Proceedings of the 5th IASTED International Conference on Communication Systems and Networks (CSN'2006), C.E. Palau Salvador (Eds.), ACTA Press, ISBN: 0-88986-606-6, pp. 208-212, Palma de Mallorca, Spain, 28-30 August 2006.

A NOVEL METHOD FOR THE DEVELOPMENT OF PERSONALIZED MOBILE TOURIST APPLICATIONS

Michael Kenteris, Damianos Gavalas Daphne Economou,
Department of Cultural Technology and Communication, University of the Aegean
Mytilene, Lesvos, Greece
ctm04007@ct.aegean.gr, dgavalas@aegean.gr, d.economou@ct.aegean.gr

ABSTRACT
The use of mobile devices as electronic tourist guides represents a recent trend that brings the concept of 'mobile tourism' into the spotlight. While much of the underlying technology is already available, there are still open challenges with respect to design, usability, portability and implementation aspects. Most existing 'mobile tourism' solutions either represent of-the-shelf applications with rigidly defined content or involve portable devices with networking capabilities that browse tourist mobile portals with the requirement of constant airtime, i.e. continuous wireless network coverage. This paper presents a 'mobile tourism' research prototype which brings together the main assets of the two abovementioned approaches. Namely, it enables the creation of portable tourist applications with rich content that matches user preferences. The users may download these customizable applications either directly to their mobile device or first to a PC and then to a mobile terminal (through infrared or bluetooth). Thereafter, network coverage is not further required as the applications execute in standalone mode and may be updated when the user returns online. Our prototype has been developed on the top of Java 2 Micro Edition which offers an ideal platform for the development of full-fledged, interactive and portable applications tailored to resource-constrained mobile devices.

KEYWORDS
Tourist application, mobile devices, J2ME, XML

1. Introduction

The vision of nomadic users having seamless, worldwide access to a range of tourist services is expected to be a reality within only a few years from now. Hence, the concept of 'mobile tourism', wherein users access tourist content through mobile devices, has recently emerged [2].

Mobile devices present many unique characteristics that makes their use as electronic tourist guides particularly attractive: (a) ubiquity and convenience, (b) positioning: using technologies like GPS users may receive and access information and services specific to their location [20], (c) personalization: unlike PCs, handheld devices are typically operated by a single user, thereby enabling the provision of personalized services by wireless web portals [8]. However, several restrictions of mobile computing need to be carefully evaluated by tourist service providers: (a) restricted energy capacity, limited computing power, amount of memory and storage space, (b) small display size, limited color and font number support, small and hard to use keyboard, (c) limited bandwidth and high cost of wireless connections. Admittedly, although the capabilities of the mobile devices increase, the 'resource gap' between mobile and stationary devices will always be there.

Most existing commercial applications and research approaches in the field of mobile tourism basically fall within two main categories that involve:

- Tourist guides with pre-installed applications, namely rigidly defined content (in text, visual and auditory format) that cannot be customized according to user preferences (e.g. [3]).

Mobile devices used to access mobile web portals and browse tourist information of interest (e.g.

- [1], [4], [5], [18]).

The second approach implies the use of a mobile or wireless network to access Internet resources; wireless web access is typically enabled by WAP[1] [21] or i-mode[2] [17] compatible devices. Recently, a number of WAP and i-mode official and unofficial services have been deployed providing tourist, context-aware [9] and location-based services [20], for instance, the i-mode's i-area [10] service. The main disadvantage of i-mode/WAP-based services is their requirement for

[1] The Wireless Application Protocol (WAP) is a wireless web standard, backed by the entire telecommunication industry (through the WAP Forum). WAP specifies a protocol stack used to deliver data within the boundaries of a mobile/wireless network and a gateway that translates requests from the WAP protocol stack to the WWW stack and vice-versa [6].

