Handheld Computing and Programming for Mobile Commerce

By: WEN-CHEN HU, JYH-HAW YEH, LIXIN FU, and HUNG-JEN YANG

Wen-Chen Hu, Jyh-Haw Yeh, Lixin Fu, Hung-Jen Yang, Handheld Computing and Programming for Mobile Commerce, International Journal of Web Information Systems, Volume 2, Issue3-4, 2006, p164-175.

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, mobile commerce has not enjoyed the same level of success as the e-commerce has so far because mobile Web contents are scarce and mostly awkward for browsing. The major reason of the problems is most software engineers are not familiar with handheld devices, let alone programming for them. To help software engineers better understand this subject, this article gives a comprehensive study of handheld computing and programming for mobile commerce. It includes live major topics: (i) mobile commerce systems, (ii) mobile handheld devices, (iii) handheld computing, (iv) server-side handheld computing and programming, and (v) client-side handheld computing and programming. The most popular server-side handheld applications are mostly functioning through mobile Web contents, which are constructed by using only few technologies and languages. On the other hand, various environments/languages are available for client-side handheld computing and programming. Five of the most popular are (i) BREW, (ii) J2ME, (iii) Palm OS, (iv) Symbian OS, and (v) Windows Mobile. They are using either C/C++ or Java programming languages. This article will explain J2ME, a micro version of Java, and Palm OS programming, using C, by giving step-by-step procedures of J2ME and Palm application development.

Index Terms—Handheld computing and programming, .J2ME, mobile commerce, mobile handheld devices, Palm OS

Article:

1. INTRODUCTION

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 such as smart cellular phones and PDAs (Personal Digital Assistants). 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 considerable harriers waiting to be overcome. One of the barriers is most software engineers are not familiar with the design and development of mobile applications I l I. This article gives a comprehensive study of handheld computing and programming for mobile commerce to help software engineers better understand this subject. It includes five major topics:

- *Mobile commerce systems:* The system for implementing mobile commerce is fairly complicated and it involves several different disciplines such as business, telecommunications, and computer programming. The mobile- commerce 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:* Mobile users use handheld devices to perform the mobile commerce transactions. A handheld device also 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:* Handheld computing is to use handheld devices to perform wireless, mobile, handheld operations such as personal data management and making phone calls. They can be achieved by using server- or client- side handheld computing and programming.
- *Server-side handheld computing and programming:* Server-side handheld computing is to use handheld devices to perform wireless, mobile, handheld operations, which require the supports of server-side computing. The most common applications of server-side handheld programming are the mobile Web contents.
- *Client-side handheld computing and programming:* Client-side handheld computing is to use handheld devices to perform handheld operations, which do not need the supports of server-side computing. Most client- side handheld programming languages are a version of either C/C++ or Java. The application development of J2ME, a version of Java, and Palm OS, using a version of C, will be given step-by-step.

II. MOBILE COMMERCE SYSTEMS

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.

A. A System Structure

A mobile commerce system is inherently interdisciplinary and could be implemented in various ways. Fig. 1 shows the structure of a mobile commerce system and a typical example of such a system [2]. 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.

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

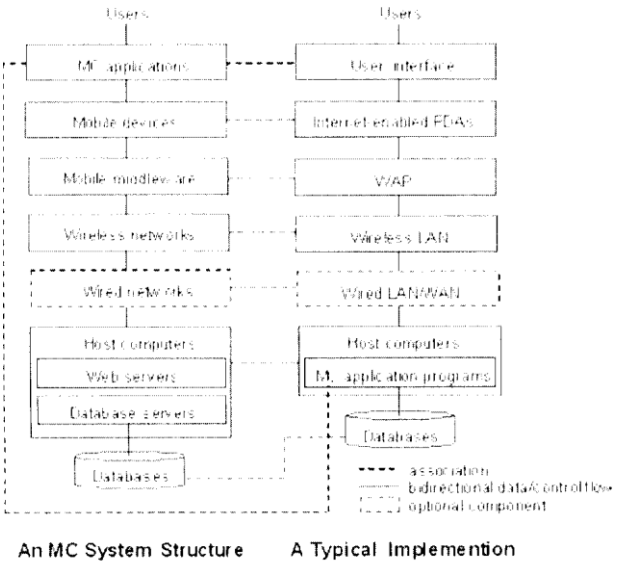

Fig. 1. A mobile commerce system structure.

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 he 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 13]. 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 [4] and i-mode [51 are the two major kinds of mobile middleware. Table I 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 802.1 1 a 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 I 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.

TABLE I
A COMPARISON BETWEEN TWO MAJOR TYPES OF MOBILE MIDDLEWARE

	WAP	i-mode
Developer	Open Mobile Alliance	NTT DoCoMo
Implementation	A protocol	A complete mobile Internet service
Web Language	WML	CHTML
Major Technology	WAP Gateway	TCP/IP development
Key Features	Widely adopted and flexible	Highest number of users and easy to use

B. Mobile Commerce Transactions

To explain how the mobile commerce components work together, Fig. 2 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 121.

I) *Mobile commerce applications:* A content provider implements an application by providing two sets of programs: client-side programs, such as user interfaces on microbrowsers, and server-side programs, such as database access and updating.

2) *Mobile handheld devices:* Handheld devices present user interfaces to the mobile end users, who specify their requests on the interfaces. The devices then relay the user requests to the other components and later display the processing results using the interfaces.

3) *Mobile middleware:* The major purpose 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. Most mobile middleware also encrypts the communication in order to provide some level of security for transactions.

4) *Wireless and mobile networks:* Mobile commerce is possible mainly because of the availability of wireless networks. User requests are delivered to either the closest wireless access point (in a wireless local area network environment) or a base station (in a cellular network environment).

5) *Wired networks:* This component is optional for a mobile commerce system. However, most computers (servers) usually reside on wired networks such as the Internet, so user requests are routed to these servers using transport and/or security mechanisms provided by wired networks.

