Int. J. Mobile Communications, Vol. X, No. X, xxxx

The Technology Landscape of Wireless Web

Damianos Gavalas*

Cultural Heritage Management Laboratory (CHMLab)
Department of Cultural Technology and Communication
Harilaou Trikoupi & Faonos St., GR – 81100
University of the Aegean, Mytilene, Lesvos, Greece
E-mail: dgavalas@aegean.gr
*Corresponding author

Daphne Economou

Cultural Heritage Management Laboratory (CHMLab)
Department of Cultural Technology and Communication
Harilaou Trikoupi & Faonos St., GR – 81100
University of the Aegean, Mytilene, Lesvos, Greece
E-mail: d.economou@ct.aegean.gr r

Abstract: The traditionally separate technologies of the Internet and mobile computing have now started to converge, bringing promises for seamless wireless Internet access through portable devices. This article represents a comprehensive review of the main technological, architectural and business issues related to the current state-of-the-art wireless web technologies: WAP, i-mode and J2ME. A brief review of relevant application development platforms and authoring tools is also included, covering Microsoft solutions with emphasis on the Microsoft .Net platform for the Mobile Web and the Macromedia Flash Lite. The article also presents a critical analysis of the main assets and weaknesses of these technologies as well as their current status and the trends that will affect their market share and customer base in the foreseeable future.

Keywords: Wireless communication, Mobile Computing, Mobile Applications, Internet, Web technologies.

Reference to this paper should be made as follows: Gavalas, C. and Economou, D. (2006) 'The Technology Landscape of Wireless Web', Int. J. Mobile Communications, Vol. x, No. x, pp.xxx-xxx.

Biographical notes: Damianos Gavalas received his BSc degree in Informatics (Computer Science) from the University of Athens, Greece, in 1995 and his MSc and PhD degree in electronic engineering from the Electronic Systems Engineering Department, University of Essex, U.K., in 1997 and 2001, respectively. He is currently an Assistant Professor in the Department of Cultural Technology and Communication, University of the Aegean, Greece. His research interests include distributed computing, mobile code, network and systems management, e-commerce, m-commerce, application development for mobile/handheld devices, wireless ad-hoc and sensor networks.

Daphne Economou received her Ptychio in Graphic Arts Technology, School of Graphic and Arts, in 1993 from the Technological Educational Institute (TEI) in Athens, Greece, her MA. in Design for Interactive Media (Multimedia), in 1996 from the Centre of Electronic Arts, Middlessex University, London, UK. and her Ph.D. in Computer Science, in 2002 from the Department of Computing and

Copyright © 2007 Inderscience Enterprises Ltd.

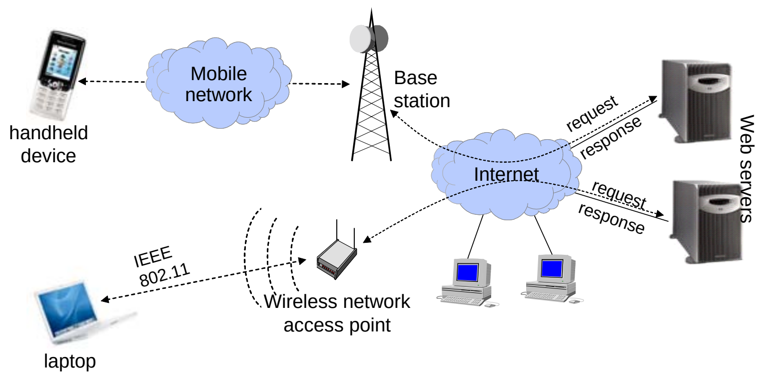
Mathematics, Faculty of Science and Engineering, Manchester Metropolitan University, UK. She is a Lecturer in Interactive Multimedia and Hypermedia at the Department of Cultural Technology and Communication, University of the Aegean since January 2004. Her research interest include: human-computer, user centred design for interactive multimedia and hypermedia, requirements gathering and design guidelines development for virtual reality environments and collaborative virtual environments for education and archaeology, content and service provision via mobile/handheld devices.

1 Introduction

The increasing demand for mobile data communications has led to the deployment of 3G mobile networks, offering higher throughput and basic multimedia services together with voice capabilities (Shim et al., 2006). Although worldwide migration to 3G has not been as fast as hyped, operators and service providers are already researching a true broadband wireless cellular system, known as 4G. In addition, wireless LANs and MANs have also evolved rapidly, complementing the wireless networking landscape (Shim et al., 2006).

The phenomenal growth of the Internet and the evolution of mobile wireless computing have naturally led to a convergence of these two worlds, giving rise to the wireless access to Internet resources by users of handheld devices. Within this relatively new paradigm, the wired Internet resources are still utilized, however through mobile terminals and a wireless network (either a wireless LAN or a mobile network), as shown in Figure 1; hence, Internet resources and services are available regardless of the end user's physical location (Varshney, 2003).

Figure 1 Information retrieval scenarios through a mobile or a wireless network.



Wireless computing presents many key characteristics that promote its wider adoption and growth: (a) ubiquity and convenience: mobile devices satisfy the need for real-time communication with no time and place constraints, (b) positioning: using technologies like GPS users may receive and access information and services specific to their location (Varshney, 2003), (c) personalization: handheld devices are typically operated by a single user, thereby enabling the provision of personalized services by wireless web portals (Ho and Kwok, 2003: Mahatanankoon *et al.*, 2006: Varshney *et al.*, 2004).