[2] i-mode is a more recent, alternative to WAP, approach for the wireless Internet. In principle, i-mode is a service of wireless Internet (unlike WAP which is a protocol stack. It has been released by the Japanese NTT DoCoMo in 1999 [6].

constant connection (airtime) of the mobile device with a mobile network to offer access to web content. Thus, users are charged for the wireless connections (either pay-per-minute or pay-per-packet billing policy applies); most importantly though, whenever a user is out of coverage of the mobile network (i.e. 'has no signal') s/he cannot access any service.

This article presents the analysis and design of a city tourist guide system and also its implementation in a prototype system, the myMytileneCity guide. Our prototype includes a datatabase-enabled, tourist web site wherein tourists planning to visit the city of Mytilene (Greece) choose the content that interests them (lodging, sightseeing, entertainment, etc.); based on that chosen content, the system automatically generates a custom application which can operate on their mobile phone or PDA. On a second stage, the users may download their application either directly to their mobile device or first to a PC and then to a mobile terminal (through infrared or bluetooth). In contrast to i-mode/WAP-enabled applications that presuppose continuous connection of service users to the service providers network, the myMytileneCity guide does not pose such requirement. That is, following its installation, the application is fully functional with no extra charge, even in places where the connection to the mobile network is not feasible. Our prototype implementation is based on Java 2 Micro Edition (J2ME) which represents an ideal platform for developing powerful and portable applications operating in asynchronous mode.

The remainder of the paper is organized as follows: Section 2 provides background information on J2ME platform. Section 3 presents the design and implementation details of our 'mobile tourism' research prototype and Section 4 concludes the paper and draws directions to further work on this area of research.

2. J2ME (Java 2 Micro Edition)

Java 2 Micro Edition (J2ME) [13], released by Sun Microsystems, is a Java-based framework for developing applications executed on resource-constrained devices. J2ME has achieved a remarkable penetration and is currently supported by virtually all mobile devices. J2ME applications are called MIDlets; MIDlets are usually packaged in *.jar files, downloaded on-the-fly from a web server and executed as standalone applications with no requirement for constant connection to a wireless network.

J2ME inherits the main assets of Java language, i.e. the capacity to develop powerful applications, platform independence, etc. Hence, developers are not restricted by the limitations of an authoring tool's functionality and may implement full-fledged innovative applications that either execute standalone or communicate with their peers or service providers, taking advantage of the J2ME's strong wireless networking support. The communication of MIDlets with web servers is carried out over the Internet's HTTP 1.1 protocol.

J2ME platform presents many advantages that indicate its suitability for developing mobile tourist guide applications:

- It inherits the main 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 applications can practically download and parse content of any 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 rich graphics that offer enhanced user experience. Because graphics are typically generated locally, network bandwidth demand is reduced [7].
- J2ME enables disconnected access and synchronization. Java-based mobile applications can run even when their hosting device is disconnected or out of the network 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 [7].

On the other hand, several weaknesses o J2ME technology should be carefully considered. First, J2ME applications have increased requirements on devices' resources: storage, processing power and memory. Second, MIDlets programming is not straightforward as it requires Java development experience. Most importantly, the download of new J2ME applications (jar files) is costly, slow and consumes network resources. With respect to the latter disadvantage, it is evident that J2ME-based mobile tourist guide applications should -ideally- be downloaded only once; synchronization with the backend server should thereafter be considered only when the user wishes do update the selected tourist content.

3. A J2ME-based Mobile Tourist Guide Application: Design and Implementation

In this section, we present a mobile tourist guide application, the myMytileneCity guide. Our prototype implementation is entirely based on the powerful Java technology, on both the web and the client tier, in order to take advantage of its inherent platform-independence and suitability for web applications. Regarding the supported format of tourist content, we have chosen XML-family technologies to enable compatibility with web standards and interoperability (current trends promote XML-based languages for content development and data interchange on the Internet).

