CLIENT-SIDE HANDHELD COMPUTING AND JAVA 2 PLATFORM, MICRO EDITION (J2ME)

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

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

Yapin Zhong
Computer Information Engineering Center
Shandong Institute of Physical Education and Sport, China
zhongyap@tom.com

Chung-wei Lee
Department of Computer Science and Software Engineering
Auburn University
Auburn, AL 36849-5347
chwlee@eng.auburn.edu

Abstract

Mobile commerce is a coming milestone after electronic commerce blossoming in the late 1990's. The success of mobile commerce applications is greatly dependent on handheld devices, by which mobile users perform the mobile transactions. Handheld computing is defined as the programming for handheld devices such as smart cellular phones and PDAs. It consists of two kinds of programming: client- and server- side programming. Various environments/languages are available for client-side handheld 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 handheld applications. This paper gives a detailed description of J2ME and briefly introduces the other four.

1. Introduction

Mobile commerce or m-commerce is defined as the exchange or buying and selling of commodities, services, or information on the Internet through the use of Internet-enabled mobile handheld devices (Hu, Lee, & Yeh, 2004). It is expected to be the next milestone after electronic commerce blossoming in the late 1990's. Internet-enabled mobile handheld devices are one of the core components of a mobile commerce system, making it possible for mobile users to directly interact with mobile commerce applications. Much of a mobile user's first impression of the application will be formed by his or her interaction with the device, therefore the success of mobile commerce applications is greatly dependent on how easy they are to use. However, programming for handheld devices is never an easy task not only because the programming languages and environments are significantly different from the traditional ones, but also because various languages and operating systems are used by handheld devices and none of them dominates.

This article gives a study of handheld computing, especially J2ME (Java 2 Platform, Micro Edition), for mobile commerce. Various environments/languages are available for client-side handheld 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. Three themes of this article are

1. Introduction of handheld computing, which includes server- and client- side computing.
2. Brief introductions of four kinds of client-side computing.
3. Detailed discussion of J2ME and J2ME programming.

Other important issues such as a handheld computing development cycle will also be discussed.

2. Background

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 computing are given as follows:

- *Client-side handheld computing*: It is the programming for handheld devices and it does not need the supports from server-side programs. Typical applications created by it include (i) address books, (ii) video games, (iii) note pads, and (iv) 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. Typical applications created by it include (i) instant messages, (ii) mobile Web contents, (iii) online video games, and (iv) wireless telephony.

This article will focus on the client-side computing. The server-side computing is briefly given next.

2.1 Server-side Handheld Computing

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 1. A database-driven mobile Web site is often implemented by using a three-tiered client/server architecture consisting of three layers:

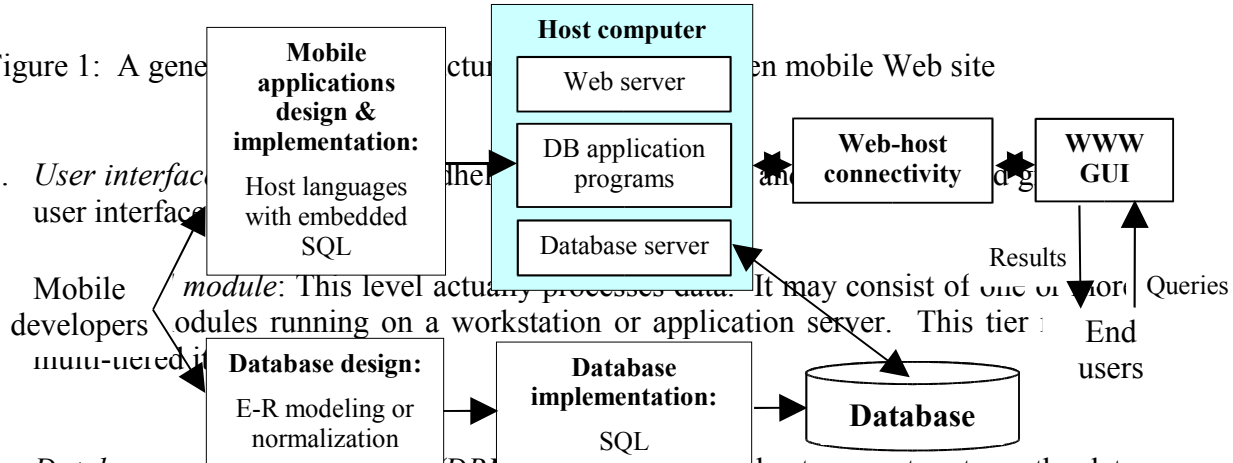


Figure 1: A gene ctu n mobile Web site

1. *User interfac* ther a d g user interfac

Mobile *module*: This level actually processes data. It may consist of one or more developers odules running on a workstation or application server. This tier multi-tiered i