6) *Host computers:* Host computers process and store all the information needed for mobile commerce applications, and most application programs can be found here. They include three major components: Web servers, database servers, and application programs and support software.

III. 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
- high mobility.

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.

Fig. 3 shows a typical system structure for handheld devices, which includes the following 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 161:

I) *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 as shown in Fig. 4.

2) Mobile central processing units: Handheld devices are becoming more sophisticated and efficient every day and mobile users are demanding more functionality from

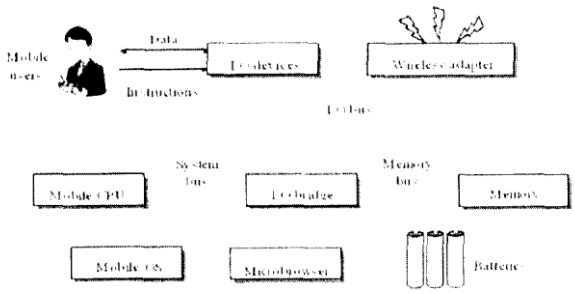

Fig. 3. A system structure of mobile handheld devices.

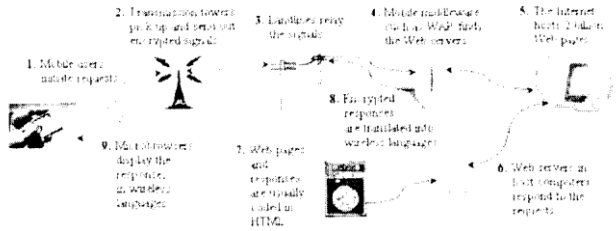

Fig. 2. A flowchart of a user request processed in a mobile commerce system.

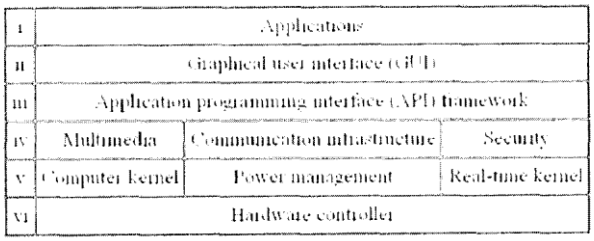

Fig. 4. A generalized mobile operating system structure.

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 four approaches to return results to the mobile users: (i) wireless language direct access, (ii) HTML direct access, (iii) HTML to wireless language conversion, and (iv) error as shown in Fig. 5.
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.

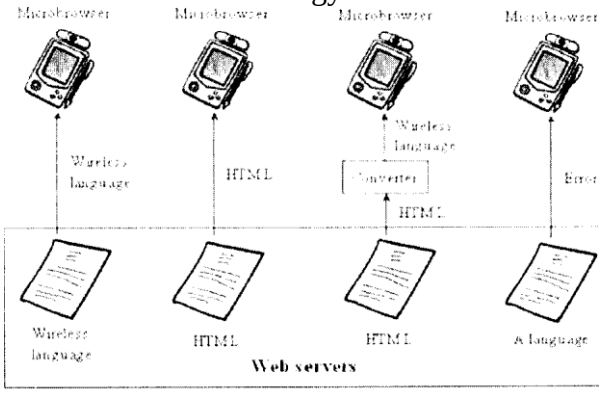

Fig. 5. Four approaches for microbrowsers to display mobile contents.

IV. HANDHELD COMPUTING AND PROGRAMMING

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 to use handheld devices such as smart cellular phones and PDAs (Personal Digital Assistants) to perform wireless, mobile, handheld operations such as personal data management and making phone calls.

Again, handheld computing includes two kinds of computing: server- and client- side handheld computing, which are defined as follows:

- *Server-side handheld computing:* It is to use handheld devices to perform wireless, mobile, handheld operations, which require the supports of server-side computing. Examples of its applications include (a) instant messages, (b) mobile Web contents, (c) online video games, and (d) wireless telephony.
- *Client-side handheld computing:* It is to use handheld devices to perform handheld operations, which do not need the supports of server-side computing. Examples of its applications include (a) address books, (b) standalone video games, (c) note pads, and (d) to-do-list.

The terms of computing and programming are sometimes confusing and misused. The handheld programming, defined as programming for handheld devices, is different from handheld computing and includes two kinds of programming too:

- *Server-side handheld programming:* It is design and development of handheld software such as CGI programs that reside on the servers.
- *Client-side handheld programming:* It is design and development of handheld software such as J2ME programs that reside on the handheld devices.

A. Server-Side Handheld Computing and Programming

Server-side handheld computing and programming usually involve complicated procedures and advanced programming such as TCP/IP network programming. This article will focus on the most popular server-side handheld computing and programming, mobile Web contents design and development. For other kinds of server-side handheld computing and programming such as instant messaging and telephony, readers may refer to other technical reports or articles. A database- driven mobile Web site is often implemented by using a three- tiered client/server architecture consisting of three layers as shown in Fig. 6: (i) user interface, (ii) functional modules, and (iii) database management systems, which 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.

User interface. It runs on a handheld device (the client) and uses a standard graphical user interface (GUI). Many mobile Web contents are using WML (Wireless Markup Language) 171, which is a mobile version of HTML (HyperText

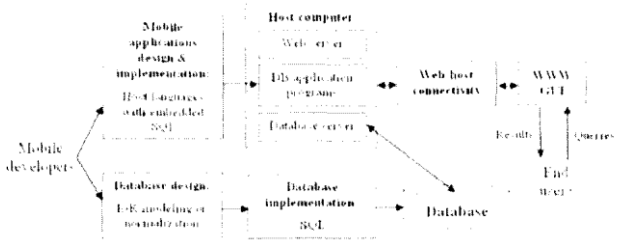

Fig. 6. A generalized system structure of a database-driven mobile web site.