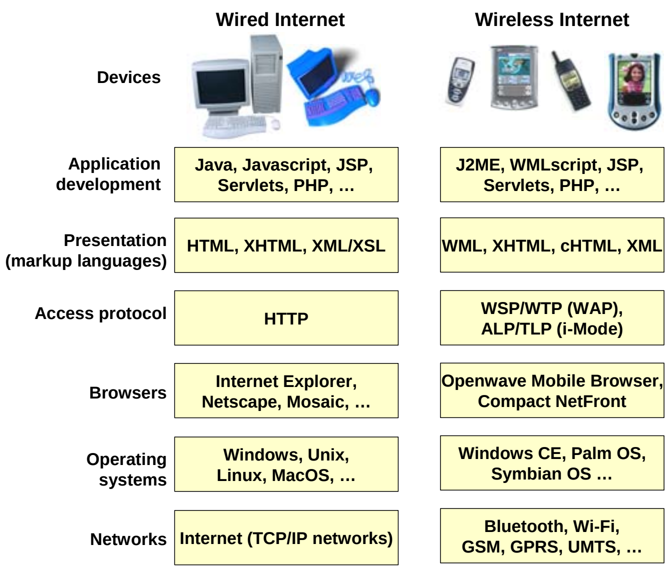
Some characteristics of mobile computing though, represent deterrent factors for its further expansion: (a) the physical characteristics of handheld devices (restricted energy capacity, small display size, limited color and font number support, small and hard to use keyboard, low processing capabilities and available memory) convey poor user experiences compared to the usage of PCs (AlShaali and Varshney, 2005: Andreou *et al.*, 2005), (b) limited bandwidth and high cost of wireless connections, (c) lack of innovative, killer applications tailored to mobile users.

Recently, technologies that use wireless infrastructures for Internet access have emerged realizing the vision of wireless Internet (Read and Maurer, 2003). The primary purpose of this paper is to review the status and current trends related to three existing technologies of the wireless web (WAP, i-mode and J2ME) and compare their main assets and shortcomings.

This article's main focus is on these technologies, acknowledging their popularity, penetration and high market share amongst wireless web users and developers. In particular, WAP, i-mode and J2ME represent the de-facto wireless web access protocol, service and development platform, respectively. Yet, wireless web is a highly dynamic field, which has lately attracted the attention of researchers and software developers, giving rise to many emergent, alternative technologies. Hence, in addition to the three abovementioned technologies, this article also reviews the key concepts and characteristics of relevant application development platforms and authoring tools (Microsoft solutions, focusing on the popular Microsoft .Net platform for the Mobile Web, and Macromedia Flash Lite) with emphasis on their networking capabilities.

A comparison of the technological status of wired and wireless Internet is illustrated in Figure 2, where features are categorized according to their corresponding functional layer (networks, operating systems, browsers, access protocols, markup languages for content presentation, programming technologies for application development). It is noted that a review of the technology status in relation with browsers software and operating systems of mobile devices is beyond the scope of this paper.

Figure 2 The technology landscape of wired and wireless Internet.



The remainder of the paper is organized as follows: Sections 2, 3 and 4 present WAP, i-mode and J2ME technologies, respectively, and discuss their main advantages and weaknesses. Section 5 summarizes the main characteristics of additional state-of-the art technologies: Microsoft .Net platform for the Mobile Web, Microsoft Smartphone and

Macromedia FlashLite. Section 6 discusses the factors that will influence the future of wireless Internet technologies and concludes the paper.

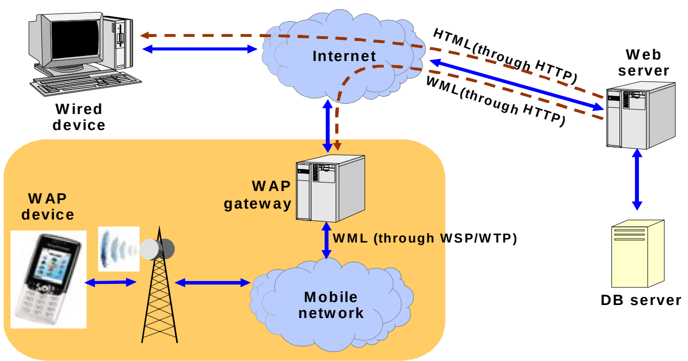
2 The Wireless Application Protocol (WAP)

WAP represents the first serious effort to emulate the success of the Internet in the wireless world. Backed by the entire telecommunication industry (through the WAP Forum) (WAP Forum, 2006), coupled with the fact that it combines two of the hottest innovations in recent times (mobile phones and the Internet), WAP raised high expectations (Kumar *et al.*, 2003: Read and Maurer, 2003).

The user of a WAP device 'calls' a WAP gateway through a mobile network and sends requests for web pages located on a WAP site. The principal role of the gateway is to translate requests from the WAP protocol stack to the WWW stack, so they can be submitted to web servers. The received data are then rendered for display by the mobile device's microbrowser. The architecture and operation of WAP is illustrated in Figure 3.

Similarly to the contents of a conventional web site, WAP sites contents are typically stored on a web server. However, they are developed in the Wireless Markup Language (WML), so that their format is tailored for presentation in small-sized screens and fast transfer over the mobile network. WML (WAP Forum, 2006) is an XML application; it follows a stricter syntax than HTML and supports WMLScript, a lightweight scripting language that enhances user interactivity with WML content. WML has been chosen instead of HTML because the latter includes information of such complexity and size that modern handheld devices can either not render or is too expensive to download.

Figure 3 WAP architecture and operation.



Essentially, WAP is a protocol stack optimized for low bandwidth wireless connections established by devices with slow processor, low memory and small screen size. The protocols comprising WAP stack are inspired by the 'wired' TCP/IP protocol stack. Hence, low-level protocols control data transfer over any mobile carrier, while high-level protocols specify the way that applications access wireless communication services. Mid-level protocols are directly mapped to Internet protocols, e.g. the Wireless Transfer Protocol (WTP) corresponds to TCP and the Wireless Session Protocol (WSP) to HTTP. Figure 4 presents screenshots from a mobile commerce WAP site developed in our laboratory for a Greek natural history museum shop (Zafeiri *et al.*, 2006). The application has been tested with the Openwave's WAP emulator (Openwave, 2006).

Figure 4 Screens of museum shop WAP site tested on the Openwave's WAP emulator: (a) intro page, (b) main menu, (c) product items list for a specific category, (d) description of a selected product.

(a) (b)

(c) (d)

The practical problems revealed with the usage of the WAP's first version (analyzed below) led to the standardization of two new versions of the protocol. Version 1.2 incorporated the 'push model', wherein the content may be pushed to mobile terminals with minimal user intervention. A more recent WAP version (WAP 2.0) leads to a reformation of WAP standard, adding support for content developed in a subset of XHTML, the XHTML MP (eXtensible HTML Mobile Profile). Moreover, in WAP 2.0, the content is transferred through HTTP, removing the requirement for WAP gateways that translate content among WSP and HTTP.

