

HANDHELD COMPUTING AND PALM OS FOR MOBILE HANDHELD DEVICES

Wen-Chen Hu
Department of Computer Science
University of North Dakota
Grand Forks, ND 58202-9015
wenchen@cs.und.edu

I-Lung Kao
IBM
Austin, TX 78758
ikao@us.ibm.com

Hung-Jen Yang
Department of Industrial Technology Education
National Kaohsiung Normal University
Kaohsiung, Taiwan
hjyang@nknucc.nknu.edu.tw

Jyh-haw Yeh
Department of Computer Science
Boise State University
Boise, Idaho 83725
jhych@cs.boisestate.edu

Abstract

Using Internet-enabled mobile handheld devices to access the World Wide Web is a promising addition to the Web and traditional e-commerce. Mobile handheld devices provide convenience and portable access to the huge information on the Internet for mobile users from anywhere and at anytime. However, most software engineers are not familiar with programming for handheld devices. Handheld computing is the programming for handheld devices such as smart cellular phones and PDAs (Personal Digital Assistants). It consists of two kinds of programming: client- and server- side programming. This paper introduces handheld computing and gives a detailed description of Palm OS programming, client-side handheld computing. Other related issues such as mobile commerce systems and handheld devices will be discussed also.

1. Introduction

It is widely acknowledged that mobile commerce is a field of enormous potential. However, it is also commonly admitted that the development in this field is constrained. There are still considerable barriers waiting to be overcome. One of the barriers is most software engineers are not familiar with handheld programming. Handheld computing is the programming for handheld devices such as smart cellular phones and PDAs (Personal Digital Assistants). This paper gives a study of handheld computing to help software engineers better understand this subject. It includes four major topics:

- *Mobile commerce systems*: The system structure includes six components: (i) mobile commerce applications, (ii) mobile handheld devices, (iii) mobile middleware, (iv) wireless networks, (v) wired networks, and (vi) host computers.
- *Mobile handheld devices*: A handheld device includes six major components: (i) a mobile operating system, (ii) a mobile central processor unit, (iii) a microbrowser, (iv) input/output devices, (v) a memory, and (vi) batteries
- *Handheld computing*: It consists of two kinds of programming: client- and server- side programming.
- *Palm OS programming*: The Palm OS Developer Suite is used to develop applications for Palm devices by handheld programmers.

This paper focuses on Palm OS programming by giving a step-by-step procedure of a Palm application development. Other client-side handheld computing is also discussed.

2. Background

This section discusses two themes of background information about this research, mobile commerce and Internet-enabled mobile handheld devices.

2.1 Mobile Commerce

With the introduction of the World Wide Web, electronic commerce has revolutionized traditional commerce and boosted sales and exchanges of merchandise and information. Recently, the emergence of wireless and mobile networks has made possible the extension of electronic commerce to a new application and research area: mobile commerce, which is defined as the exchange or buying and selling of commodities, services, or information on the Internet through the use of mobile handheld devices. In just a few years, mobile commerce has emerged from nowhere to become the hottest new trend in business transactions. To explain how the mobile commerce components work together, Figure 1 shows a flowchart of how a user request is processed by the components in a mobile commerce system, along with brief descriptions of how each component processes the request (Hu, et al, 2004).