TABLE II

A COMPARISON AMONG FIVE HANDHELD-COMPUTING LANGUAGES/ENVIRONMENTS

	BREW	J2ME	Palm OS	Symbian OS	Windows Mobile
Creator	Qualcomm	Sun	PalmSource	Symbian	Microsoft
(L)anguage/ (E)nvironment	E	L	E	E	E
PDA Market Share as of 2005	N/A	N/A	3^{rd}	4^{th}	1^{st}
Smartphone Market Share as of 2006	?	N/A	4^{th}	1^{st}	5^{th}
Primary Host Language	C/C++	Java	C/C++	C++	C/C++
Target Devices	Phones	PDAs/ phones	PDAs	Phones	PDAs/ phones

Markup Language), in short. WML is not well received because most site owners are reluctant to create a mobile version of their Web sites and some microbrowsers are able to view and adapt to HTML pages. Producing contents that will run on the many mobile devices available to users is not an easy task. To help solve this problem, the World Wide Web Consortium is working on a new technology. The W3C's Device Independent Authoring Language [81 is a markup language used to make single versions of Web sites and applications available to a wide range of cellular phones, PDAs, and other mobile machines. Systems that implement DIAL recognize devices and adapt the content appropriately to take into account issues such as display size and resolution 191.

Functional modules. 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. One common module is using JDBC (Java DataBase Connectivity) to access and manage databases.

B. Client-Side Handheld Computing and Programming

Various environments/languages are available for client- side handheld computing and programming. Five of the most popular are (i) BREW, (ii) J2ME, (iii) Palm OS, (iv) Symbian OS, and (v) Windows Mobile. They apply different approaches to accomplishing the development of mobile applications. Fig. 7 shows a generalized development cycle applied by them and Table II gives a comparison among the five languages/environments. The second half of this article is devoted to the details of J2ME and Palm OS and brief introductions of thc other three are given in this section.

I) BREW (Binary Runtime Environment for Wireless): BREW is an application development platform created by Qualcomm for CDMA (Code Division Multiple Access) - based mobile phones 1 10]. The CDMA is a digital wire less telephony transmission technique and it has two major features:

- The CDMA allows multiple frequencies to be used simultaneously (Spread Spectrum).
- The CDMA standards used for second-generation mobile telephony are the IS-95 standards.

BREW is a complete, end-to-end solution for wireless applications development, device configuration, application distribution, and billing and payment. It includes three major components:

- BREW SDK (software development kit) for application developers,
- BREW client software and porting tools for device manufacturers, and
- BREW Distribution System (BDS) that is controlled and managed by operators—enabling them to easily get applications from developers to market and coordinate the billing and payment process.

It also includes the following special features:

- Qualcomm designed the BREW platform from the chip out instead of scaling down a product developed for PCs or PDAs.
- It supports other languages beyond native C/C++, including alternative execution environments such as Java and XML.

The BREW client software serves as a common denominator across all types and tiers of devices. It exposes a common set of application programming interfaces (APIs) for standardized development of wireless applications. The BREW client can act as an extended platform for other environments (such as VMs) and allow any type of browser (HTML, WAP, cHTML, etc.) to run on BREW as an application.

2) Symbian OS: Symbian is a software licensing company that develops and supplies the open operating system-Symbian OS—for data-enabled mobile phones [1 I 1. Symbian was established as a private independent company in June 1998. It is an independent, for-profit company whose mission is to establish Symbian OS, whose architecture is given in

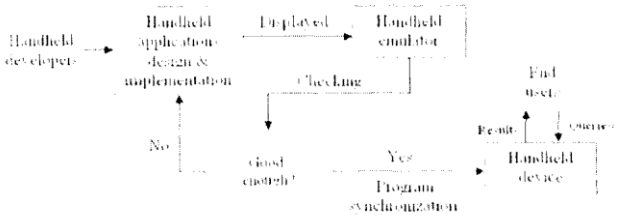

Fig. 7. A generalized client-side handheld application development cycle.

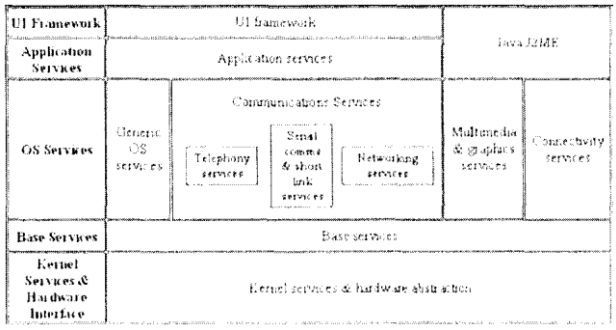

Fig. 8. Symbian OS architecture.

Fig. 8, as the world standard tor mobile digital data systems, primarily for use in cellular telecoms. It is owned by Ericsson (15.6%), Nokia (47.9%), Panasonic (10.5%), Samsung (4.5%), Siemens (8.4%) and Sony Ericsson (13.1%). Headquartered in the UK, it has more than 1,300 staff with offices in Japan, Sweden, UK and the USA and a development centre in Bangalore in 2006. It is the most popular mobile operating system for smartphones. Cumulative shipments of Symbian OS phones since Symbian's formation reached 70.5 million phones in 2006. Key features of Symbian OS include:

- *Suite of application engines.* Contacts, schedule, messaging, browsing, utility and system control; appointments and business cards; integrated APIs for data management, text, clipboard and graphics.
- *Messaging.* Multimedia messaging, enhanced messaging and SMS; Internet mail; attachments; fax.
- *Multimedia.* Audio and video support for recording, playback and streaming; image conversion; direct access to screen and keyboard for high performance; graphics accelerator API.
- *Communications protocols.* Wide-area networking stacks including TCP/IP and WAP, personal area networking support including infrared, Bluetooth wireless technology and USB.
- *Mobile telephony.* GSM circuit switched voice and data and packet-based data; CDMA circuit switched voice, data and packet-based data; SIM, RUIM and UICC Toolkit.
- *Security.* Full encryption and certificate management, secure protocols, WIM framework and certificate-based application installation.
- *Developing for Symbian OS.* Content development options include: C++, Java MIDP 2.0 and PersonalJava 1.1.1a, and WAP.
- *User inputs.* Full keyboard, numeric mobile phone keypad, voice, handwriting recognition and predictive text input.