It is now acknowledged that WAP technology has been overhyped, unrealistically raising consumers' expectations although it has gained remarkable acceptance. WAP technology offers advantages like:

- It supports WML that has been specially designed for optimizing content presentation on terminals with small-sized screens and low processing capabilities.
- Data are 'compressed' (encoded) prior to their transfer to the requesting WAP device so as to reduce the delay experienced by users connected through slow wireless links.
- The lightweight protocol stack comprising WAP is specially designed for minimizing the consumption of wireless bandwidth and making WAP standard independent of the underlying mobile network (e.g. GSM, GPRS, 3G).

However, WAP is considered a commercial failure, mainly owned to its inherent weaknesses:

- WAP 1.0 (which still maintains the largest user base among WAP compatible devices) has been mainly operated over circuit-switched systems, wherein charges apply on connection time basis. That represents a major counterincentive since users are not keen on surfing the net with a tiny screen over unacceptably slow and overrated wireless connections (Palomäki, 2004). In contrast, WAP 2.0 may operate over packet-switched (e.g. GPRS, 3G) networks taking advantage of their higher transmission rates and their billing policy (pay-per-kilobyte instead of pay-per-minute).

- WAP typically operates on GSM networks with transmission speed of 9.6 Kbps, wherein users need to dial the gateway's number to get online. Thus, it might take several minutes to access a WAP site, especially during peak hours. Together with the small screen size and the hard-to-use keypad, WAP conveys an overall usage experience which is unacceptable for traditional Internet users (Palomäki, 2004).
- WAP devices may only access pages 'translated' to WML, thus excluding the vast majority of web content. Besides, WML poses restrictions on the maximum file size, while WML pages are not straight-forward to author as WML follows a strict syntax (possible syntax errors are not overlooked by WAP browsers, as in the case of HTML).
- WAP gateways represent a security hole since encrypted data are decrypted on their passage over the gateway and re- encrypted prior to their re-transmission (this problem has been addressed in WAP 2.0) (Kumar *et al.*, 2003: Varshney *et al.*, 2004).

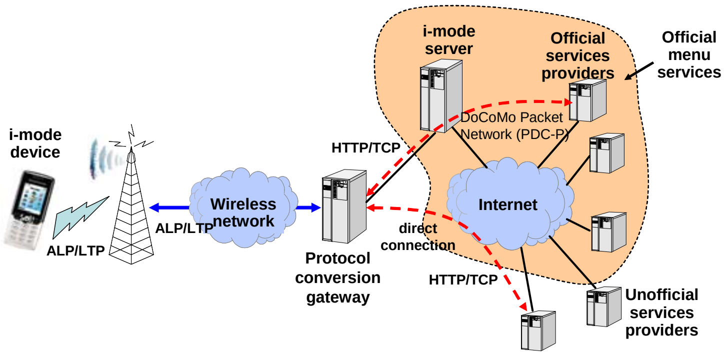
The industrial giants that back the WAP Forum and the technological enhancements of WAP 2.0 are expected to increase the adoption of WAP technology in the near future. WAP is an application protocol suite designed to function over any bearer service. Currently, most WAP service providers still support WAP 1.x, version, however, version 2.0's backward compatibility should lead to relatively rapid upgrades. These upgrades also make sense since the industry is moving towards adopting XML-based formats for almost every kind of data interchange. Through supporting XHTML MP, WAP 2.0 has made a step forward towards web compatibility (XHTML is expected to replace HTML as the de-facto web standard) and simplified the rapid development of WAP content. Finally, the operation of WAP over packet-switched networks will decrease the cost of WAP services and will probably broaden WAP's customer base.

3 i-mode

i-mode (NTT DoCoMo, 2006) is a more recent, alternative to WAP, approach for the wireless Internet, which addresses the main weaknesses revealed with the practical implementation of WAP. In principle, i-mode is a service of wireless Internet (in contrast with WAP which is a protocol stack), with large subscriber base in Japan. It has been released by the Japanese NTT DoCoMo in 1999. The phenomenal success of i-mode is mainly owned to its carefully designed services and business model. The data transfer is carried out by DoCoMo proprietary protocols: ALP (corresponding to HTTP) and LTP (corresponding to TCP). Initially, the device connects to protocol conversion gateway (translates packets among LTP and TCP). The gateway maintains a broadband connection with the i-mode server and returns the "official" (approved by NTT DoCoMo) services menu. Requests for unofficial services bypass the i-mode server and are directly routed – through the Internet– to the corresponding service providers (see Figure 5).

Unlike WAP, i-mode borrows from successful web document markup standards and supports the cHTML (Compact HTML) (cHTML, 2006) language which is based on HTML. cHTML is designed with the restrictions of the wireless infrastructure in mind, such as limited bandwidth connections and devices with small screens and limited functionality. By removing certain 'heavy-weight' features of conventional HTML the speed of content delivery is substantially increased (albeit reducing display sophistication). Recently, i-mode has made a further step towards compatibility with web standards, adding support for XHTML Basic (based on the WAP 2.0's XHTML MP standard, with the addition of some elements from the full version of XHTML).

Figure 5 The i-mode system's architecture.



In Japan, i-mode uses a proprietary packet-switched network (PDC-P) and employs a pricing policy based on the volume of transferred data (Grech, 2004). Thus, the i-mode service is continuously activated, i.e. the users are always 'online'. It is noted that i-mode is independent of the underlying network infrastructure, for instance in Europe, i-mode implementations run over GPRS networks.

WAP and i-mode are alternative technology solutions for the wireless web. However, i-mode is considered a major commercial success as opposed to WAP which is regarded as a failure. The main reasons for that i-mode's success are:

- i-mode uses cHTML which, as a subset of HTML, simplifies content development and is certainly more compatible with existing web content (Ishii, 2004).
- i-mode operates over packet-switched networks, hence, its users are 'always online' and may surf in i-mode sites without worrying about the connection duration, due to the 'pay-per-kilobyte' pricing policy (Grech, 2004: Ishii, 2004).
- Tens of thousands of i-mode (official and unofficial) sites currently exist (most of them are unofficial), offering a broad range of services (Ishii, 2004).
- The upper limit for cHTML pages size is 5 Kbytes (although sizes up to 2 Kbytes are recommended), considerably higher to the 1.4 Kbytes limit of WML.