The design of our tourist guide application follows a two-steps approach: On the first step (illustrated in Figure 1),

the user interested in a particular tourist destination visits a web site including information about restaurants, accommodation, sightseeings, events, night life etc. The user appends information of interest to its 'web suitcase' (abstraction of personal account) which may be stored and retrieved upon a future visit. When the user 'checks out', the suitcase's content is transformed to XML format, using the powerful Java API for XML Processing (JAXP) [14]. Following that, the system automatically generates a jar file bundle that includes the MIDlet application (to be executed on the user's mobile phone) and the selected XML-based tourist content.

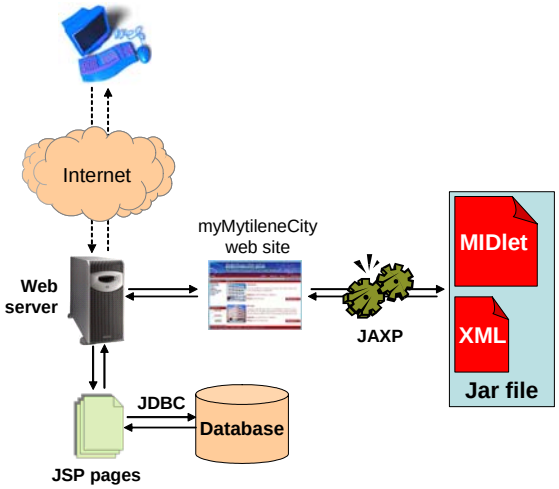

Figure 1. Dynamic generation of a J2ME application through remote interaction with myMytileneCity web site.

Our web site comprises dynamic pages enabled by JavaServer Pages (JSP) server-side programming technology [15] (in addition to using 'traditional' HTML/CSS). The tourist content is retrieved from a MySQL relational database, while the interface between the JSP pages and the database is provided by a JDBC (Java DataBase Connectivity) MySQL driver. In the future, we plan to develop a 'parallel' mobile portal that will enable access to (i-mode/WAP/J2ME-compatible) mobile devices to remotely browse tourist information.

On the second step, the user downloads the generated jar file (temporarily saved on the web server) to its mobile device. Herein, the user is provided two options (see Figure 2):

(a) Directly download the jar file to his/her mobile device (through end-to-end HTTP); depending on the jar file size this may be a time-consuming and costly solution as it engages the wireless channel for relatively long time.

(b) Download the jar file on two phases: first to a PC/laptop (though HTTP) and then to the mobile device (through bluetooth or infrared); although separated in two phases, this method is usually more cost-effective and fast as it takes advantage of the higher transfer rates of the wired Internet and the wireless bluetooth protocol.

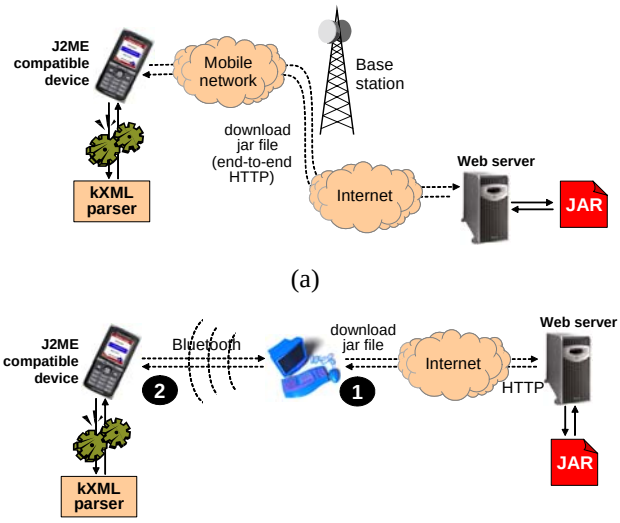

Figure 2. Scenarios for downloading the MIDlet application to the mobile device: (a) Direct download to the mobile device through a mobile network and the Internet; (b) First download to a PC/laptop and then to the mobile device (through bluetooth).

