' is for recording the current state of the site, and 'TimeToLive' is a time period to see if the access ability of user is still valid or not. The field 'showLinks' is the direction to the web page to redirect to the appropriate web link. Finally, the field 'user' is the user's account, and the 'id' is random number for each access phase of the user.

After the user has submitted the command, &&RDON(1), to e-MARE via the operational e-mail, the web page of site number 1 will be changed as soon as possible. The returned e-mail is shown as follows:

```
Activating Public Display Point
Username: minhuang
SiteNo: 1
Status: Successful.
ID: 9469170
Service Time:0.102436065673828
```

It means that the identification number is 9469170, and this number is the token for the user to further access the web document. However, if the remote web site is being handled by someone else, the returned e-mail will tell the user that this remote web site is unavailable. In such case, the contents of the return e-mail will just like the contents of following:

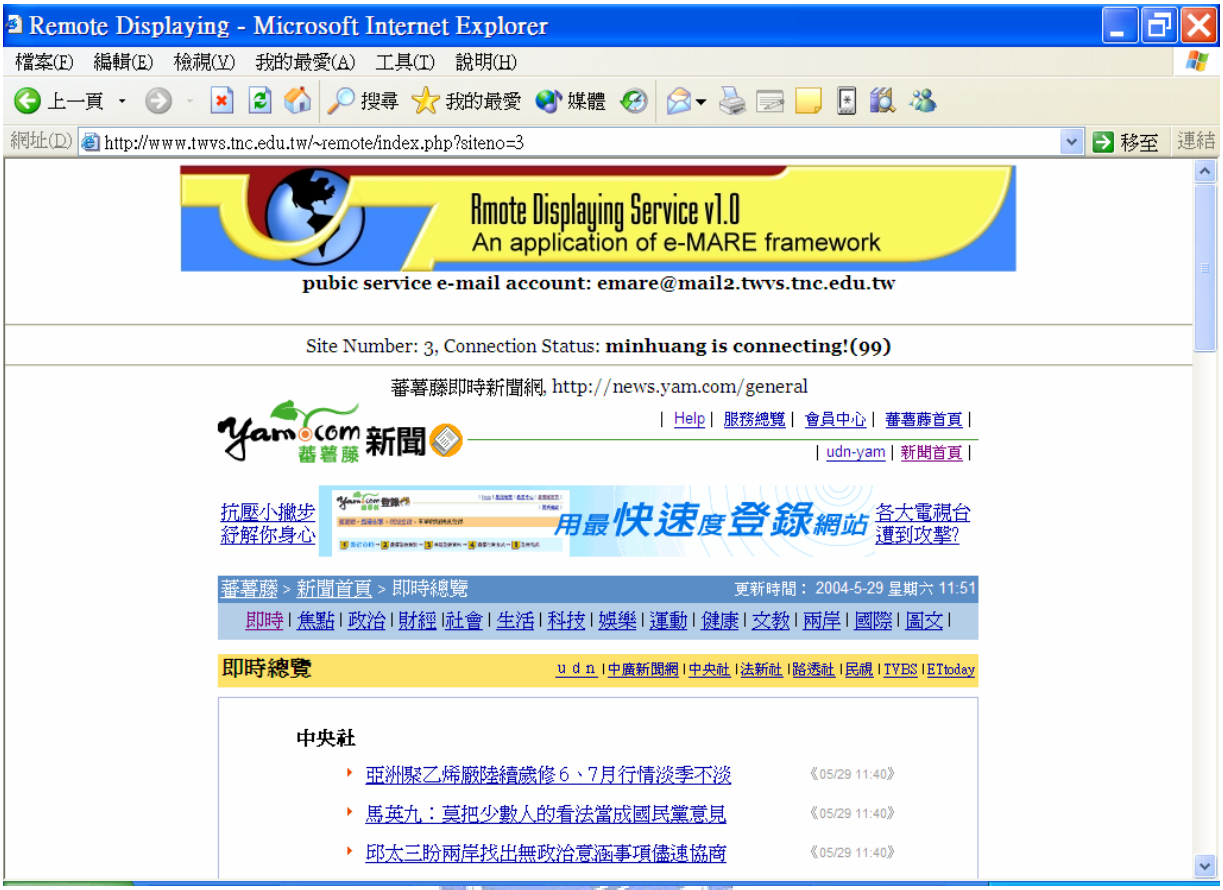
```
Activating Public Display Point
Username: minhuang
SiteNo: 1
Status: The site number is already activated by others.
Service Time:0.0305600166320801
```