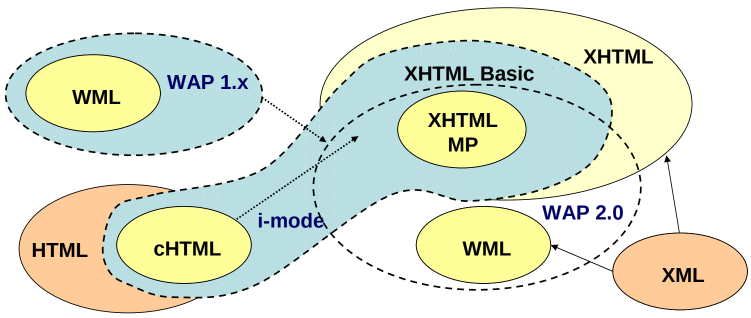
However, there are some disadvantages of i-mode compared with WAP:

- cHTML does not support a script language, while WML is accompanied by WMLscript.
- i-mode is a monopoly of NTT DoCoMo. Thus, i-mode devices need to comply with the specifications determined by this company which entirely controls the evolution of i-mode. In contrast, WAP technology is telecom operator and device-independent since the WAP Forum includes more than 500 members, practically all the important players in the wireless arena.

Last, a common disadvantage of the two competing technologies is that they both require constant connection (airtime) of the mobile device with a bearer network to offer access to Internet resources. Thus, when a user is out of coverage of the mobile network (i.e. 'has no signal') he/she cannot have access to any service.

The question "which of the two (WAP or i-mode) will prevail" is not easy to answer, not only because of their rapid evolution and the fluidity in the field of wireless communications. Certainly, i-mode will significantly influence the future of wireless Internet technologies, mainly due to its current popularity. Hence, one of the two competitors may prevail; maybe none, since the possibility of a co-operation among the WAP Forum and NTT DoCoMo towards the specification of a common standard cannot be overestimated. Current trends though demonstrate that the two technologies converge to supporting subsets of the XML-based XTML language (XHTML MP for WAP 2.0, XHTML Basic for i-mode). The convergence of wireless web markup languages is shown in Figure 6.

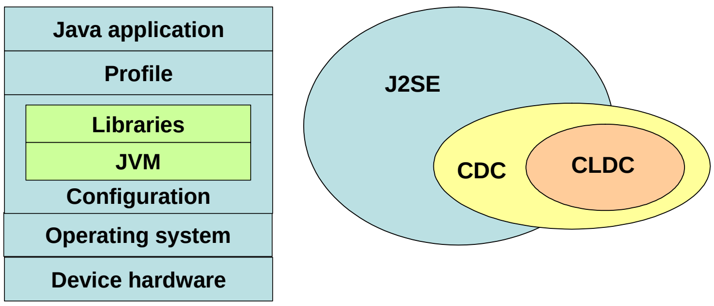
Figure 6 The convergence of wireless web markup languages.



4 J2ME (Java 2 Micro Edition)

The need for defining a computing platform that could execute Java applications and be supported by small electronic and embedded devices led to the development of J2ME (J2ME, 2006) by Sun Microsystems, in 1999. J2ME is the "little brother" of Java technologies family (J2EE and J2SE editions cover the needs of distributed server environments and PCs/laptops, respectively). It should be emphasized that J2ME is neither a protocol stack (like WAP) nor a service (like i-mode), but a framework for developing applications executed on resource-constrained devices. However, J2ME platform has strong wireless networking support and enables programming applications able of accessing a broad range of web content formats; therefore, J2ME is relevant to this article, as it may be considered as an (alternative to WAP and i-mode) wireless Internet technology.

Figure 7 (a) The five levels of the modular J2ME architecture; (b) The relation among J2SE, CDC and CLDC.



(a) (b)

J2ME follows a modular design (see Figure 7a) which aims at simplifying the support for a broad range of small devices. Thus, it introduced the concepts of *configuration* and *profile*. A J2ME configuration defines a minimum Java platform for a 'horizontal' family of devices, i.e. devices with similar processing and memory limitations, user interface requirements and connection capabilities. A configuration, hence, specifies the minimum features of a Java Virtual Machine (JVM) and a minimum set of libraries (groups of Java classes); that way, a developer knows that s/he can make use of these libraries to build applications intended for devices that implement a particular configuration. Today, there exist two different configurations: CDC (Connected Device Configuration) for devices with sufficient memory and computational resources, and CLDC (Connected Limited Device Configuration) which is subset of CDC referring to smaller, mobile devices (see Figure 7b).

CLDC configuration supports personal, mobile devices (much less powerful than CDC supported devices[1]), with characteristics: (a) 160-512 KB of total memory dedicated to the Java platform, (b) 16-bit or 32-bit processor, (c) restricted energy capacity, (d) wireless connection capabilities, possibly through low transmission rates. Due to the broad range of devices encompassing the abovementioned characteristics, CLDC follows a “minimum common denominator” policy, wherein a minimum set of capabilities is assumed. It is noted that none of CDC and CLDC are subsets of J2SE since these two configurations include additional libraries for executing applications within the device families they represent. The JVM integrated into CLDC implementation is called Kilobyte Virtual Machine (KVM) since it requires only a few KB of available memory. KVM is not a complete J2SE virtual machine.

Profiles are implemented on the top of configurations (a configuration is the base for one or more profiles). Typically, a profile includes libraries specialized in the unique characteristics of a particular class of devices. Java applications are built on the top of a particular configuration – profile pair and based on their specified class libraries. At the time these lines were written, the only available profile specified on the top of CLDC was MIDP (Mobile Information Device Profile). The Java applications developed over MIDP profile (and CLDC configuration) are called MIDlets, usually packaged in *.jar files.