3) Windows Mobile: Windows Mobile is a compact operating system for mobile devices based on the Microsoft Win32 API 1121. It is designed to be similar to desktop versions of Windows. In 1996, Microsoft launched Windows CE, a version of the Microsoft Windows operating system designed specially for a variety of embedded products, in- eluding handheld devices. However, it was not well received primarily because of battery-hungry hardware and limited functionality, possibly due to the way that Windows CE was adapted for handheld devices from other Microsoft 32-bit desktop operating systems. Windows Mobile includes three major kinds of software: (i) Pocket PC, (ii) Smartphones, and (iii) Portable Media Centers, which let users

take recorded TV programs, movies, home videos, music, and photos transferred from Microsoft Windows XP-based PC anywhere.

Pocket PCs. Pocket PC enables you to store and retrieve e-mail, contacts, appointments, games, exchange text messages with MSN Messenger, browse the Web, and so on. Pocket PCs were designed with better service for mobile users in mind and offers far more computing power than Windows CE. It provides scaled-down versions of many popular desktop applications, including Microsoft Outlook, Internet Explorer, Word, Excel, Windows Media Player, and others. It also includes threc major kinds of software:

- *Pocket PC:* It puts the power of Windows software into a Pocket PC, giving you time to do more with the people and things that matter.
- *Pocket PC Phone Edition:* It combines all the standard functionality of a Windows Mobile-based Pocket PC with that of a feature-rich mobile phone.
- *Ruggedized Pocket PC:* It lets you do more of what

matters to you even in the toughest user environments.

Smartphones. Smartphone supplies functions of a mobile phone, but also integrates PDA-type functionality, such as emails, instant messages, music, and Web surfing, into a voice- centric handset. Windows Mobile-based Smartphone integrates PDA-type functionality into a voice-centric handset comparable in size to today's mobile phones. It is designed for one- handed operation with keypad access to both voice and data features. The Smartphone is a Windows CE-based cellular phone. Like the Pocket PC, all Smartphones regardless of manufacturer share the same configuration of Windows CE. Also, Smartphones come bundled with a set of applications such as an address book, calendar, and e-mail program. Fig. 9 shows the Smartphone architecture, which provides a core set of services that will abstract a variety of underlying links for both voice and data services I131. The primary Smartphone architecture consists of four layers:

1) *Applications/UI:* The top level refers to the Smartphone shell and customer-level applications such as Pocket Internet Explorer, the Inbox, the control panel, and the phone dialer.
2) *Logic:* This level contains system application logic that can be used by the application layer. Examples of

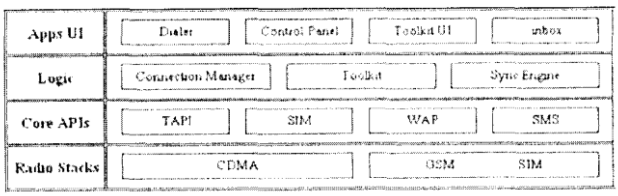

Fig. 9. Windows Mobile-based Smartphone architecture.

this include the control of network connections and synchronization capabilities.

3) *Core APIs:* This level provides the interfaces between the low-level architecture components and the application/logic layers. By developing applications targeted at this layer and the one above, developers do not need to know the underlying low level details in order to take full advantage of their capabilities.

4) *Radio Stack:* The bottom level refers, in general, to the architectural components responsible for voice and data control and data transmission.

V. J2ME (JAVA 2 PLATFORM, MICRO EDITION)

J2ME 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 114]. Like its counterparts for the enterprise (J2EE), desktop (J2SE) and smart card (Java Card) environments, J2ME includes Java virtual machines and a set of standard Java APIs defined through the Java Community Process, by expert groups whose members include device manufacturers, software vendors, and service providers.

A. ,J2ME Architecture

The J2ME architecture, as shown in Fig. 10, comprises a variety of configurations, profiles, and optional packages that implementers and developers can choose from, and combine to construct a complete Java runtime environment that closely fits the requirements of a particular range of devices and a target market. There are two sets of J2ME packages, which target different devices:

- *High-end devices:* They include Connected Device Configuration (CDC), Foundation and Personal Profile.
- *Entry-level devices and smart phones:* They include Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP).

Configurations comprise a virtual machine and a minimal set of class libraries and they provide the base functionality for a particular range of devices that share similar characteristics, such as network connectivity and memory footprint. Profiles provide a complete runtime environment for a specific device category.

B. J2ME Programming

This sub-section gives an example of J2ME programming [15]. Other client-side handheld programming is similar to this. Fig. 11 shows the Sun Java Wireless Toolkit, which is a toolbox for developing wireless applications that are based on J2ME's CLDC and MIDP. The toolkit includes the emulation environments, performance optimization and tuning features, documentation, and examples that developers need to bring efficient and successful wireless applications to market quickly. The following steps showing how to develop an MIDP application, a simple "Hello, World!" program, under Microsoft Windows XP:

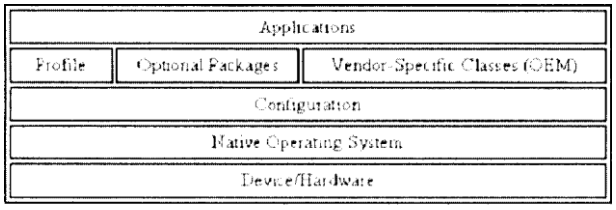

Fig. 10. J2ME architecture.

1) Download *Sun Java Wireless Toolkit 2.3 Beta,* which includes a set of tools and utilities and an emulator for creating Java applications that run on handheld devices, at http : / / java . sun . com/ products / sjwtoolkit / down-load-2_3 . html.

2) Run MIDIet, an MIDP application, development environment *KToolbar* as shown in Fig. 11 by selecting the following Windows options:

Start ⇒ *All Programs* ⇒ *SunJava Wireless Toolkit 2.3 Beta* ⇒ *KToolbar*

3) Create a new project by giving a project name such as HelloSuite and a class name such as HelloMIDlet as shown in Fig. 12. After the project HelloSuite is created, the KToolbar will display the message shown in Fig. 13, which tells where to put the Java source files, application resource files, and application library files.