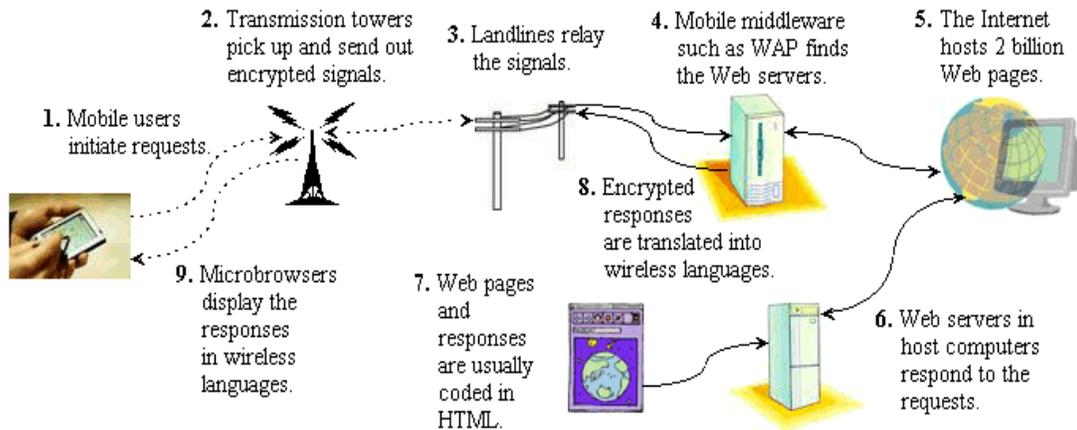


Figure 1: A flowchart of a user request processed in a mobile commerce system

1. *Mobile commerce applications:* Electronic commerce applications are numerous, including auctions, banking, marketplaces and exchanges, news, recruiting, and retailing, to name but a few. Mobile commerce applications not only cover the electronic commerce applications, but also include new applications, which can be performed at any time and from anywhere by using mobile computing technology, for example, mobile inventory tracking.
2. *Mobile handheld devices:* An Internet-enabled mobile handheld device is a small general-purpose, programmable, battery-powered computer that is capable of handling the front end of mobile commerce applications and can be operated comfortably while being held in one hand. It is the device with which mobile users interact directly with mobile commerce applications.
3. *Mobile middleware:* The term middleware refers to the software layer between the operating system and the distributed applications that interact via the networks. The primary mission of a middleware layer is to hide the underlying networked environment's complexity by insulating applications from explicit protocols that handle disjoint memories, data replication, network faults, and parallelism (Geihs, 2001). The major task of mobile middleware is to seamlessly and transparently map Internet contents to mobile stations that support a wide variety of operating systems, markup languages, microbrowsers, and protocols. WAP and i-mode are the two major kinds of mobile middleware. According to an article in Eurotechnology.com (Eurotechnology Japan K. K., n.d.), 60 percent of the world's wireless Internet users use i-mode (NTT DoCoMo, n.d.), 39 percent use WAP (Wireless Application Protocol Forum Ltd., n.d.), and 1 percent use Palm middleware. Table 1 compares i-mode and WAP.
4. *Wireless and wired networks:* Wireless communication capability supports mobility for end users in mobile commerce systems. Wireless LAN, MAN, and WAN are the major components used to provide radio communication channels so that mobile service is possible. In the WLAN category, the Wi-Fi standard with 11 Mbps throughput dominates the current market. However, it is expected that standards with

much higher transmission speeds, such as IEEE 802.11a and 802.11g, will replace Wi-Fi in the near future. Compared to WLANs, cellular systems can provide longer transmission distances and greater radio coverage, but suffer from the drawback of much lower bandwidth (less than 1 Mbps). In the latest trend for cellular systems, 3G standards supporting wireless multimedia and high-bandwidth services are beginning to be deployed.

5. *Host computers*: A user request such as database access or updating is actually processed at a host computer, which contains three major kinds of software: (i) Web servers, (ii) database servers, and (iii) application programs and support software.

2.2 Internet-enabled Mobile Handheld Devices

Mobile users interact with mobile commerce applications by using small wireless Internet-enabled devices, which come with several aliases such as handhelds, palms, PDAs, pocket PCs, and smart phones. To avoid any ambiguity, a general term, mobile handheld devices, is used in this article. Mobile handheld devices are small general-purpose, programmable, battery-powered computers, but they are different from desktop PCs or notebooks due to the following special features:

- limited network bandwidth,
- small screen/body size, and
- Mobility.