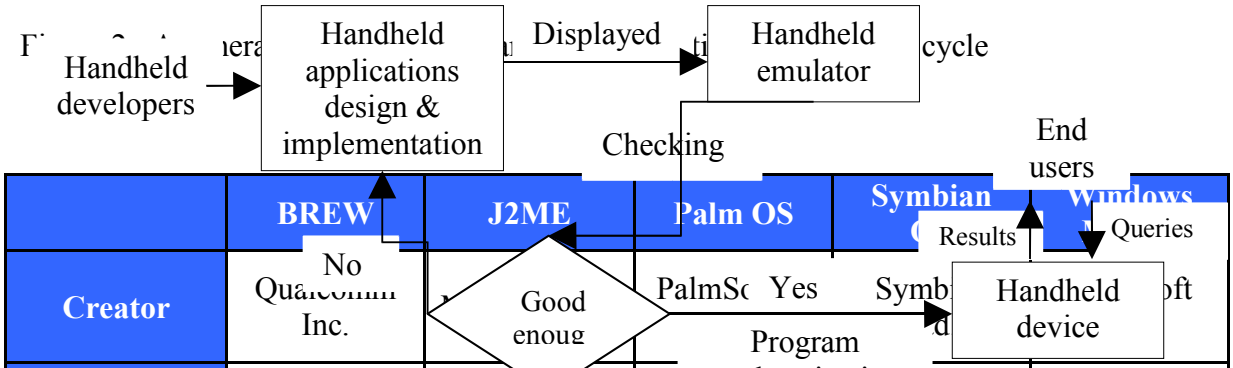
3. *Database management system (DBMS)*: A DBMS on a host computer stores the data required by the middle tier.

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. Client-side Handheld Computing

Various environments/languages are available for client-side handheld 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. Figure 2 shows a generalized development cycle applied by them

and Table 1 gives the comparison among the five languages/environments. The second half of this paper is devoted to J2ME details and brief introductions of the other four are given in this section.



	BREW	J2ME	Palm OS	Symbian	Windows
Creator	Qualcomm Inc.		PalmSc	Symb	oft
Language/ Environment	Environment	Language	E	onmen t	Environment
Market Share (PDA) as of 2004	N/A	N/A	2nd	N/A	1st
Market Share (Smartphone) as of 2005	?	N/A	3rd	1st	2nd
Primary Host Language	C/C++	Java	C/C++	C/C++	C/C++
Target Devices	Phones	PDAs & phones	PDAs	Phones	PDAs & phones

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

3.1 BREW (Binary Runtime Environment for Wireless)

BREW is an application development platform created by Qualcomm Inc. for CDMA-based mobile phones (Qualcomm Inc., 2003). CDMA is a digital wireless telephony transmission technique and its standards used for 2G mobile telephony are the IS-95 standards championed by Qualcomm. BREW is a complete, end-to-end solution for wireless applications development, device configuration, application distribution, and billing and payment. The complete BREW solution includes

- 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.

3.2 Palm OS

Palm OS, developed by Palm Source Inc., is a fully ARM-native, 32-bit operating system running on handheld devices (PalmSource Inc., 2002). 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. 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.

As of August 2005, no hardware products run Palm OS Cobalt and all devices use Palm OS Garnet. Likely as a result of Palm OS Cobalt's lack of adoption, PalmSource has shifted to developing Palm OS Cobalt's APIs on top of a Linux kernel.