The management of MIDlets execution as well as their installation / un-installation is controlled by the Application Management Software (AMS) running on the same device and is typically provided by manufacturers. J2ME applications based on MIDlets are packaged in *.jar files (compression, similar to zip). A MIDlet jar usually comes along with *.jad files (Java Application Descriptors) that provide application deployment descriptions (name, version, size, etc). An example MIDlet application and the result of its execution by an emulator are shown in Figure 8. Evidently, the development of J2ME applications is more complex compared to authoring WML, cHTML or XHTML pages; yet, J2ME provides means for developing powerful interactive applications since they are based on an extensive subset of the powerful Java language and not on a markup language.

MIDlets are either downloaded on-the-fly from a web server (via a wireless link) or through a bluetooth connection from a PC and executed as standalone applications with no requirement for constant connection to a wireless network; however, they are capable of connecting and interacting with web sites downloading information on-demand. The communication of MIDlets with web servers is carried out over the Internet’s HTTP 1.1 protocol.

[1] CDC configuration is intended for 64-bit devices with total memory (RAM + ROM) of at least 2 MB. Since this class of devices is not mobile, CDC does not fall within the scope of this paper and is not further examined.

Figure 8. An example MIDlet application and the result of its execution shown by an emulator.

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.MIDlet;

public class TinyMIDlet extends MIDlet
    implements CommandListener {
  public void startApp() {
    Display display = Display.getDisplay(this);

    Form mainForm = new Form("GUI");
    mainForm.append("Welcome to the world of
      MIDlets!\n");
    mainForm.append("Hello World!");

    Command exitCommand = new
      Command("Exit", Command.EXIT, 0);
    mainForm.addCommand(exitCommand);
    mainForm.setCommandListener(this);
  }

  public void pauseApp () {}

  public void destroyApp(boolean
      unconditional) {}

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

J2ME exhibits many similarities with current standards of wireless web as it enables the retrieval of web content. J2ME's advantages are summarized in the following:

- J2ME inherits the assets of Java language: the capacity to develop powerful applications, platform independence (execution on any device supporting CLDC/MIDP regardless of the underlying hardware or the operating system), etc.
- J2ME advances one step further from WAP and i-mode: apart from simply browsing content, the user can download over the air full-fledged applications (based on an extensive subset of Java programming language rather than on a markup language).
- J2ME supports the Internet security standard HTTPS (HTTP Secure) which enables the encryption and end-to-end secure transfer of sensitive information (e.g. in mobile payment transactions).
- J2ME applications can practically download and parse any content format, e.g. text, XML, WML, XHTML, serialized Java objects, etc. Of course, the presentation of content authored in a markup language requires the use of specialized parsers.
- Developers can implement interactive applications with richer graphics that offer enhanced user experience. Because graphics are typically generated locally, network bandwidth demand is reduced (Gupta and Srivastava, 2001).
- J2ME enables the communication of applications with web servers over the HTTP/S protocol independently of the underlying network protocols. The connection may be established using the wireless TCP, WTP (WAP) or LTP (i-mode) transport protocols. Thus, J2ME utilizes the existing networking infrastructures and may also co-exist with other wireless Internet technologies on the client tier.
- J2ME enables disconnected access and synchronization. Java-based mobile applications can run even when their hosting device is disconnected or out of the

coverage area. The user can run and interact with applications in standalone mode, and later synchronize with the backend infrastructure. This is in contrast with WAP and i-mode that require constant connection with the mobile network (Gupta and Srivastava, 2001).

J2ME though presents several disadvantages that should not be overlooked:

- The development of J2ME applications is certainly more complex than authoring WML or cHTML pages and requires Java programming skills.
- Existing web (HTML) content should be translated to a format accessible by J2ME applications.
- The download of new J2ME applications (jar files) is slow and consumes network resources. Also, these applications have increased requirements on devices' resources: storage, processing power and memory.
- Unlike WML and cHTML content which is rendered by compatible microbrowsers, J2ME applications display information on a graphical user interface (GUI) implemented upon specific APIs, part of the MIDP profile. This GUI is proprietary, namely it displays content tailored to itself (in contrast, any valid WML / cHTML file can be displayed by a microbrowser).

J2ME technology is still evolving. Apart from the standard APIs supported by CDLC/MIDP devices, a series of (optional) APIs have been proposed and currently are under implementation. Examples are the "Location API for J2ME" (facilitates the implementation of location-based services), the "Mobile 3D Graphics API for J2ME" (use of lightweight 3D graphics such as games and animated messages), etc (J2ME, 2006).

J2ME provides a vehicle for creating complex applications with higher degree of interactivity compared to their WAP or i-mode counterparts. Finally, the independence of J2ME communication from the underlying transport protocol is expected to lead to a synergy of J2ME and WAP/i-mode worlds wherein, for instance, WAP will provide the transport mechanism and J2ME the functionality.

5 Additional Wireless Web Technologies: Development Platforms and Authoring Tools

This section includes a brief description of emergent wireless web technologies, relevant to the popular WAP, i-mode and J2ME technologies discussed in the previous sections. In particular, the Microsoft technologies for mobile application development are reviewed with special focus on the popular Microsoft .Net platform for the Mobile Web. In addition, Macromedia Flash Lite mobile applications authoring environment is presented.

5.1. Microsoft Technologies for Mobile Application Development and Deployment

Microsoft's entering the mobile market has been characterized by the release of a proprietary operating system, namely, Microsoft Windows Mobile 2003 (Microsoft Windows Mobile, 2006) and the provision of developer support to program mobile devices. Specifically, a subset of the rich .NET Framework, called Microsoft .NET Compact Framework, provides a runtime engine preloaded in the device's memory in order to facilitate mobile application deployment. Therefore, Windows Mobile becomes an enabling platform for development and deployment of mobile applications supporting a variety of handheld devices and Wireless/Mobile Operators (see Figure 9). In particular:

- Pocket PCs can run mobile versions of Microsoft Office or other Windows Mobile supported third-party applications and offer Internet connectivity through Wi-Fi and wireless hotspots.
- Pocket PC Phones extend Pocket PCs functionality providing mobile phone capabilities and Internet access over the mobile network (GPRS, 3G, etc)
- Smartphones provide standard phone capabilities and support a smaller set of applications.

Most of these handheld devices support Bluetooth or infrared which allows their interaction with more Bluetooth/infrared-enabled peripherals.