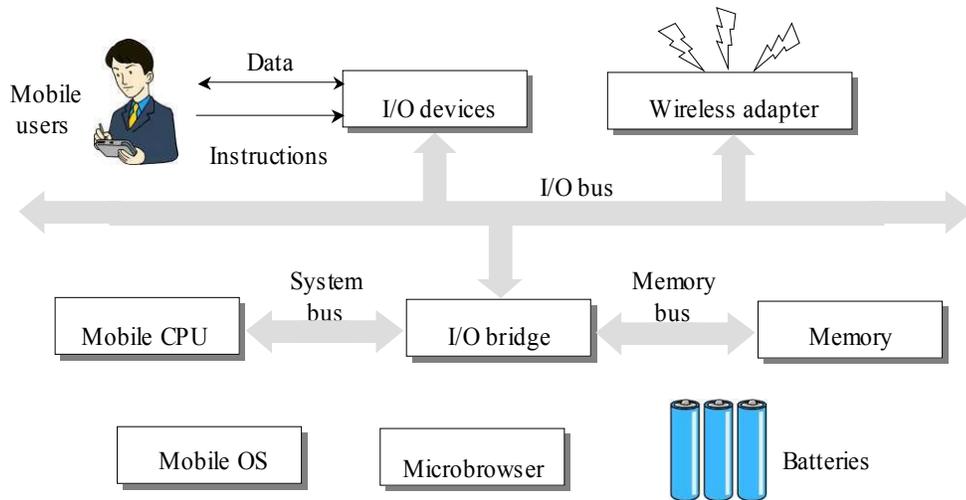


Figure 2: System structure of mobile handheld devices

Short battery life and limited memory, processing power, and functionality are additional features, but these problems are gradually being solved as the technologies improve and new methods are constantly being introduced. The limited network bandwidth prevents the display of most multimedia on a microbrowser. Though the Wi-Fi and 3G networks go some way toward addressing this problem, the wireless bandwidth is always far below

the bandwidth of wired networks. The small screen/body size restricts most handheld devices to using a stylus for input. Figure 2 shows a typical system structure for handheld devices, which includes the following six major components (Hu, et al, 2005):

1. *Mobile operating systems*: Simply adapting desktop operating systems for handheld devices has proved to be futile. A mobile operating system needs a completely new architecture and different features to provide adequate services for handheld devices. A generalized mobile operating system structure can be visualized as a six-layer stack: (i) applications, (ii) GUI, (iii) API framework, (iv) multimedia, communication infrastructure, and security, (v) computer kernel, power management, and real-time kernel, and (vi) hardware controller.
2. *Mobile central processing units*: Handheld devices are becoming more sophisticated and efficient every day and mobile users are demanding more functionality from their devices. To achieve this advanced functionality, in addition to the obvious feature, low cost, today's mobile processors must have the following features: (i) high performance, (ii) low power consumption, (iii) multimedia capability, and (iv) real-time capability. The cores and architectures designed by Cambridge-based ARM Holdings Ltd. have begun to dominate the mobile CPU market.
3. *Microbrowsers*: Microbrowsers are miniaturized versions of desktop browsers such as Netscape Navigator and Microsoft Internet Explorer. They provide graphical user interfaces that allow mobile users to interact with mobile commerce applications. Microbrowsers usually use one of the following four approaches to return results to the mobile user: (i) wireless language direct access, (ii) HTML direct access, (iii) HTML to wireless language conversion, and (iv) error.
4. *Input/output devices*: Various I/O devices have been adopted by mobile handheld devices. The only major output device is the screen, but there are several popular input devices, among them: (i) keyboards and (ii) touch screens/writing areas that need a stylus.
5. *Memory*: Three types of memory are usually employed by handheld devices: (i) RAM, (ii) ROM, and (iii) flash memory. Hard disks, which provide much more storage capacity, are likely to be adopted by handheld devices in the near future.
6. *Batteries*: At present, rechargeable Lithium Ion batteries are the most common batteries used by handheld devices. However, the life of this kind of battery is short and the technology will not significantly improve unless and until manufacturers begin to switch to fuel cells, which may not happen for at least several years.

Synchronization connects handheld devices to desktop computers, notebooks, or peripherals to transfer or synchronize data. Without needing serial cables, many handheld devices now use either an infrared (IR) port or Bluetooth technology to send information to other devices.

3. Handheld Computing

Handheld computing is a fairly new computing area and a formal definition of it is not found yet. Nevertheless, the authors define it as follows:

Handheld computing is the programming for handheld devices such as smart cellular phones and PDAs (Personal Digital Assistants). It consists of two kinds of programming: client- and server- side programming.

The definitions of client- and server- side programming are as follows:

- *Client-side handheld computing*: It is the programming for handheld devices and it does not need the supports from server-side programs. Some applications created by it are (a) address books, (b) video games, (c) note pads, and (d) to-do-list.
- *Server-side handheld computing*: It is the programming for wireless mobile handheld devices and it needs the supports from server-side programs. Some applications created by it are (a) instant messages, (b) mobile Web contents, (c) online video games, and (d) wireless telephony.