3.3 Symbian OS

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). It is an independent, for-profit company whose mission is to establish Symbian OS as the world standard for mobile digital data systems, primarily for use in cellular telecoms. Symbian OS includes a multi-tasking multithreaded core, a user interface framework, data services enablers, application engines, integrated PIM functionality, and wireless communications. It is a descendant of EPOC, which is a range of operating systems developed by Psion for handheld devices.

3.4 Windows Mobile

Windows Mobile is a compact operating system for mobile devices based on the Microsoft Win32 API (Microsoft Corp., 2005). 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, including 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:

- *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.

- *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.
- *Portable Media Centers*: Portable Media Centers let users take recorded TV programs, movies, home videos, music, and photos transferred from Microsoft Windows XP-based PC anywhere.

3.4.1 Windows Mobile-based Pocket PCs

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 three 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.

3.4.2 Windows Mobile-based Smartphones

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.

4. 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 (Sun Microsystem Inc., 2002a). 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.

4.1 J2ME Architecture

The J2ME architecture 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. Each combination is optimized for the memory, processing power, and I/O capabilities of a related category of devices. The result is a common Java platform that takes full advantage of each type of device to deliver a rich user experience.

4.2 J2ME Programming

This sub-section gives an example of J2ME programming (Sun Microsystem Inc., 2004). Other client-side handheld programming is similar to this. Figure 3 shows the Sun Java Wireless Toolkit©, which is a toolbox for developing wireless applications that are based on J2ME's Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP), and designed to run on cell phones, mainstream personal digital assistants, and other small mobile devices. 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:

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/download-2_3.html*.
2. Run MIDlet, an MIDP application, development environment *KToolbar* as shown in Figure 3 by selecting the following Windows commands:

```
Start ► All Programs ► Sun Java Wireless Toolkit 2.3 Beta ►
   KToolbar
```

Figure 3: A screenshot of KToolbar after launching

3. Create a new project by giving a project name such as *HelloSuite* and a class name such as *HelloMIDlet* as shown in Figure 4. After the project *HelloSuite* is created, the KToolbar will display the message shown in Figure 5, which tells where to put the Java source files, application resource files, and application library files.

Figure 4: A screenshot of KToolbar after clicking on the button *New Project*

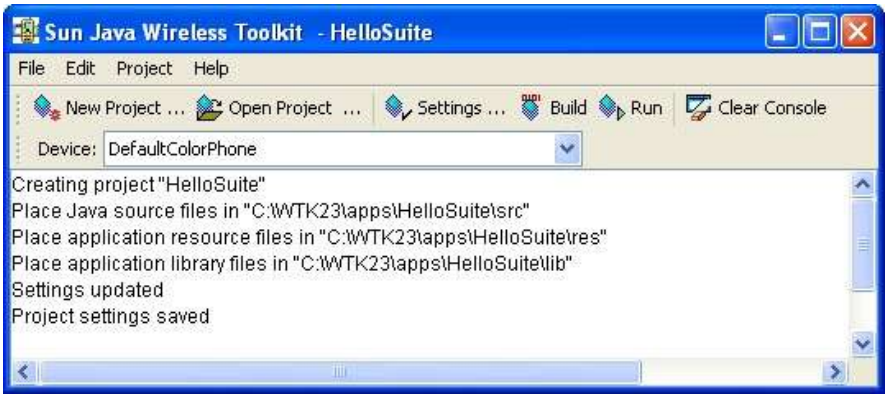


Figure 5: A screenshot of KToolbar after a project *HelloSuite* created

4. Create a J2ME source program and put it in the directory "C:\WTK23\apps\HelloSuite\src\." Figure 6 gives a J2ME example, which displays the text "Hello, World!" and a ticker with a message "Greeting, world."