4) Create a J2ME source program and put it in the directory C:\WTK23\apps\HelloSui-te\src\ Fig. 14 gives a J2ME example, which displays the text "Hello, World!" and a ticker with a message "Greeting, world."

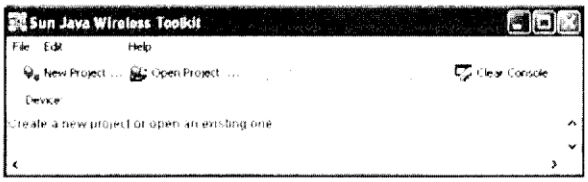

Fig. 11. A screenshot of KToolbar after launching.

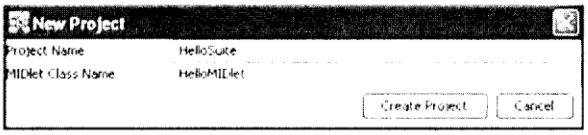

Fig. 12. A screenshot of a pop-up window after clicking on the button `New Project` of KToolbar.

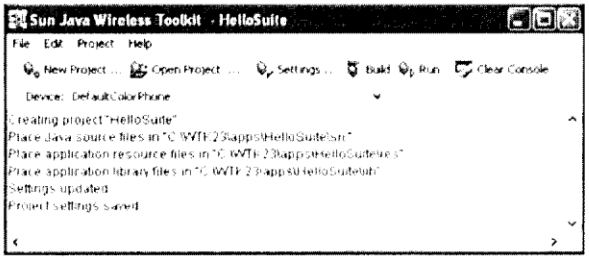

Fig. 13. A screenshot of KToolbar after a project `HelloSuite` created.

C:\WTK23\apps\HelloSuite\src\HelloMIDlet.java

```
// This package defines MIDP applications and the interactions between
// the application and the environment in which the application runs.
import  javax.microedition.midlet.*;

// This package provides a set of features for user interfaces.
import  javax.microedition.lcdui.*;

public class HelloMIDlet  extends MIDlet  implements CommandListener {

  public void  startApp( ) {
    Display  display     = Display.getDisplay( this );
    Form     mainForm    = new  Form   ( "HelloMIDlet" );
    Ticker   ticker      = new  Ticker ( "Greeting, World" );
    Command  exitCommand = new  Command( "Exit", Command.EXIT, 0 );

    mainForm.append           ( "\n\n          Hello, World!" );
    mainForm.setTicker        ( ticker );
    mainForm.addCommand       ( exitCommand );
    mainForm.setCommandListener( this );
    display.setCurrent        ( mainForm );
  }

  public void  pauseApp ( ) { }

  public void  destroyApp( boolean unconditional ) {
    notifyDestroyed( );
  }

  public void  commandAction( Command c, Displayable s ) {
    if ( c.getCommandType( ) == Command.EXIT )
      notifyDestroyed( );
  }
}
```

Fig. 14. An example of the MIDlet program `HelloMIDlet.java`.

Package	Classes and Description
User Interface	`javax.microedition.lcdui` The UI API provides a set of features for implementation of user interfaces for MIDP applications
	`javax.microedition.lcdui.game` The Game API package provides a series of classes that enable the development of rich gaming content for wireless devices
Persistence	`javax.microedition.rms` It provides a mechanism for MIDlets to persistently store data and later retrieve it
Application Lifecycle	`javax.microedition.midlet` The MIDlet package defines MIDP applications and the interactions between the application and the environment in which the application runs
Networking	`javax.microedition.io` The MID Profile includes networking support based on the Generic Connection framework from the *Connected Limited Device Configuration*
Audio	`javax.microedition.media` The MIDP 2.0 Media API is a directly compatible building block of the Mobile Media API (JSR 135) specification
	`javax.microedition.media.control` This package defines the specific Control types that can be used with a Player
Public Key	`javax.microedition.pki` Certificates are used to authenticate information for secure Connections
Core	`java.io` Provides classes for input and output through data streams
	`java.lang` MID Profile Language Classes included from Java 2 Standard Edition
	`java.util` MID Profile Utility Classes included from Java 2 Standard Edition

Fig. 16. Mobile Information Device Profile (MIDP) package list.

5) Build the project by clicking on the Build button. The Build includes compilation and pre-verifying.
6) Run the project by clicking on the Run button. An emulator will be popped up and displays the execution results of the built project. For example, Fig. 15 shows an emulator displays the execution results of HelloSuite.
7) Upload the application to handheld devices by using USB cables, infrared ports, or Bluetooth wireless technology.

C. .J2ME References

Fig. 16 shows the packages provided by the MIDP 1161. The packages j avax . * are the extensions to standard Java packages. They are not included in the JDK or JRE. They must be downloaded separately.

VI. PALM OS

Palm OS is a fully ARM-native, 32-bit operating system designed for used on Palm handhelds and other third-party devices. 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 [171.

A. Palm OS Architecture

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 smart- phones and integrated wireless (WiFi/Bluetooth) handhelds.

Fig. 17 shows the structure of Palm OS 5, which consists of live layers:

1) *Applications:* They include all of the built-in Palm OS applications, such as address book, date book, and memo pad.
2) *PACE (Palm Application Comparibility Environment):* PACE provides a 68K application environment that is equivalent to Palm OS 4.1. PACE handles the data translation required for a 68K application to run on Palm OS 5. For example, 68K applications read and write data in big-endian mode, hut Palm OS 5 views data in little-endian mode. When a 68K application calls a Palm OS function, PACE handles the translation of

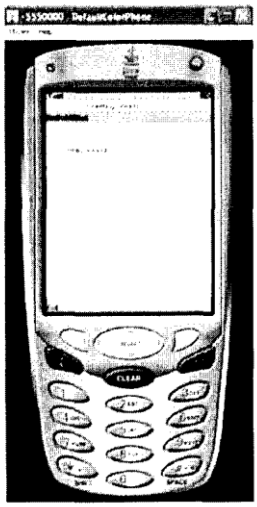

Fig. 15. A screenshot of an emulator displaying the execution results of `HelloSuite`.