3.1 Definitions

Some popular mobile environments/languages for client-side handheld programming are listed below:

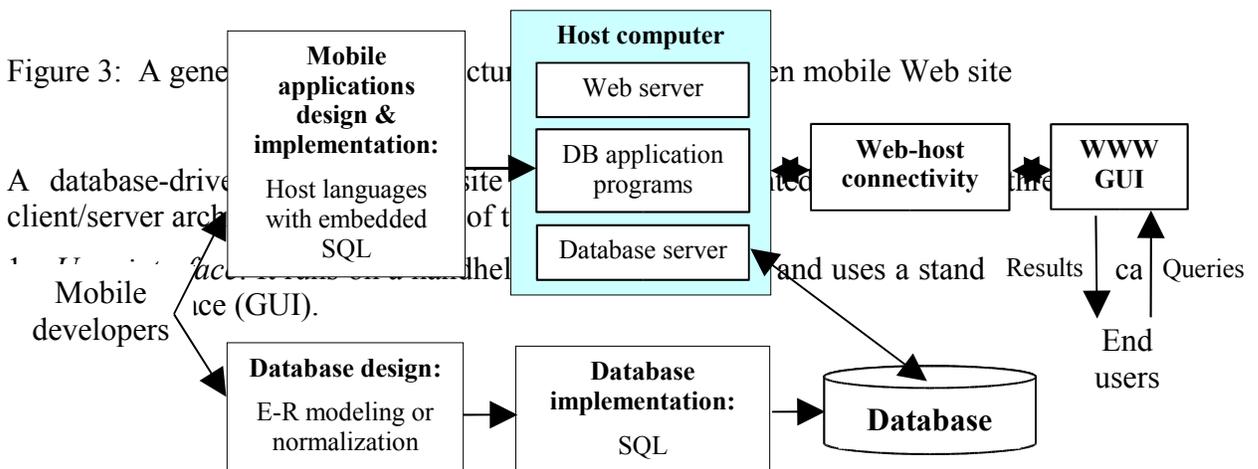
- *BREW (Binary Runtime Environment for Wireless)*: It is an application development platform created by Qualcomm Inc. for CDMA-based mobile phones (Qualcomm Inc., 2003).
- *J2ME (Java 2 Platform, Micro Edition)*: J2ME, developed by Sun Microsystems Inc., provides an environment for applications running on consumer devices, such as mobile phones, PDAs, and TV set-top boxes, as well as a broad range of embedded devices (Sun Microsystem Inc., 2002).
- *Palm OS*: Palm OS, developed by Palm Source Inc., is a fully ARM-native, 32-bit operating system running on handheld devices. Using Palm OS to build handheld applications will be introduced later.
- *Symbian*: Symbian Ltd. is a software licensing company that develops and supplies the advanced, open, standard operating system—Symbian OS—for data-enabled mobile phones (Symbian Ltd., 2005).
- *Windows Mobile*: Windows Mobile is a compact operating system for mobile devices based on the Microsoft Win32 API. It is designed to be similar to desktop versions of Windows (Microsoft Corp., 2005).

They apply different approaches to accomplishing handheld computing. Table 1 shows the comparison among these five handheld-computing languages/environments.

	BREW	J2ME	Palm OS	Symbian OS	Windows Mobile
Creator	Qualcomm Inc.	Sun Microsystems Inc.	PalmSource Inc.	Symbian Ltd.	Microsoft Corp.
Language/Environment	Environment	Language	Environment	Environment	Environment
Market Share (PDA) as of 2004	N/A	N/A	2 nd	N/A	1 st
Market Share (Smartphone) as of 2005	?	N/A	3 rd	1 st	2 nd
Primary Host Language	C/C++	Java	C/C++	C/C++	C/C++
Target Devices	Phones	PDA's & phones	PDA's	Phones	PDA's & phones

Table 1: A comparison among five handheld-computing languages/environments

Most applications created by this kind of programming such as instant messaging require network programming such as TCP/IP programming, which will not be covered in this paper. The most popular application of server-side handheld computing is database-driven mobile Web sites, whose structure is shown in Figure 3. A database-driven mobile Web site is often implemented by using a three-tiered client/server architecture consisting of three layers:



2. *Functional module*: This level actually processes data. It may consist of one or more separate modules running on a workstation or application server. This tier may be multi-tiered itself.
3. *Database management system (DBMS)*: A DBMS on a host computer stores the data required by the middle tier.