Figure 9 Windows mobile applications displayed on (a) O2 XDA; (b) Motorola MPx200; (c) Toshiba E805.

(a) (b) (c)

Microsoft have also released a specific Software Development Toolkit, namely, Smartphone 2003 SDK (Lee, 2004), to support development of mobile applications for handheld devices. Smartphone 2003 SDK integrates with .NET Compact Framework facilitating mobile application development using Visual Studio .NET[2]. There is however an inherent limitation when designing applications for the Smartphone or the Pocket PC that relates to the different ways user input is entered into the device. Pocket PCs and

---

[2] Microsoft Visual Studio (Microsoft Visual Studio, 2006) is an integrated development environment that assists programmers to create programs, web sites, web applications and web services that run on various platforms. Supported platforms include Microsoft Windows servers and workstations, Pocket PC, Smartphones, and web browsers. Visual Studio includes support for several programming languages, such as Visual Basic, Visual C++, Visual C# and Visual J#.

Pocket PC Phones provide a stylus and a touch-sensitive screen, whilst Smartphones come with a keypad and have a smaller screen. This imposes design constraints related to the Application User Interfaces per device type. The following figure offers an indication of the different Look and Feel per device type.

The following subsection focuses on a popular mobile applications development platform of Microsoft, the .Net platform for the Mobile Web.

5.2. Microsoft .Net platform for the Mobile Web

The ASP.NET mobile controls (ASP.NET Mobile Controls, 2006) (formerly known as the Microsoft Mobile Internet Toolkit, MMIT) represent a mobile application development platform, recently released by Microsoft. They are fully integrated within Visual Studio and extend the power of the Microsoft .NET Framework to build mobile web applications by enabling ASP.NET to deliver markup content to a wide variety of mobile devices.

In particular, the ASP.NET mobile controls provide an easy way to build mobile web applications that generate the appropriate markup language (WML, xHTML, HTML or cHTML) and rendering for web-enabled cell phones, WAP phones, PDAs, Pocket PCs and pagers. The programming of ASP.NET mobile controls is enabled by the Mobile Internet Toolkit (MIT)[3] development environment (see Figure 10). The main asset of MIT is that it provides server-side mobile controls (including user interface elements such as list, command, call, calendar, etc.) with rich device identification mechanisms; developers simply utilize ASP.NET pages (for no particular target device) which automatically identify the device that posted a request[4] and render the appropriate content.

Summarizing, the main strengths of ASP.NET mobile controls are:

- no need to perform browser checks and deliver the appropriate content based on the target device (this makes an application faster to develop and easier to maintain);
- developers only need to learn ASP.NET and .NET mobile controls (no need for markup language authoring skills);
- easy to use programming model and drag-and-drop application development with Visual Studio.NET.

In contrast, the main limitations of this technology are:

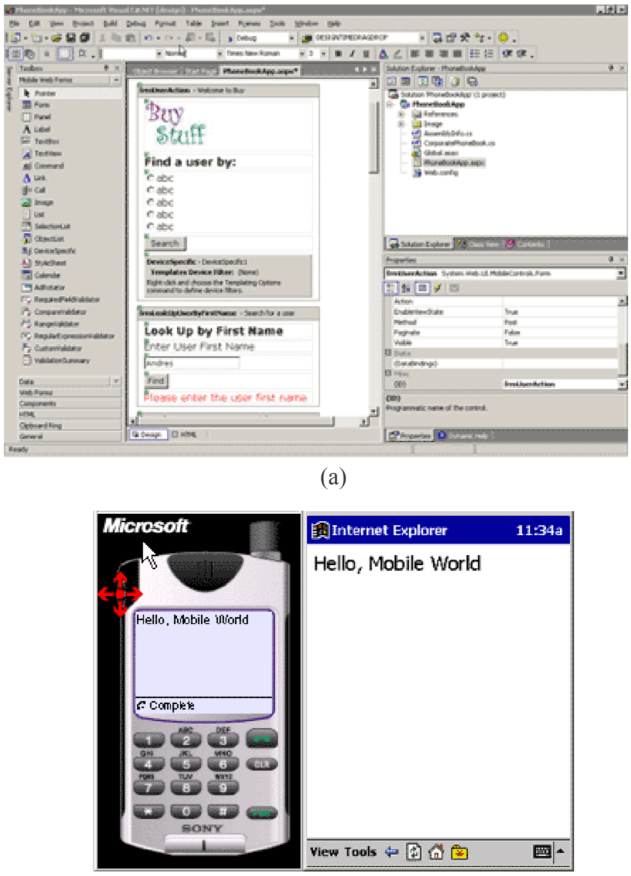
- the target devices are limited to Microsoft products and operating systems (unlike the J2ME platform-independent applications);
- when a new version of WML or HTML is released, developers need to wait until Microsoft announces support for the new version within its .NET mobile controls.

---

[3] The Mobile Internet Toolkit contains, among others: (a) Mobile Web Forms Controls that generate markup language for different devices; (b) the Mobile Internet Designer that works with the Visual Studio .NET to provide a drag-and-drop mobile development environment with Visual Development Tools; (c) browser capabilities, rich enough to extend ASP.NET device capabilities to mobile devices.

[4] Accurate information about the display capabilities of the target device is essential for the successful rendering of mobile controls. At a minimum, mobile controls need the following information about a device: markup language (HTML, WML, cHTML), browser, number of display lines, cookie support, screen size.

Figure 10 (a) A mobile application developed in Visual Studio .NET with the Mobile Internet Toolkit; (b) rendering the "Hello World" program on a cell phone and a Pocket PC.