If the user has successfully grabbed the right for controlling one of the remote displaying sites, the screen snapshot will be just like the screen which is shown in Figure 7.10.

As shown in Figure 7.10, the web page now is showing the connected user’s information and the value of TimeToLive. Meanwhile, the system will return an identification number to the user. The identification number is a unique number for the verification between the user and the system in the latter accessing works. Only the user who provides the identification number can further interact with the web page of the PDP. For example, in this case, if the user has composed the following commands after she has registered the remote displaying site no.1:

```
&&RDC(9469170,1,3);
```

The screen of PDP siteno 1 will soon be changed into the screen which is shown in Figure 7.11.


Remote Displaying - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
搜尋
我的最愛
媒體
網址(D)
移至
連結

http://www.twvs.tnc.edu.tw/~remote/index.php?siteno=3

Rmote Displaying Service v1.0
An application of e-MARE framework

pubic service e-mail account: emare@mail2.twvs.tnc.edu.tw

Site Number: 3, Connection Status: minhuang is connecting!(99)

蕃薯藤即時新聞網, http://news.yam.com/general

Yam.com 蕃薯藤 新聞
| Help | 服務總覽 | 會員中心 | 蕃薯藤首頁 |
| udn-yam | 新聞首頁 |
抗壓小撇步 紓解你身心
用最快速度登錄網站
各大電視台 遭到攻擊?