The three-tier design has many advantages over traditional two-tier or single-tier designs, the chief one being: The added modularity makes it easier to modify or replace one tier without affecting the other tiers.

3.2 Palm OS

Palm OS is a fully ARM-native, 32-bit operating system designed for use on Palm handhelds and other third-party devices. Palm OS runs on almost two out of every three PDAs. Its popularity can be attributed to its many advantages, such as its long battery life, support for a wide variety of wireless standards, and the abundant software available. The plain design of the Palm OS has resulted in a long battery life, approximately twice that of its rivals. It supports many important wireless standards, including Bluetooth and 802.11b local wireless and GSM, Mo-bitex, and CDMA wide-area wireless networks (PalmSource Inc., 2002). Two major versions of Palm OS are currently under development:

- *Palm OS Garnet*: It is an enhanced version of Palm OS 5 and provides features such as dynamic input area, improved network communication, and support for a broad range of screen resolutions including QVGA.
- *Palm OS Cobalt*: It is Palm OS 6, which focuses on enabling faster and more efficient development of smartphones and integrated wireless (WiFi/Bluetooth) handhelds.

3.2.1 Palm OS Programming

The Palm OS Developer Suite, which is the official development environment and tool chain from PalmSource, is intended for software developers at all levels. It is a complete IDE (Integrated Development Environment) for

- Protein applications (all ARM-native code) for Palm OS Cobalt and
- 68K applications for all shipping versions of the Palm OS.

The following steps show how to develop a Palm OS application, a simple “Hello, Mobile world!” program, under Microsoft Windows XP:

1. Download *Palm OS Developer Suite* at http://www.palmos.com/dev/tools/dev_suite.html.
2. Activate the Eclipse Workbench IDE as shown in Figure 4 under the Windows environment by selecting the following commands

Start ► All Programs ► PalmSource ► Palm OS Developer Suite

May select a default workspace at “C:\Program Files\PalmSource\Palm OS Developer Suite\workspace.”



Figure 4: A screenshot of the Palm OS Developer Suite

3. Create a new project by selecting a wizard: There are three Palm OS application types as shown in Figure 5:
 - *Palm OS 68K Application*,
 - *Palm OS 68K Application with PACE Native Objects*, and
 - *Palm OS Protein Application*.

There are also two kinds of *make files*:

- *Standard make*: It provides a generic set of makefiles that you can modify and tailor for your specific application build.
- *Managed make*: It dynamically generates your makefile based on the contents of your project folders.



Figure 5: A screenshot showing Palm OS application and *make* types

4. Create a Palm OS C/C++ program and put it in the directory “C:\Program Files\PalmSource\Palm OS Developer Suite\workspace\Hello\.” Figure 6 gives a Palm example, which displays the text “Hello, Mobile world!” an image, and a button “OK.”

```
C:\Program Files\PalmSource\Palm OS Developer Suite\workspace\Hello\HelloWorld.c
```

```

#include <PalmOS.h>

#define Form1 1000
#define OK 1003

UInt32 PilotMain( UInt16 cmd, void *cmdPBP, UInt16 launchFlags ) {
    short err;
    EventType e;
    FormType *pfrm;

    if ( cmd == sysAppLaunchCmdNormalLaunch ) {
        FrmGotoForm( Form1 );

        while( 1 ) {
            EvtGetEvent( &e, 100 );
            if ( SysHandleEvent( &e ) ) continue;
            if ( MenuHandleEvent( (void *) 0, &e, &err ) ) continue;

            switch ( e.eType ) {
                case ctlSelectEvent:
                    if ( e.data.ctlSelect.controlID == OK )
                        goto _quit;
                    break;

                case frmLoadEvent:
                    FrmSetActiveForm( FrmInitForm( e.data.frmLoad.formID ) );
                    break;

                case frmOpenEvent:
                    pfrm = FrmGetActiveForm( );
                    FrmDrawForm( pfrm );
                    break;

                case menuEvent:
                    break;

                case appStopEvent:
                    goto _quit;
                    break;

                default:
                    if ( FrmGetActiveForm( ) )
                        FrmHandleEvent( FrmGetActiveForm( ), &e );
                    break;
            }
        }
    }
    _quit:
        FrmCloseAllForms( );

    }
    return 0;
}