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( );
  }
}
```

Figure 6: An example of an MIDlet program *HelloMIDlet.java*

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, Figure 7 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.

Figure 7: A screenshot of an emulator displaying the execution results of *HelloSuite*

4.3 Mobile Information Device Profile (MIDP) Packages

Table 2 gives the packages provided by the MIDP (Sun Microsystem Inc., 2002b). The packages *javax.** are the extensions to standard Java packages. They are not included in the JDK or JRE. They must be downloaded separately.

Package	Classes and Descriptions
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`**: The Mobile Information Device Profile provides a mechanism for MIDlets to persistently store data and later retrieve it.
Application Lifecycle	**`javax.microedition.midlet`**: The MIDlet package defines Mobile Information Device Profile applications and the interactions between the application and the environment in which the application runs.
Networking	**`javax.microedition.lcdui.io`**: 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.

Table 2: Mobile Information Device Profile (MIDP) package list

5. Conclusion

Mobile commerce is a coming milestone after electronic commerce blossoming in the late 1990's. The success of mobile commerce applications is greatly dependent on handheld devices, by which mobile users perform the mobile transactions. Handheld computing is defined as the programming for handheld devices such as smart cellular phones and PDAs. It consists of two kinds of programming: client- and server- side programming. Various environments/languages are available for client-side handheld programming. Five of the most popular are

1. *BREW*: It is created by Qualcomm Inc. for CDMA-based smartphones.
2. *J2ME*: J2ME is an edition of the Java platform that is targeted at small, standalone or connectable consumer and embedded devices.
3. *Palm OS*: It is a fully ARM-native, 32-bit operating system running on handheld devices.
4. *Symbian OS*: Symbian OS is an industry standard operating system for smartphones, a joint venture originally set up by Ericsson, Nokia, and Psion.
5. *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.

A number of mobile operating systems 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 (Research In Motion Ltd., 2005). Because the handheld device is small and has limited power and memory, the mobile OSs' requirements are significantly less than those of desktop OSs. Although a wide range of mobile handheld devices are available in the market, the operating systems, the hub 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, Microsoft Smartphone, Palm OS, Linux, and RIM OS (Symbian Ltd., n.d.).
- *PDAs*: Microsoft Pocket PC, Palm OS, RIM OS, and Linux (WindowsForDevices, 2004).

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.

References

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.

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

PalmSource Inc. (2002). Why PalmOS? Retrieved June 23, 2005, from *http://www.palmsource.com/palmos/Advantage/index_files/v3_document.htm*

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*

Research In Motion Ltd. (2005). BlackBerry application control—an overview for application developers. Retrieved January 05, 2006, from *http://www.blackberry.com/knowledgecenterpublic/livelink.exe/fetch/2000/7979/1181821/832210/BlackBerry_Application_Control_Overview_for_Developers.pdf?nodeid=1106734&vernum=0*

Sun Microsystem Inc. (2002a). Java 2 Platform, Micro Edition. Retrieved January 12, 2006, from *http://java.sun.com/j2me/docs/j2me-ds.pdf*

Sun Microsystem Inc. (2002b). Mobile Information Device Profile Specification 2.0. Retrieved October 25, 2005, from *http://sdlc-esd.sun.com/ESD3/JSCDL/midp/2.0-fr/midp-2_0-fr-spec.pdf?AuthParam=1140032198_f30ec3251f5b66844a6d06c2aba8d400&TUrl=LxU7fwxQLugy+ErpKYcvIWlVeSBzPG60JI8nt3z9f3UsBDtcUw13tW6+&TicketId=dlV2PwVJNeo5/Q==&GroupName=SDLC&BHost=sdlcweb9e.sun.com&FilePath=/ESD3/JSCDL/midp/2.0-fr/midp-2_0-fr-spec.pdf&File=midp-2_0-fr-spec.pdf*

Sun Microsystem Inc. (2004). J2ME Wireless Toolkit 2.2—user's guide. Retrieved October 21, 2005, from http://java.sun.com/j2me/docs/wtk2.2/docs/UserGuide.pdf

Symbian Ltd. (2005). Symbain OS Version 9.2. Retrieved December 20, 2005, from *http://www.symbian.com/technology/symbianOSv9.2_ds_0905.pdf*

Symbian Ltd. (n.d.). Symbian fast facts. Retrieved January 26, 2005, from *http://www.symbian.com/about/fastfaqs.html*

Wilson, J. (2005). What's new for developers in Windows Mobile 5.0. Retrieved January 14, 2006, from *http://msdn.microsoft.com/smartclient/default.aspx?pull=/library/en-us/dnppcgen/html/whatsnew_wm5.asp&print=true*

WindowsForDevices.com. (2004). Windows CE zooms past Palm. Retrieved August 23, 2005, form *http://www.windowsfordevices.com/news/NS6887329036.html*