Upon completion of the jar file download to the J2ME-compatible device, the latter is installed and loaded by the local AMS module[3] (integrated within the J2ME platform). The MIDlet application is thereafter executed in standalone mode with no wireless connectivity requirement (the user later synchronizes to the backend system only to update the originally selected tourist content). On the client tier, we have designed a user-friendly MIDlet menu that allows easy browsing of selected content; the latter is included within the downloaded jar file (in XML format) and rendered for display by the 'lightweight' kXML parser [11].

The following figures present various parts of our prototype usage. Figure 3 shows screenshots of the web site used by visitors to select tourist content of interest. The web site has been designed so as to resemble e-commerce sites. It is noted that, in addition to the above-mentioned functionality, the myMytileneCity web site offers advanced search facilities for tourist content that matches specific criteria and allows users to share their opinions and experiences with respect to destinations they have already visited, make recommendations to other users, etc.

[3] Application Management Software (AMS) controls the management (start, pause, termination) of MIDlets execution as well as their installation / un-installation; AMS is typically provided by the device's manufacturer.

(a)

(b)

Figure 3. Screenshots from the myMytileneCity web site: (a) the main page; (b) contents of user's account (includes two lodging preferences).

Figure 4 demonstrates an excerpt of generated XML code, describing the users' selected content (this code corresponds to the user preferences shown in Figure 3b). Finally, Figure 5 illustrates representative screens of a mobile device emulator (main menu, list of lodging preferences, detailed description of a selected hotel). It is noted that the application's menus are created dynamically. Specifically, the MIDlet first extracts the XML file from the jar file. Then it parses the XML code to dynamically build the main menu; for instance, the option 'Accommodation' will not be displayed in the menu, unless the user has indeed selected at least one hotel of interest. The second-level lists of content items (e.g. list of selected hotels) are populated at startup to avoid the unnecessary processing burden and delay associated with lists' items parsing and display each time that the user clicks on a specific menu item.

The Sun Java Wireless Tookit [19] has been used for our prototype development (which also integrates a J2ME emulator); the MIDlet's menus templates have been designed using J2ME Polish tool[4] [12].

[4] J2ME Polish comprises a collection of tools for developing J2ME applications: build tools for creating application bundles for multiple devices and multiple locales; a device database that helps to adjust applications to different handsets; tools for designing GUIs using simple CSS text-files; utility classes.

```
<?xml version="1.0" encoding="UTF-8"?>
<city name="Mytilene">
    <content id="2" type="Accommodation">
        <sub id="20" name="Hotels">
        <cname>Hotel Erato</cname>
        <cdescription>Fine view of the Port. Easy access
                …..</cdescription>
        <caddress>Pavlos Bostani 2, Mytilene</caddress>
        <cphone>22510-41160</cphone>
        <cimage>erato.jpg</cimage>
        </sub>
    </content >
    <content id="2" type="Accommodation">
        <sub id="20" name="Hotels">
        <cname>Blue Sea</cname>
        <cdescription>Situated on the left side of the Port
                …..</cdescription>
        <caddress>Kountouriotou 22, Mytilene</caddress>
        <cphone>22510-23994-995</cphone>
        <cimage>bluesea.jpg</cimage>
        </sub>
    </content >
</city>
```

Figure 4. An excerpt of XML code describing the users' accommodation preferences.

(a) (b)

(c) (d)

Figure 5. Screens of a mobile phone emulator executing the myMytileneCity guide application.

4. Conclusions & Future work

In this article we presented a J2ME-based 'mobile tourism' research prototype. Our design objective has

been twofold: First, to enable the automated creation of portable, personalized tourist applications with rich and customized content. Second, to minimize the wireless connectivity requirement for users of the mobile tourist guide application: following the application's download and installation, network coverage is not further required as the applications execute on standalone mode; users should return online only to update their chosen tourist content.

Our prototype relies in Java and XML technologies for enhanced efficiency, portability, interoperability and compatibility with web content standards. Implementation experiences revealed that J2ME offers an ideal platform for the development of powerful, interactive and portable applications tailored to small devices with restricted processing, memory and storage resources.