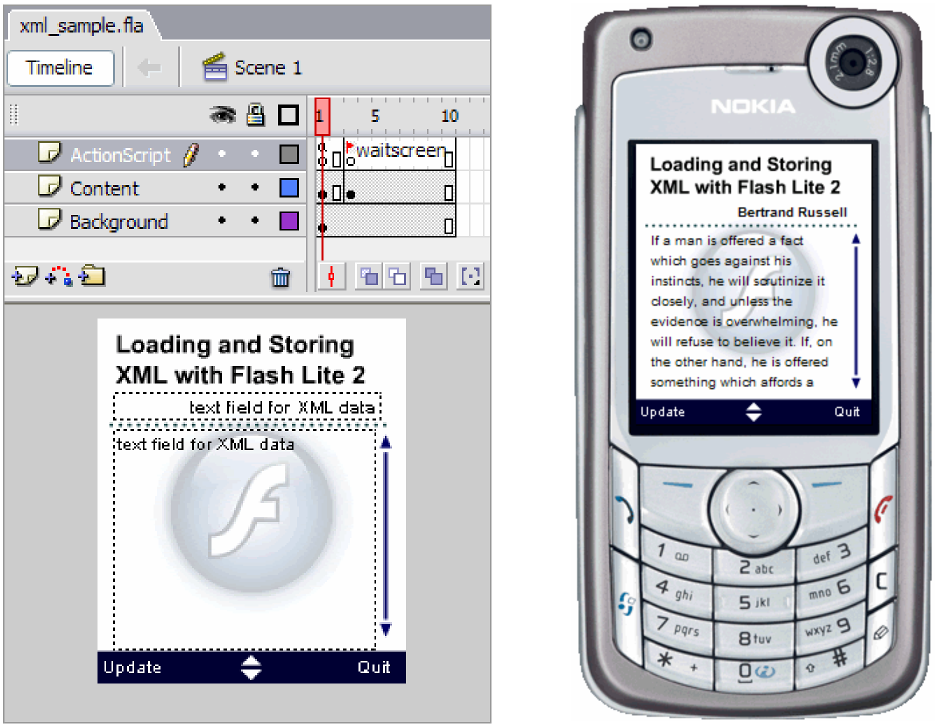
(a)

(b)

5.3. Macromedia Flash Lite

Macromedia Flash Lite (Adobe, 2006) is one of the most commonly used multimedia authoring tools specifically created to enable companies to easily and rapidly deploy content to mobile devices. There has been explosive adoption of Flash Lite by Original Equipment Manufacturers (OEMs), operators and developers which is quickly growing worldwide. This growth is driven by a variety of causes. The Flash Lite authoring environment (as shown in Figure 11) provides the designers and developers a new level of expressiveness, efficiency and interactivity for content creation. In addition, the Flash Lite rendering engine (Flash Player SDK 7 to date) is optimized for consumer electronic devices, enabling consumer electronics manufacturers, system integrators and browser companies to create high impact products and services, with full web browsing capabilities that leverage the vast number of Internet sites featuring Flash content. Another cause for its quick adoption by mobile technologies industry players is that developers already skilled in working with Flash MX Professional can easily switch into using Flash Lite to design applications for mobile devices. The reduction of the amount of technical knowledge required for the creation of applications and content for mobile devices, allows a wider community of developers to enter and compete in the mobile world.

Figure 11 Screens of a Flash Lite authoring environment and the result of its execution shown by an emulator.



Flash Lite provides a small and light profile for memory and processor-restricted mass-market mobile phones. By acting as a layer which overlays existing operating systems and application environments it allows a reduction in complexity through the creation of a firm, uniform foundation upon which to build advanced applications and content services. This layered design approach allows the direct deployment of Flash Lite content across handsets and it enables the implementation of a standard UI across these devices, which can be modified separately from the actual Flash Lite player implementation, allowing for rapid customization and over-the-air updates.

Macromedia Flash Lite coupled with other developers' products and services (like QUALCOMM's BREW (QUALCOMM BREW, 2006)) targets at addressing the distinct and varied needs of wireless operators, handset manufacturers, publishers, developers and end-users. These products and services include: open, extensible client platform that supports robust system and application software including personalized and branded user interfaces for mass-market devices; a J2EE-based, modular delivery system that enables the distribution of content, applications, and user interfaces to wireless devices across all air interfaces; a dedicated professional services team that supports the integration of customized implementations; and the wireless marketplace to support the monetization of applications and services developed in all programming languages.

Flash Lite uses the most common controls (including menus, check boxes, and radio buttons) and it makes a request to a server using HTTP, offering the great advantage of expediting the delivery of advanced applications and content services. However, it is not open-source, is does not support dynamic content maintenance and it requires MS Windows compatible devices for the development of multimedia projects, as well as for the run-time.

6 Discussion & the Future of Wireless Internet

It has been almost a decade since WAP Forum was founded and the traditionally separate technologies of the Internet and mobile computing started converging. Within this period, three major wireless web technologies have emerged: WAP, i-mode and J2ME. Current trends push these technologies towards XML-based languages for content development and presentation and web protocols for data transfer (end-to-end HTTP) and security (HTTPS, SSL). This remarkable convergence to web-compatible standards brings us closer to the vision of unifying the wired and wireless web landscapes and enabling the seamless access of mobile terminals to real web content.

Despite the progress already made, sufficient ground still needs to be covered. In this section, we identify the major factors expected to influence the future of wireless Internet. First, the evolution of broadband wireless networks, that is, the pricing policy of mobile operators, the transmission rates and the process of 3G networks deployment will affect the trail of wireless Internet. Thus, the users of 3G networks will enjoy faster wireless connections, be always online and be able to use location-based services (Varshney, 2003). Second, the design of a new generation of devices (with capabilities of managing any format of multimedia content, simplified user input and content browsing) will also enhance the perspectives of wireless web. Another important factor is the development of a new (or the domination of an existing) wireless web access standard. In addition, the convergence towards a single markup language, entirely compatible with wired web standards (current trends on web technologies would mandate the use of a simplified version of XHTML for that role); a global web content standard would boost the development of services accessible by mobile devices and simplify the conversion of existing content.

Finally, the evolution of mobile services and applications will also play a significant role. Alongside the increased transfer rates offered by new generation mobile networks, the first applications that take advantage of these higher rates will emerge (multimedia presentations, telemedicine services, mobile teleconference, interactive entertainment, high-quality music download, etc). This new class of applications will profit from the growth of content and sites accessible by mobile devices (mobile portals). Last, but not least, it is expected that innovative services will emerge to capitalize the unique characteristics of mobile devices, e.g. provide location-based services. These services should also satisfy requirements like: (a) usability, (b) user interfaces intended for users not necessarily familiarized with technology and/or PCs and Internet usage (AlShaali and Varshney, 2005).