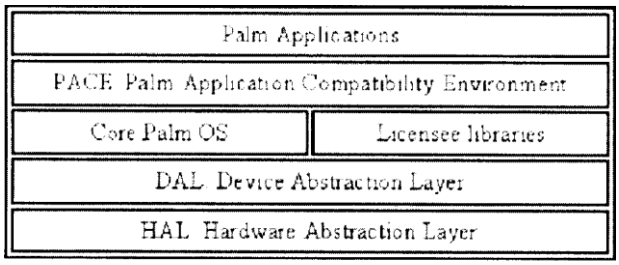

Fig. 17. Palm OS 5 block diagram.

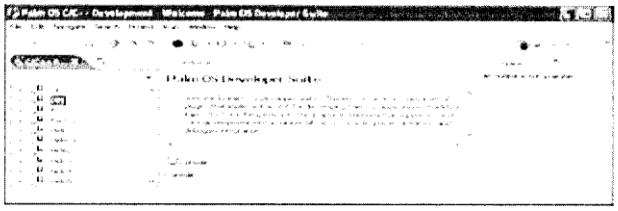

Fig. 18. A screenshot of the Palm OS Developer Suite.

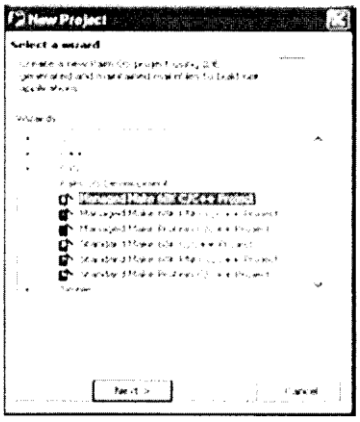

Fig. 19. A screenshot showing Palm OS application and make types.

the parameters, objects, and structure layouts so that existing applications do not have to he updated to handle the change of endianness. PACE creates "shadow structures" for the 68K application's data that allow the 68K application to run under Palm OS 5.

3) *Core Palm OS and Licensee Libraries:* This layer includes two functions:
   - *Palm OS runtime services:* This function requires the desktop system library files (DLLs). These DLLs are located in the Palm OS directory, and are loaded when the code that they contain needs to be executed.
   - *Communication stacks:* Palm OS uses the communication stacks for NetLib and Telephony components. Palm OS can optionally redirect NetLib calls to the host machine TCP/IP stack.
4) *DAL (Device Abstraction Layer): It* is responsible for insulating Palm OS from the underlying system and hardware.
5) *HAL (Hardware Abstraction Layer):* It allows a computer operating system to interact with a hardware device at a general or abstract level rather than at a detailed hardware level.

B. 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 and install the Palm OS Developer Suite at http://www.palmos.com/dev/tools/dev_suite. html.
2) Activate the Eclipse Workbench IDE as shown in Fig. 18 under the Windows environment by selecting the following options:

3) Create a new project by selecting a wizard. There are three Palm OS application types as shown in Fig. 19:
   - *Paim 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 make- tiles 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.
4) Create a Palm OS C/C++ program and put it in the directory C:\Program Files\PalmSource\Palm OS Dev eloper Suite \workspace \HelloWorld\. Fig. 20 gives a Palm example, which displays the text "Hello, Mobile world!," an image, and a button "OK" on a Palm device. 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, users may need to constantly refresh the project HelloWorld by right clicking on the mouse on the project name as shown in Fig. 21. If the project includes resources (with an .xrd filename extension) such as buttons and images, the Palm OS Resource Editor at the following path:

   $Start \Rightarrow All\ Programs \Rightarrow PalmSource \Rightarrow Tools \Rightarrow Palm\ OS\ Resource\ Editor$

   $Start \Rightarrow AllPrograms \Rightarrow PalmSource \Rightarrow Palm\ OS\ Developer\ Suite$

   could be used to create the resources as shown in Fig. 22.

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

// This header is from the Palm SDK and contains the needed reference
// materials for the use of Palm API and its defined constants.
#include <PalmOS.h>

// The following IDs are from using Palm Resource Editor
#define   Form1  1000
#define   OK     1003

// ---------------------------------------------------------------------
// PilotMain is called by the startup code and implements a simple
// event handling loop
// ---------------------------------------------------------------------
UInt32 PilotMain( UInt16 cmd, void *cmdPBP, UInt16 launchFlags ) {
  Short       err;
  EventType   e;
  FormType    *pfrm;

  if ( cmd == sysAppLaunchCmdNormalLaunch ) {
    // Displays the Form with an ID 1000.
    FrmGotoForm( Form1 );