蕃薯藤 > 新聞首頁 > 即時總覽 更新時間： 2004-5-29 星期六 11:51

即時 | 焦點 | 政治 | 財經 | 社會 | 生活 | 科技 | 娛樂 | 運動 | 健康 | 文教 | 兩岸 | 國際 | 圖文 |

即時總覽

u d n | 中廣新聞網 | 中央社 | 法新社 | 路透社 | 民視 | TVBS | ETtoday

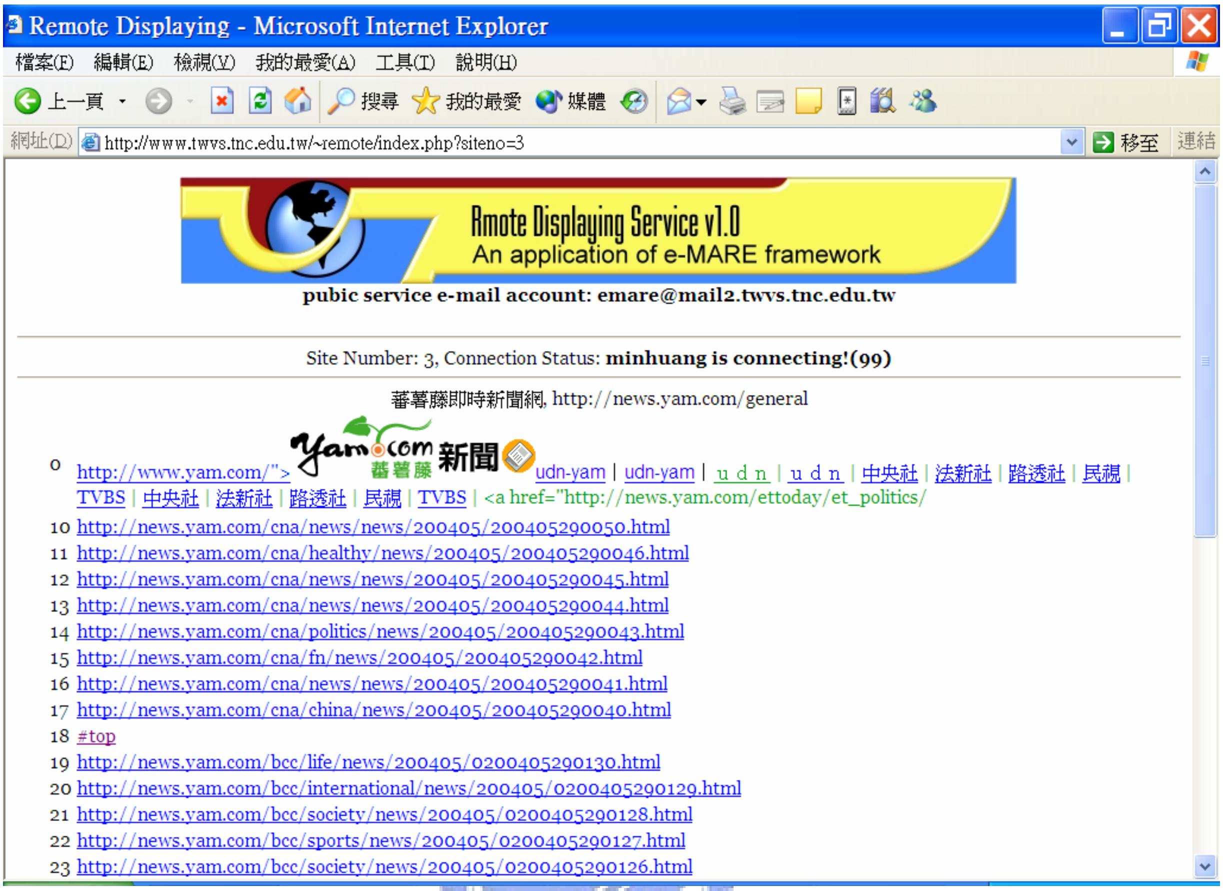
中央社

- 亞洲聚乙烯廠陸續歲修６、7月行情淡季不淡 《05/29 11:40》
- 馬英九：莫把少數人的看法當成國民黨意見 《05/29 11:40》
- 邱太三盼兩岸找出無政治意涵事項儘速協商 《05/29 11:40》

Figure 7.11: The screen snapshot after the &&RDC commands has issued.

1896

Now, the user may utilize the &&RDC(id, 0, *no*) series commands to further control the position of web documents. Of course, if the use has no any actions to this remote displaying site for a long time, the remote displaying site will soon be changed to the screen which is shown in Figure 7.9. If the user has submitted the &&RDL(id, 1, *no*) series commands, the web page of remote displaying site will like the screen which is shown in Figure 7.12.


Remote Displaying - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
搜尋
我的最愛
媒體
網址(D)
移至
連結
Rmote Displaying Service v1.0
An application of e-MARE framework

http://www.twvs.tnc.edu.tw/~remote/index.php?siteno=3

pubic service e-mail account: emare@mail2.twvs.tnc.edu.tw

Site Number: 3, Connection Status: **minhuang is connecting!(99)**

蕃薯藤即時新聞網, http://news.yam.com/general

0 http://www.yam.com/"> Yam.com 蕃薯藤 新聞 udn-yam | udn-yam | u d n | u d n | 中央社 | 法新社 | 路透社 | 民視 | TVBS | 中央社 | 法新社 | 路透社 | 民視 | TVBS | <a href="http://news.yam.com/ettoday/et_politics/
10 http://news.yam.com/cna/news/news/200405/200405290050.html
11 http://news.yam.com/cna/healthy/news/200405/200405290046.html
12 http://news.yam.com/cna/news/news/200405/200405290045.html
13 http://news.yam.com/cna/news/news/200405/200405290044.html
14 http://news.yam.com/cna/politics/news/200405/200405290043.html
15 http://news.yam.com/cna/fn/news/200405/200405290042.html
16 http://news.yam.com/cna/news/news/200405/200405290041.html
17 http://news.yam.com/cna/china/news/200405/200405290040.html
18 #top
19 http://news.yam.com/bcc/life/news/200405/0200405290130.html
20 http://news.yam.com/bcc/international/news/200405/0200405290129.html
21 http://news.yam.com/bcc/society/news/200405/0200405290128.html
22 http://news.yam.com/bcc/sports/news/200405/0200405290127.html
23 http://news.yam.com/bcc/society/news/200405/0200405290126.html

Figure 7.12: Utilizing the &&RDL commands to show the URL list in PDP.