A summary of the main features and comparison among the main three wireless web technologies reviewed in this article may be found in Table 1.

Table 1 Comparison of the three major wireless Internet technologies.

	WAP	**i-mode**	**J2ME**
Type of technology	Protocol stack	Service	Set of Java APIs for application development
Standardization	Open standard of WAP Forum	Proprietary (monopoly of NTT DoCoMo)	CLDC/MIDP specifications released by Sun Microsystems and supported by almost all mobile device manufacturers
Target devices	Mobile phones, PDAs, palmtops	Mobile phones	Mobile phones, PDAs, palmtops
Accessible content format	WML, XHTML MP (WAP 2.0)	cHTML, XHTML Basic (will soon be supported)	any (text, XML, WML, cHTML, HTML, XHTML, serialized objects, …)
Client tier technology	WML-compatible microbrowser	cHTML-compatible microbrowser	J2ME application (CLDC, MIDP), XML parser
Pricing policy	per minute (per kilobyte when implemented over GPRS/3G networks)	per kilobyte	per kilobyte
Support for location-based services	No	Yes (identification of the region where the user is located)	Yes (precise location identification though the optional 'Location API')
Transport mechanism between client and server	WSP over WTP (end-to-end HTTP in WAP 2.0)	ALP over LTP	HTTP/HTTPS over any transport protocol
Support for disconnected operation	No	No	Yes
Support for scripting	Yes (WMLscript)	No	Not needed (based on Java)
Security	Security hole at the WAP gateway where protocol conversion is done (resolved in WAP 2.0)	Support for SSL	End-to-end security (support for HTTPS)
Compatibility with existing web content	Incompatibility of WML (content translation required), enhanced compatibility with XHTML MP (WAP 2.0)	Satisfactory compatibility of cHTML, enhanced compatibility with XHTML Basic	Content translation required
Simplicity of application development	WML enforces strict syntax; content is not straight-forward to develop even for web developers	Easy, especially for web developers	Difficult (Java programming skills required)
Content presentation	Standardized (content readable by WML browsers)	Content readable by cHTML browsers, but cHTML is not a	Proprietary (content format intended for a particular application)

		W3C standard	
Support of push model	Yes (in WAP 1.2)	Possible (e.g. SMS notifying for incoming email)	Yes (in MIDP 2.0)
Regions with high penetration	Europe, Japan	Japan	Everywhere
Typical user	Professional	Young person (for leisure)	Young person (for leisure)
Popular applications	News, finance	E-mail, games	Games

References

Adobe (2006), Macromedia Flash Lite, http://www.adobe.com/products/flashlite/.

AlShaali, S. and Varshney, U. (2005), "On the Usability of Mobile Commerce", International Journal of Mobile Communications, Vol. 3, No. 1, pp. 29 – 37.

Andreou, A., Leonidou, C., Chrysostomou, C., Pitsillides, A., Samaras, G. and Schizas, C. (2005), "Key Issues for the Design and Development of Mobile Commerce Services and Applications", International Journal of Mobile Communications, Vol. 3, No. 3, pp. 303-323.

ASP.NET Mobile Controls (2006), http://www.asp.net/mobile/.

cHTML (2006), http://www.w3.org/TR/1998/NOTE-compactHTML-19980209/.

Grech, S. (2004) "i-mode pricing", http://www.netlab.hut.fi/opetus/s38042/k03/topics/i-modepricing.pdf.

Gupta, A. and Srivastava, M. (2001), "Integrated Java Technology for End-to-End m-Commerce", http://developers.sun.com/techtopics/mobility/midp/articles/mcommerce/.

Ho, S.Y. and Kwok, S.H. (2003), "The Attraction of Personalized Service for Users in Mobile Commerce: An Empirical Study", ACM SIGecom Exchanges, Vol. 3, No. 4, pp. 10-18.

Ishii, K. (2004), "Internet Use via Mobile Phone in Japan", Telecommunications Policy, 28(1), pp. 43-58.

J2ME (Java 2 Platform Micro Edition) (2006), http://java.sun.com/j2me/.

Kumar, V., Parimi, S. and Agrawal, D.P. (2003), "WAP: Present and Future", IEEE Pervasive Computing, Vol. 2, No. 1, pp. 79-83.

Mahatanankoon, P., Wen H.J. Lim, B. (2006), "Evaluating the Technological Characteristics and Trust Affecting Mobile Device Usage", International Journal of Mobile Communications, Vol. 4, No. 6, pp. 662 - 681.

Microsoft Visual Studio (2006), http://msdn.microsoft.com/vstudio/.

Microsoft Windows Mobile (2006), http://www.microsoft.com/windowsmobile/.

NTT DoCoMo (2006), http://www.nttdocomo.com/.

Openwave WAP emulator (2006), http://developer.openwave.com/dvl/.

Palomäki, J. (2004), "Case WAP: Reasons for Failure", Proceedings of the Research Seminar on Telecommunications Business II, http://www.tml.hut.fi/Opinnot/T-109.551/2004/reports/wap.doc.

QUALCOMM BREW (2006), http://brew.qualcomm.com/brew/en/.

Read, K. and Maurer, F. (2003), "Developing Mobile Wireless Applications", IEEE Internet Computing, Vol. 7, No. 1, pp. 81 - 86.

Shim, J.P, Varshney, U., Dekleva, S. and G. Knoerzer (2006), "Mobile and Wireless Networks: Services, Evolution & Issues", International Journal on Mobile Communications, Vol. 4, No. 4, pp. 405-417.

Varshney, U. (2003), "Issues, Requirements and Support for Location-Intensive Mobile Commerce Applications", International Journal of Mobile Communications, Vol. 1, No. 3, pp.247–263.

Varshney, U., Malloy, A., Ahluwalia, P. and Jain, R. (2004), "Wireless in the Enterprise: Requirements, Solutions and Research Directions", International Journal of Mobile Communications, Vol. 2, No. 4, pp. 354–367.

WAP Forum (2006), http://www.wapforum.org/.

Zafeiri, K., Gavalas, D. and Balla, A. (2006), "Selling Culture: Implementation of e-Commerce and WAP-based Prototypes", International Review on Computers and Software, Vol. 1, No. 1, pp 68-76.