```

Figure 6: An example of a Palm OS program *HelloWorld.c*

For how to create Palm OS applications, check Palm OS Developer Documentation at <http://www.palmos.com/dev/support/docs/>. In order to display the current status on

the Eclipse, may need to constantly refresh the project *HelloWorld* by right clicking on the mouse on the project name as shown in Figure 7.

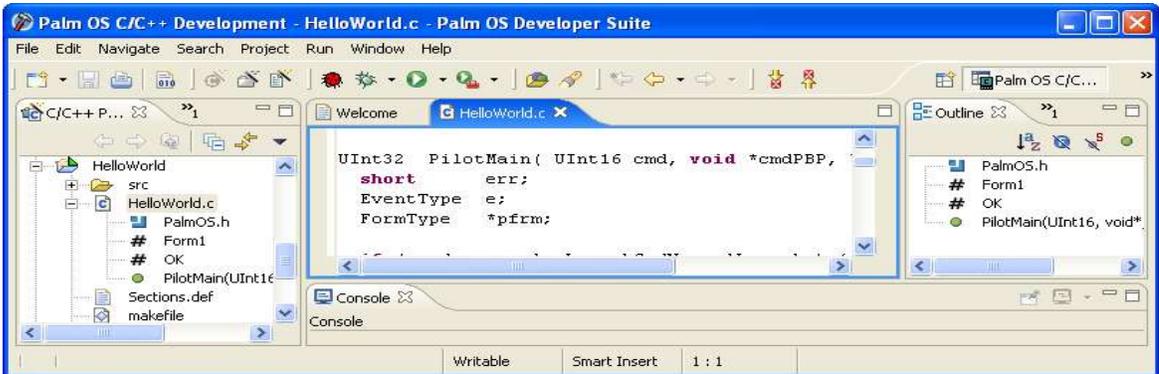


Figure 7: A screenshot of the Palm OS Developer Suite after the project *HelloWorld* is created

If the project includes resources (.xrd) such as buttons and images, the Palm OS Resource Editor at

Start ► All Programs ► PalmSource ► Tools ► Palm OS Resource Editor

could be used to create the resources as shown in Figure 8.



Figure 8: A screenshot of the Palm OS Resource Editor

5. Build the project *HelloWorld*.
6. Activate a Palm OS emulator by selecting

Start ► All Programs ► PalmSource ► Tools
 ► Palm OS Emulator

7. Drag the icon of *Hello.prc* (Palm Application file) at “C:\Program Files\PalmSource\Palm OS Developer Suite\workspace\Hello5\Debug\ Hello.prc” to the Palm OS emulator. Figure 9 shows the execution result of the project *HelloWorld*.
8. If the application is finalized, synchronize the application to handheld devices by selecting

Start ► All Programs ► Palm Desktop ►
 Palm Desktop

after downloading the Palm Desktop at
<http://www.palmos.com/dev/tools/desktop/>.



Figure 9: A screenshot of the execution results of the project *HelloWorld*

3.2.2 Palm OS Programmer’s API Reference

Table 2 gives the Palm OS programmer’s API reference of Palm OS 68K SDK (PalmSource Inc., n.d.). It includes four major sections (i) user interface, (ii) system management, (iii) communications, and (iv) libraries.

Function	Descriptions
User Interface	User interface APIs include events, notifications, attention, control, dialogs, forms, lists, menus, scroll bars, etc.
System Management	Provides largest number of functions such as alarm, debug, file streaming, graffiti, I/O, memory, pen, sound, time, windows, etc. for system management.
Communications	Provide various communication functions such as IR, modem, network, telephony, etc.
Libraries	Include miscellaneous libraries such as Internet, Bluetooth, cryptography, etc.