The application of Public Displaying mechanism will provide the mobile to control the public display point by their mobile devices. Because the commands for accessing PDP only need three simple commands, &&RDON, &&RDC, &&RDL, even the very low level GSM cellular phone can be used in this mechanism.

Chapter 8

Conclusions

The major disadvantages of mobile devices to be the information retrieval platform is the limitation of display size, low computing ability, limited power, and the unstable and expensive network connection. To provide the ability and maximum flexibility for the mobile user to directly access the Internet and private database on the LAN through their mobile device, moving most of the works to the back-end server is a very important idea. In this dissertation, we enhance and expand this idea, and propose the e-MARE framework to maximize the mobile user's flexibility.

There are some products and services letting the user to access the information on the Internet through the mobile devices in the market, but for now, these products and services are just providing some simple and poor interface for the user to operate their remote information. There is no a whole and complete solution for versatile heterogeneous mobile devices. We compare those products or researches to our e-MARE framework at Table 8.1.

Table 8.1: The comparison between e-MARE and other related works.

Items	Our Approach	Info-On-Demand	WWW4mail	Mobile Hotmail	iSMS	AvantGo
Supported Platforms	All	All	Desktop PC	Cellular Phone	Cellular Phone	Cellular Phone and PDA
Information Sources	All	Private Database	WWW	Personal Mailbox	All	WWW
Execute Remote Programs	Yes	Simple Batch Program	No	No	Yes	No
Interactive Flexibility	Yes	No	No	Partial	Partial	No
Main Access Media	E-mail and All	E-mail	E-mail	SMS	SMS	Sync Software
Real-Time Access	Yes	Yes	Yes	Yes	Yes	No
Cost	Low	Low	Moderate	Very Expensive	Expensive	Expensive

As we can see in Table 8.1, the e-MARE framework is a very complete solution about the related issues which were mentioned earlier. Via e-MARE, a mobile user can not only send the simple commands to operate the remote mobile agents, but also write his own mobile agents to the server by his limited mobile device. Because the framework provides the mobile user maximum flexibility to access their information through the mobile device, we expect the framework will be the easy and powerful tool for the mobile users to retrieve his interested information on the Internet or private database on the local area network.

Utilizing the instructions provided by e-MARE, the user can easily operate the mobile information retrieval agent embedded in the main server to retrieve the interested information from the Internet or private databases via her mobile devices. Not just the embedded mobile agent, the user can design her own mobile agent at their mobile device, and then launch that agent to the remote host on her own purpose.

Our proposed e-MARE has many advantages. Firstly, for the mobile user, the networking connection will be required only at the time while the e-mail was being sending and receiving. The networking connection doesn’t need when the user is designing the mobile agent or reading the results. Thus, our mechanism will greatly reduce the cost of networking connection for mobile users. Secondly, the e-mail application is the most popular application for networked devices. It is an easy and fast implementation for any devices. Moreover, the user interface of e-mail application is simple and similar, thus, e-MARE is very easy to be deployed into any mobile devices. Finally, the operational e-mail can be treated as a normal e-mail message. It can be read, replied, or even forwarded to other's mailbox. Then, the system can achieve the characteristics of reusability, cross-platform, and simplicity. Of course, all of the works can be done at desktop PC or notebook too.

In conclusions, the main contribution of this dissertation is to propose an e-mail-based mobile agent runtime environment, and implements a prototype system named e-MARE. Through the mechanism, no matter what the device is, and no matter where the user at, the user can use the unified user interface to design, launch, and operate the mobile agent to achieve her own purpose. Because of the off-line characteristic of e-mail protocol, the mobile agent has more time to retrieve and analyze the documents fetched from the network and the Internet. The system may prepare some results with good quality as an e-mail, and then send it to user's mailbox. Under this framework, the user can do anything just as she is using the desktop PC with good networking environment, even though she is just using a mobile device with poor networking environment and poor computation power.