    // Main event loop
    while( 1 ) {
      // Doze until an event arrives or 100 ticks are reached
      EvtGetEvent( &e, 100 );
      // System gets first chance to handle the event
      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;
}
```

Fig. 20. An example of the Palm OS program `HelloWorld.C`.

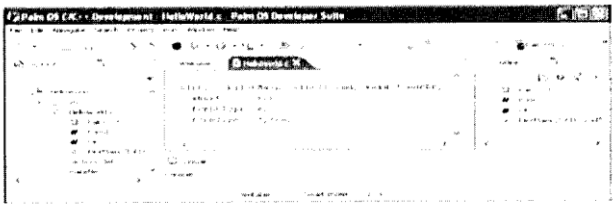

Fig. 21. A screenshot of the Palm OS Developer Suite after the project `HellowWorld` is created.

Fig. 22. A screenshot of the Palm OS Resouree Editor.

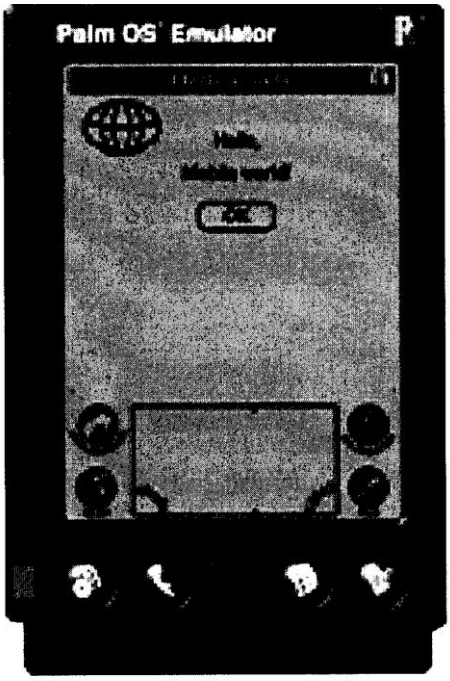

Fig. 23. A screenshot of the execution results of the project HelloWorld.

5) Build the project HelloWorld.
6) Activate a Palm OS emulator by selecting the following options:

Start ⇒ *All Programs* ⇒ *PalmSource* ⇒ *Tools* ⇒ *Palm OS Emulator*

7) Drag the icon of Hello .prc (Palm Application File) at C:\ProgramPiles\PalmSource\Palm OS Develo per Suite\workspace\Hello5\Debug\Hello.prc to the Palm OS emulator. Fig. 23 shows the execution results of the project HelloWorld.
8) If the application is finalized, synchronize the application to handheld devices by selecting the following options:

C. Palm References

Since this article is never intended to be a comprehensive Palm programming guide, this part provides the interested readers more Palm information for further references. Fig. 24 shows four documents for Palm OS SDK (Software Development Kit). Further information of the two documents *Palm OS Programmer,s Companion* and *Palm OS Programmer's APJ Reference* arc given next.

The Palm OS Programmer's Companion [17], [18] provides extensive conceptual and "how-to" development information in the Companion, and provides official reference information of Palm OS 68K functions and data structures in the reference. Fig. 25 gives descriptions of the Palm OS Programmer's Companion. Fig. 26 gives descriptions of the Palm OS Programmer's API (Application Programming Interface) reference of Palm OS 68K SDK [20]. It includes four major sections (i) user interface, (ii) system management, (iii) communications, and (iv) libraries.

Start ⇒ *All Programs* ⇒ *Palm Desktop* ⇒ *Palm Desktop*

after downloading and installing the *Palm Desktop* at http://www.palmos.com/dev/tools/desktop/.

Document	Description	URLs
Palm OS Programmer's API Reference	An API reference document that contains descriptions of all Palm OS function calls and important data structures	
Palm OS Programmer's Companion, Vol I & II	A multi-volume guide to application programming for the Palm OS. This guide contains conceptual and "how to" information that complements the Reference	
Constructor for Palm OS	A guide to using Constructor to create Palm OS resource files	
Palm OS Programming Development Tools Guide	A guide to writing and debugging Palm OS applications with the various tools available	

Fig. 24. Palm OS SDK Documentation.

Volume	Description
I	Gives fundamental knowledge of Palm OS programming such as event loop and user interface
II	Describes the handheld's communications capabilities such as Bluetooth and network communication

Fig. 25. Descriptions of the Palm OS Programmer's Companion.

Function	Description
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

Fig. 26. Descriptions of the Palm OS Programer's API Reference.

VII. CONCLUSION

Mobile commerce is a coming milestone after electronic commerce blossoming in the late 1990's. However, it is also commonly admitted that the development in this Held is constrained. There are some considerable barriers waiting to he overcome. One of the barriers is most software engineers are not familiar with handheld programming, which is the programming for handheld devices such as smart cellular phones and PDAs (Personal Digital Assistants). This article attempts to give a comprehensive study of handheld computing to help software engineers better understand this subject.

A. Summary

Five themes of handheld computing are covered in this paper:

- *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:* Mobile users use handheld devices to perform the mobile commerce transactions. A handheld device also 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: Handheld* computing is to use handheld devices to perform wireless, mobile, handheld operations such as personal data management and making phone calls. They can be achieved by using server- or client- side handheld computing and programming.
- *Server-side handheld computing and programming:* Server-side handheld computing is to use handheld devices to perform wireless, mobile, handheld operations, which require the supports of server-side computing. The most common server-side handheld applications are the mobile Web contents.
- *Client-side handheld computing and programming:* Client-side handheld computing is to use handheld devices to perform handheld operations, which do not need the supports of server-side computing. Various environments/languages are available for client-side handheld computing and programming. Five of the most popular are
  - *BREW:* It is created by Qualcomm Inc. for CDMAbased smartphones.
  - *.J2ME:* J2ME is an edition of the Java platform that is targeted at small, standalone or connectable consumer and embedded devices.
  - *Palm OS:* It is a fully ARM-native, 32-bit operating system running on handheld devices.
  - *Symbian OS:* Symbian OS is an industry standard operating system for smartphones, a joint venture originally set up by Ericsson, Nokia, and Psion.
  - *Windows Mobile:* Windows Mobile is a compact operating system for handheld devices based on the Microsoft Win32 API. It is a small version of Windows, and features many pocket. versions of popular Microsoft applications, such as Pocket Word, Excel, Access, PowerPoint, and Internet Explorer. They apply different approaches to accomplishing the development of handheld applications and it is almost impossible to predict which approaches will dominate the client-side handheld computing in the future, as the Windows to desktop PCs. Most client-side handheld programming languages are a version of either C/C++ or Java. This article step-by-step explains how to develop applications of J2ME, a version of Java, and Palm OS, using a version of C.