Table 2: Palm OS programmer’s API reference

4. Conclusion

Using Internet-enabled mobile handheld devices to access the World Wide Web is a promising addition to the Web and traditional e-commerce. Mobile handheld devices provide convenience and portable access to the huge information on the Internet for mobile users from anywhere and at anytime. However, most software engineers are not familiar with programming for handheld devices. Handheld computing is the programming for handheld devices such as smart cellular phones and PDAs (Personal Digital Assistants). This paper gives a study of handheld computing and includes four major topics:

- *Mobile commerce systems*: Mobile commerce is defined as the exchange or buying and selling of commodities, services, or information on the Internet through the use of mobile handheld devices. The system structure includes six components: (i) mobile commerce applications, (ii) mobile handheld devices, (iii) mobile middleware, (iv) wireless networks, (v) wired networks, and (vi) host computers.
- *Mobile handheld devices*: A mobile handheld device is a small general-purpose, programmable, battery-powered computer that is capable of handling the front end of mobile commerce applications and can be operated comfortably while being held in one hand. It includes six major components: (i) a mobile operating system, (ii) a mobile central processor unit, (iii) a microbrowser, (iv) input/output devices, (v) a memory, and (vi) batteries
- *Handheld computing*: It consists of two kinds of programming:
 - Client-side programming: It is the programming for handheld devices and it does not need the supports from server-side programs.
 - Server-side programming: It is the programming for wireless mobile handheld devices and it needs the supports from server-side programs.
- *Palm OS programming*: Two major versions of Palm OS are currently under development:
 - *Palm OS Garnet*: It is an enhanced version of Palm OS.
 - *Palm OS Cobalt*: It is the Palm OS 6.

This article focuses on Palm OS programming by giving a step-by-step procedure of a Palm application development. The Palm OS Developer Suite is used to develop applications for Palm devices by handheld programmers.

References

- Eurotechnology. (n.d.). Frequently Asked Questions about NTT-DoCoMo's i-mode. Retrieved October 9, 2005, from <http://www.eurotechnology.com /imode/faq.html>
- Geihs, K. (2001). Middleware challenges ahead. *IEEE Computer*, 34(6), 24-31.
- Hu, W.-C., Lee, C.-w., & Yeh, J.-h. (2004). Mobile commerce systems. Shi Nansi, editor, *Mobile Commerce Applications*, pages 1-23, Idea Group Publishing.

- Hu, W.-C., Yeh, J.-h., Chu, H.-J. Chu, and Lee, C.-w. Lee. (2005). Internet-enabled mobile handheld devices for mobile commerce. *Contemporary Management Research*, 1(1):13-34.
- Microsoft Corp. (2005). What's new for developers in Windows Mobile 5.0? Retrieved August 29, 2005, from http://msdn.microsoft.com/mobility/windowsmobile/howto/documentation/default.aspx?pull=/library/en-us/dnppcgen/html/whatsnew_wm5.asp
- NTT-DoCoMo. (n.d.). *i-mode*. Retrieved October 2, 2005, from <http://www.nttdocomo.com/>
- PalmSource Inc. (2002). Why PalmOS? Retrieved June 23, 2005, from http://www.palmsource.com/palmos/Advantage/index_files/v3_document.htm
- PalmSource Inc. (n.d.). Palm OS programmer's API reference. Retrieved August 15, 2005 from <http://www.palmos.com/dev/support/docs/palmos/PalmOSReference/ReferenceTOC.html>
- Qualcomm Inc. (2003). BREW and J2ME—A complete wireless solution for operators committed to Java. Retrieved February 12, 2005, from http://brew.qualcomm.com/brew/en/img/about/pdf/brew_j2me.pdf
- Sun Microsystems Inc. (2002). Java 2 Platform, Micro Edition. Retrieved January 12, 2006, from <http://java.sun.com/j2me/docs/j2me-ds.pdf>
- Symbian Ltd. (2005). Symbian OS Version 9.2. Retrieved December 20, 2005, from http://www.symbian.com/technology/symbianOSv9.2_ds_0905.pdf
- Wireless Application Protocol Forum Ltd. (n.d.). *WAP* (Wireless Application Protocol). Retrieved July 21, 2005 from <http://www.wapforum.org/>