References

[1]. Baeza-Yates Ricardo and Ribeiro-Neto Berthier, “Modern Information Retrieval”, Addison Wesley, 1999.

[2]. Min-Huang Ho, Ming-Chun Cheng, Yue-Shan Chang and Shyan-Ming Yuan, “A GA-Based Dynamic Personalized Filtering For Internet Search Service on Multi-Search Engine”, IEEE Canadian Conference On Electrical And Computer Engineering - CCECE 2001, Canada 2001.

[3]. Min-Huang Ho, Yue-Shan Chang and Shyan-Ming Yuan, “Multi-Agent based Teacher-Cooperated Course Scheduling”, IASTED International Conference On Intelligent Systems and Control - ISC 2001, Nov. 19-22, 2001, pp. 232-237, Tampa, USA.

[4]. Min-Huang Ho, Yue-Shan Chang, Wen-Chen Sun and Shyan-Ming Yuan, “A Wrapper Generator for Integrated Information Retrieval”, CS&I'2002 Sixth International Conference On Computer Science and Informatics - CS&I'2002, March 8-12, 2002, pp. 367-371, Durham, NC, USA.

[5]. Min-Huang Ho, Yue-Shan Chang, Shyan-Ming Yuan and Winston Lo, “An Agent-Based Personalized Search on a Multi-Search Engine Based on Internet Search Service”, Intelligent Data Engineering and Automated Learning - IDEAL 2000, pp. 404-409, Hong Kong.

[6]. Gerhard Weiss, “Multiagent Systems”, The MIT Press, 1999.

[7]. Danny B. Lange, Mitsuru Oshima, “Programming and Deploying Java Mobile Agents with Aglets”, Addison-Wesley, Sept., 1998.

[8]. William R. Cockayne, Michael Zyda, “Mobile Agents”, Prentice Hall, Jan. 1998.

[9]. Yue-Shan Chang and Shyan-Ming Yuan, “A Framework for Integrating Information Retrieval on CORBA”, Proceeding of the 12th International Workshop on Database and Expert Systems Applications, pp. 180-184. Sept., 2001.

[10]. Maria Esther Vidal, Louiqa Raschid, Jean-Robert Gruser, “A Meta-Wrapper for Scaling up to Multiple Autonomous Distributed Information Sources”, Proceedings of the Third International Conference of Cooperative Information Systems, pp. 148-157, August 20-22, 1998 New York.

[11]. Hector Garcia-Molina, Yannis Papakonstantinou, Dallan Quass, Anand Rajaraman, Yehoshua Sagiv, Jeffery Ullman, Vasilis Vassalos, Jennifer Widom, “The TSIMMIS approach to mediation: data models and languages”, Journal of Intelligent Information System, pp. 117-132, 8 (2), March-April, 1997.

[12]. Hector Garcia-Molina, Joachim Hammer, Kelly Ireland, Yannis Papakonstantinou, Jerey Ullman, Jennifer Widom, “Integrating and Accessing heterogeneous information sources in TSIMMIS”, Proceeding of the AAAI Symposium on Information Gathering, pp. 61-64, Stanford, California, 1995.

[13]. Gerald Huck, Peter Frankhouser, Karl Aberer, Enich J. Neuhold, “Jedi: extracting and synthesizing information form the Web”, Proceedings. 3rd IFCIS International Conference on Cooperative Information Systems, pp. 32-41, 1998.

[14]. Naveen Ashish and Craig Knoblock, “Semi-Automatic Wrapper Generation for Internet Information Sources”, Proceedings of the Second IFCIS International Conference on Cooperative Information Systems, COOPIS ’97, pp. 160-169, 1997.

[15]. Alberto O and Mendelzon Gerorge A., “Querying the World Wide Web”, Fourth International Conference on Parallel and Distributed Information Systems, pp. 80-91, December 18-20, 1996.