B. The Market Share of Mobile Operating Systems

A number of mobile operating systems, as shown in the previous list, with small footprints and reduced storage capacity have emerged to support the computing-related functions of mobile devices. For example, Rearch In Motion Ltd's BlackBerry 8700 smartphone uses RIM OS and provides Web access, as well as wireless voice, address book, and appointment applications [2 H. Because the handheld device is small and has limitcd power and memory, the mobile OSes. requirements are significantly less than those of desk- or laptop OSes. Although a wide range of mobile handheld devices

are available in the market, the operating systems, the huh of the devices, are dominated by just few major organizations. The following two lists show the operating systems used in the top brands of smart cellular phones and PDAs in descending order of market share:

- *Smart cellular phones:* Symbian OS, Linux, RIM OS, Palm OS, Windows Mobile-based Smartphone, and others [22].
- *PDAs:* Microsoft Pocket PC, RIM OS, Palm OS, Symbian OS, Linux, and others [23].

The market share is changing frequently and claims concerning the share vary enormously. It is almost impossible to predict which will be the ultimate winner in the battle of mobile operating systems. Because cach mobile OS has its own unique development tools and programming languages, handheld computing and programming, especially the client-side ones, become

difficult. A dominant mobile OS in the future may not be all that had as the Microsoft's Windows do to the PCs.

REFERENCES

1. D. Kiely. Wanted: programmers for handheld devices. IEEE Computer, 34(5):12-14, 2001.
2. W.-C. Hu, C.-w. Lee, and J.-h. Yeh. Mobile commerce systems. In Shi Nansi, editor, Mobile Commerce Applications, pages 1-23, Idea Group Publishing, 2004.
3. K. Geihs. Middleware challenges ahead. IEEE Computer, 34(61:24-31, 2001.
4. Open Mobile Alliance Ltd. WAP (Wireless Application Protocol). Retrieved September 21, 2006, from the World Wide Web: http://www.wapforum.org/.
5. NTT DoCoMo Inc. i-mode Technology. Retrieved July 2, 2006, from the World Wide Web: http://www.nudocomo.com/technologies/ present/imodetechnology/index.html, 2006.
6. W.-C. Flu, J.-h. Yeh, H.-J. Chu, and C.-w. Lee. Internet-enabled mobile handheld devices for mobile commerce. Contemporary Management Research, 1(11:13-34, 2005.
7. Open Mobile Alliance. Wireless Markup Language (WML), Version 2.0. Retrieved August ²8, 2006, from the World Wide Web: http://www.openmobilealliance.org/tech/affiliates/wap/wap-238-wm1- 20010911-a.pdf, ²001.
8. World Wide Web Consortium. Device Independent Authoring Language. Retrieved September 09, 2006, from the World Wide Web: http://www.w3.org/TR/dial, 2006.
9. L. D. Paulson (2006). DIAL, eases mobile-content development. IEEE Computer, 39(81:26.
10. Qualcomm Inc. BREW and J2ME-- A Complete Wireless Solution for Operators Committed to Java. Retrieved July 12, 2006, from the World Wide Web: http://brew.qualcomm.com/brew/en/img/about/ pdf/brew_j2me.pdf, 2003.
11. Symbian Ltd. Symbain OS Version 9.2. Retrieved May 20, 2006, from the World Wide Web: hup://www.symbian.com/technology/ sym-bianOSv9.2_ds_0905.pdf, 2005.
12. Microsoft Corp. What's Ness' for Developers in Windows Mobile 5.0,, Retrieved Rine 07, 2006, from the World Wide Web: http://msdn. microsoft.com / mobility / windowsmobile/ howto /documentation/default. aspx?pull=/library/en-usk/dnppcgen/html/whatsnew_wm5.asp, 2005.
13. Film, T. Developing Applications for Windows Mobile-based Smartphones. Retrieved September 1², 2006, from the World Wide Web: http://msdnmicrosoft.com/library/defaulLasrur1=/library/en-us/ dnppcgen/html/devappsp.asp, ²002.
14. Sun Microsystem Inc. Java 2 Platform, Micro Edition.
15. Retrieved March 15, 2006, from the World Wide Web: http://jaya.sun.com/j2Ineklocs/j2me-ds.pdf, 2002.
16. Sun Microsystem Inc. J2ME Wireless Toolkit 2.2—User's Guide. Retrieved April 21, 2006, from the World Wide Web: http://java. sun.com/j2me/docs/wtk2.2/docs/UserGuide.pdf, 2004.
17. Sun Microsystem Inc. Mobile Informarion Device Profile Specification 2.0. Retrieved May 25, 2006, from the World Wide Web: http://jcp.org/aboutJava/communityprocess/final/jsr118/, 2002.

18. PalmSource Inc. Why PalmOS? Retrieved June 23, 2006, from the World Wide Web: http://www.palmsource.com/palmos/Advantage/ index_files/v3_document.htm, 200².
19. PalmSource Inc. Palm OS Programmer's Companion, Vol. I. Retrieved February ²1, 2006, from the World Wide Web: http://www.palmos.com/ dev/support/docs/palmos/PalmOSCompanion/Companion'IOC.html. 2004.
20. PalmSource Inc. Palm OS Programmer's Companion, Vol. //. Retrieved February 21, 2006, from the World Wide Web: http://www.palmos.com/ dev/support/docs/palmos/PalmOSCoinpanion2/Companion2TOC.html, 2004.
21. PalmSource Inc. Palm OS Programmer's API Reference. Retrieved August 15, ²006, from the World Wide Web: http://www.pamos.com/ dev/support/docs/palmos/PamOSReference/ReferenceTOC.html, 2004.
22. Research In Motion Ltd. BlackBerry Application Control—An Overview for Application Developers. Retrieved April 24, 2006, from the World Wide Web: http://www.blackberry.com/knowledgecenterpublic/ livelink.exe/fetch/²000/7979/1181821/832210/Black Berry_Application_ Control _Over v iew_for_Developers .pdrnodeid=1106734& vernum=0, 2005.
23. Symbian Ltd. Fast Facts. Retrieved June 12, 2006, Iron the World Wide Web: http://www.symbian.com/about/fastfacts/fastfacts.html, 2006.
24. Gartner Inc. Gartner Says Worldwide PDA Shipments Increased .12 Percent in the Second Quarter of 2005. Retrieved January 13, 2006, from the World Wide Web: http://www.gartner.com/ press_releases/asset_ I 33230_1 I .html, 2005.