Our prototype application will form the basis for our field trials. These will involve measurements of the delay and network overhead imposed by mobile device–server interactions, and also evaluation of the storage and memory requirements of our application. We also plan to perform usability tests with units being tested by members of our laboratory and the general public.

Regarding future work, we plan to 'look' at the following directions:

- Use of the optional MIDP 'Location API' [16] which has been very recently released by Sun Microsystems to extend the tourist guide application so as to provide orientation, navigation and other location-based services (e.g. notifying the user when he/she walks next to a selected landmark).
- Development of a user-friendly authoring tool for automatic generation of J2ME code (e.g. design of user interfaces) based on user-defined criteria in order to remove the requirement for Java programming skills to develop J2ME applications.
- Design and implementation of algorithmic solutions for suggesting near-optimal, daily tourist itineraries for tourists interested in visiting a specific group of sites (e.g. museums, archeological sites, parks, zoos, etc.) These algorithms will take into account several parameters such as the user profile, the period of stay, the whether conditions, etc.

References

[1] H. Anegg, H. Kunczier, E. Michmayr, G. Pospischil, and M. Umlauft (2002), "LoL@: Designing a Location Based UMTS Application", Elecktrotechnik und Informationstechnik, 191(2), pp. 48-51.

[2] B. Brown and M. Chalmers (2003), "Tourism and Mobile Technology", Proceedings of the European Conference on Computer Supported Collaborative Work (CSCW'2003), Kluwer Academic Press, pp 335-355.

[3] B. Brown and M. Perry (2001), "Of Maps and Guidebooks: Designing Geographical Technologies", ACM SIGGROUP Bulletin, 22(3), pp. 28-32.

[4] K. Cheverst, N. Davies, K. Mitchell, A. Friday and C. Efstratiou (2000), "Developing a Context-Aware Electronic Tourist Guide: Some issues and Experiences", Proceedings of the 2000 Conference on Human Factors in Computing Systems (CHI'00), pp. 17-24.

[5] Dublin Tourism on the Wireless Web, http://www.visitdublin.com/wireless/.

[6] D. Gavalas, D. Economou and M. Kenteris, "Status and Trends of Wireless Web Technologies", Proceedings of the 5th IASTED International Conference on Communication Systems and Networks (CSN'2006).

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

[8] S.Y Ho and S.H. Kwok (2003), "The Attraction of Personalized Service for Users in Mobile Commerce: An Empirical Study", ACM SIGecom Exchanges, 3(4), pp. 10-18.

[9] I. Hochstatter, A. Küpper, M. Schiffers, L. Köthner (2003), "Context Provisioning in Cellural Networks", Proceedings of the 8th IEEE International Workshop on Mobile Multimedia Communications.

[10] i-area: Location Based Services, http://www.nttdocomo.com/corebiz/services/imode/iarea.html.

[11] kXML parser, http://kxmlrpc.objectweb.org/.

[12] J2ME Polish, http://www.j2mepolish.org/.

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

[14] Java API for XML Processing (JAXP), http://java.sun.com/webservices/jaxp/index.jsp.

[15] JavaServer Pages Technology, http://java.sun.com/products/jsp/.

[16] JSR 179: Location API for J2ME, http://www.jcp.org/en/jsr/detail?id=179.

[17] NTT DoCoMo, http://www.nttdocomo.com/.

[18] R. Oppermann and M. Specht (2000), "A Context-Sensitve Nomadic Exhibition Guide", Proceedings of the 2nd Symposium on Handheld and Ubiquitous Computing (HUC2K), pp. 127-149.

[19] Sun Java Wireless Tookit, http://java.sun.com/products/sjwtoolkit/.

[20] U. Varshney (2003), "Issues, Requirements and Support for Location-Intensive Mobile Commerce Applications", International Journal of Mobile Communications, 1(3), pp.247–263.

[21] Wireless Application Protocol (WAP) Forum, http://www.wapforum.org/.