[16]. David Mattox, Len Seligman, Ken Smith, “Rapper: A Wrapper Generator with Linguistic Knowledge”, Proceedings of the second international workshop on Web information and data management, pp. 6-11, November 02-06, Kansas City, Missouri, USA, 1999.

[17]. Ling Liu, Wei Han, David Buttler, Calton Pu, Wei Tang, “An XML-based Wrapper for Web Information Extraction”, proceedings of SIGMOD 1999, pp. 540-543, PA, Philadelphia, USA, 1999.

[18]. Yue-Shan Chang, Min-Huang Ho, Wen-Chun Sun, and Shyan-Ming Yuan, “A unified interface for integrating information retrieval”, Journal of Computer Standards & Interfaces, Vol. 23, No. 4, pp. 325-340, 2001.

[19]. Yue-Shan Chang, Shyan-Ming Yuan, Winston Lo, “A New Multi-Search Engine for Querying Data through Internet Search Service on CORBA”, Internal Journal of Computer Networks, Vol.34, No.3, pp. 467-480, Sept. 2000.

[20]. Ling Liu, Calton Pu, Wei Han, “XWRAP: An XML-enabled Wrapper Construction System for Web Information Sources”, 16th International Conference on Data Engineering (ICDE'2000), pp.611-621, 2000.

[21]. Arnaud Sahuguet, and Fabien Azavant, “Building Intelligent Web Applications Using Lightweight Wrappers”, Data and Knowledge Engineering, vol. 36, no. 3, pp.283-316, 2001.

[22]. Erik Selberg and Oren Etzioni, “The MetaCrawler Architecture for Resource Aggregation on the Web”, IEEE Expert, pp.11-14, Jan.-Feb. 1997

[23]. D. Dreilinger, A. E. Howe, “Experiences with Selecting Search Engines Using Metasearch”, ACM Transaction on Information System, Vol.15, No.3, pp. 195-222, July 1997.

[24]. Object Management Group, Inc. The Common Object request Broker (CORBA): Architecture and Specification, v2.0, November 1995.

[25]. Chia-Hui Chang, Ching-Chi Hsu, “Enabling Concept-Based Relevance Feedback for Information Retrieval on the WWW”, IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 4, pp. 595-609, July/August 1999.

[26]. Alexandros Moukas and Pattie Maes, “Amalthaea: An Evolving Multi-Agent Information Filtering and Discovery System for the WWW”, Autonomous Agents and Multi-Agent System, Vol. 1, no. 1, pp. 59-88, 1998.

[27]. Liliana Ardissono, C. Barbero, Anna Goy, Giovanna Petrone, “An Agent Architecture for Personalized Web Stores”, Proceedings of AGENT 1999, pp. 182-189, 1999.

[28]. William Wei Song, David Wai-Lok Cheung, and C. Tan, “A Semantic Similarity Approach to Electronic Document Modeling and Integration”, Web Information Systems Engineering, 2000. Proceedings of the First International Conference on Vol.1, pp. 116-124, 2000.

[29]. T. G. Rose and P. J. Wyard, “An Intelligent WWW Agent for Similarity-Based Searching”, IEE Colloquium, pp. 7/1-7/6, 1997.

[30]. Jaya Sil, Amit Konar, “Building Perception for Scheduling and Executing a Task Using Multi-Agent Systems, Proceedings of International Conference on Information Technology: Coding and Computing, pp. 675-679, 2001.

[31]. Sachiyo Arai, Katia Sycara, “Multi-Agent reinforcement Learning for Planning and Scheduling Multiple Goals”, Proceedings of 4th International Conference on MultiAgent Systems, pp. 359-360, 2000.

[32]. Mark Burstein, George Ferguson, and James Allen, “Integrating Agent-based Mixed-Initiative Control with

an Existing Multi-Agent Planning System", Proceedings of 4th International Conference on MultiAgent Systems, pp. 389-390, 2000.

[33]. Toramatsu Shintani, Takayuki Ito, and Katia Sycara, "Multiple Negotiations Among Agents for a Distributed Meeting Scheduler", Proceedings of 4th International Conference on MultiAgent Systems, pp. 435-436, 2000.

[34]. Michael Wooldridge and Nicholas R. Jennings, "Intelligent agents: Theory and practice", The Knowledge Engineering Review, vol. 10, no. 2, pp. 115-152, 1995.

[35]. http://www.e-mail-agent.com/

[36]. http://www.expita.com/howto1.html#www

[37]. http://www.expita.com/

[38]. http://www.mexeforum.org/

[39]. Roch H. Glitho, Edgar Olougouna, "Mobile Agents and Their Use for Information Retrieval: A Brief Overview and Elaborate Case Study", IEEE Networks, January/February 2002, pp. 35-41, 2002.

[40]. Herman Chung-Hwa Rao, Di-Fa Chang, Yi-Bing Lin, "iSMS: An Integration Platform for Short Message Service and IP Networks", IEEE Networks, March/April 2001, pp. 48-55, 2001.

[41]. Chong Xu, Dongbin Tao, "Building Distributed Application with Aglet", http://www.geocities.com/xuchong/aglet/.

[42]. Ravi Jain and Farooq Anjum, "Mobile Agents for Personalized Information Retrieval: When Are They a Good Idea?", 2000 IEEE Wireless Commun. and Net. Conf. (WCNC), vol. 1, Sept. 2000, pp. 242-245, 2000.

[43]. Yue-Shan Chang, Min-Huang Ho, Shyan-Ming Yuan, "Supporting unified interface to wrapper generator in Integrated Information Retrieval ", Computer Standards & Interfaces, Vol 24/4, Aug 2002, pp. 291-309, 2002.

[44]. Matti Sumujarvi, Sanna Mustonen, *GPRS Introduction*, http://www.ad.tut.fi/aci/courses/7601010/2002/ saatu/set_2/ GPRS_intro.pdf.

[45]. Yue-Shan Chang, Kai-Chih Liang, Ming-Chun Cheng, Shyan-Ming Yuan, "Prototyping an Integrated Information Gathering System on CORBA", Journal of Systems and Software, Vol. 72, No. 2, Jun 2004, pp. 281-294, 2004.

[46]. Paulo Jorge Marques, Luis Moura Silva, Joao Gabriel Silva, "A Flexible Mobile-Agent Framework for Accessing Information Systems in Disconnected Computing Environments", 2000 IEEE, pp. 242-245, 2000.

[47]. Hsin-Ta Chiao, Ming-Chun Cheng, Yue-Shan Chang, Shyan Ming Yuan, "A Component-based Reconfigurable Mobile Agent System for Context-Aware Computing", Lecture Notes in Computer Science (LNCS) of Springer-Verlag, Vol. 2457, pp. 273-282.

[48]. Michael Mahan, "Agents and Mobile Handsets", 2001 International Conference on Distributed Computing Systems, April 2001 pp. 448 -451, 2001.

[49]. Mitsuru Oshima, Guenter Karjoth, and Kouichi Ono, Aglets Specification 1.1 Draft, http://www.trl.ibm.co.jp/aglets/spec11.html, September 8th 1998.

[50]. Luz M. Quiroga, Javed Mostafa, "An Experiment in Building Profiles in Information Filtering: the Role of Context of User Relevance Feedback", Information Processing and Management, Vol. 38, No. 5, pp.

671-694, Sept. 2002.

[51]. Liang Zhang, Jiannong Cao, Sajal K. Das, “Mailbox-Based Scheme for Mobile Agent Communication”, IEEE Computer Society, pp. 54-60, Sept. 2002.

[52]. Stefano Porcarelli, Felicita Di Giandomenico, Andrea Bondavalli, Massimo Barbera, and Ivan Mura, “Service-Level Availability Estimation of GPRS”, IEEE Transactions on Mobile Computing, Vol. 2, No. 3, July-September 2003, pp. 233-247, 2003.

[53]. Soumen Chakrabarti, “Mining the web”, Morgan Kaufmann Publishers, San Francisco CA, 2003