

An innovative mobile electronic tourist guide application

Michael Kenteris · Damianos Gavalas ·
Daphne Economou

Received: 28 February 2007 / Accepted: 26 June 2007 / Published online: 25 September 2007
© Springer-Verlag London Limited 2007

Abstract “Mobile tourism” represents a relatively new trend in the field of tourism and involves the use of mobile devices as electronic tourist guides. While much of the underlying technology is already available, there are still open challenges with respect to design, usability, portability, functionality 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 access tourist content with the requirement of constant airtime, i.e., continuous wireless network coverage. This paper presents the design and implementation issues of a “mobile tourism” research prototype, which brings together the main assets of the two aforementioned approaches. Namely, it enables the creation of portable tourist applications with rich content that matches user preferences. The users may download these personalized applications (optimized for their specific device’s model) 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. The dynamically created tourist

applications also incorporate a “push model”, wherein new tourist content is forwarded to the mobile terminal with minimal user intervention as soon as it is added or updated by the administrator. Our prototype has been developed on the top of Java 2 Micro Edition (J2ME) which offers an ideal platform for the development of full-fledged, interactive and portable applications tailored for resource-constrained mobile devices. The paper presents our development experiences with J2ME and highlights its main advantages and shortcomings in relation to the implementation of such kind of applications. Finally, an empirical evaluation of user experience with the mobile application prototype is presented.

Keywords Tourist application · Mobile devices · Electronic tourist guide · J2ME · XML · Web · Push model · Usability tests

1 Introduction

The vision of nomadic users having seamless, worldwide access to a range of tourist services is expected to become a reality within only a few years from now. Hence, the concept of “mobile tourism” has recently emerged wherein users access tourist content through mobile devices [6].

Mobile devices present many unique characteristics that make their use as electronic tourist guides particularly attractive, such as ubiquity and convenience; positioning: by employing technologies like GPS, users may receive and access information and services specific to their location [45]; personalization: unlike PCs, handheld devices are typically operated by a single user, thereby enabling the provision of personalized services by wireless web portals [20].

M. Kenteris (✉) · D. Gavalas · D. Economou
Cultural Heritage Management Laboratory (CHMLab),
Department of Cultural Technology and Communication,
University of the Aegean, Harilaou Trikoupi & Faonos St,
81100 Mytilene, Lesvos, Greece
e-mail: mkederis@otenet.gr

D. Gavalas
e-mail: dgavalas@aegean.gr

D. Economou
e-mail: d.economou@ct.aegean.gr

However, several restrictions of mobile computing need to be carefully evaluated by tourist service providers: restricted energy capacity, limited computing power, amount of memory and storage space; small display size, limited color and font number support, small and hard to use keyboard; 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 three main categories that involve:

- tourist or museum 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., [7, 37]);
- mobile devices used to access mobile web portals and browse tourist information of interest (e.g., [13]);
- mobile electronic guides devices that use either wireless [2, 10, 39] or mobile network connections [1, 35] to access context-aware services.

The second approach implies the use of a mobile or wireless network to access Internet resources; wireless web access is typically enabled by WAP¹ [46] or i-mode² [38] compatible devices.³ The main disadvantage of i-mode/WAP-based services is their requirement for 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). Similarly, the third approach assumes some type of network connection and tracking systems (e.g., GPS or infrared beacons) to provide location-based services. Hence, whenever a user is out of coverage of the mobile network (i.e., “has no signal”) he/she cannot access any service [16].

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 database-enabled tourist web site wherein, on a first stage, tourists planning to visit the city of Mytilene (Lesvos Island, 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 to the service provider’s 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. Moreover, in contrast to existing applications, the myMytileneCity guide incorporates a “push model”. Namely, it enables mobile users to be notified and update their personalized application as soon as new content or services, which the user has registered to, is added or updated by the web site’s administrator. 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: Sect. 2 reviews related works and Sect. 3 discusses issues related to the design of mobile devices-enabled applications for tourism. Section 4 provides background information on the J2ME platform and Sect. 5 presents the implementation details of our “mobile tourist guide” research prototype. Section 6 summarizes our experiences gained from the prototype development. Section 7 discusses the empirical evaluation of user experience with the mobile application prototype and Sect. 8 concludes the paper and draws directions to further work. Finally, the Appendix includes UML sequence diagrams that explain our proposed application’s design and functionality.

2 Related work

Tourism is a worldwide industry which involves the propagation of large amounts of information. As in most industries, Information Technology has penetrated the field of tourism for the manipulation of such information. The convergence of IT and communications technologies and the rapid evolution of the Internet has been one of the most influential factors in tourism that change travelers’ behavior.

At the same time, the rapid growth of mobile devices (mobile phones, palmtops, PDAs, etc.) user base raises a

¹ The Wireless Application Protocol (WAP) is a wireless web standard (released in 1997), backed by the entire telecommunication industry (through the WAP Forum). Essentially, WAP specifies a protocol stack used to submit and 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, so they can be submitted to web servers.

² i-mode is a more recent, alternative to WAP, approach for the wireless Internet, which addresses the main weaknesses arisen with the practical implementation of WAP. In principle, i-mode is a service of wireless Internet (unlike WAP which is a protocol stack), with a large subscriber base in Japan. It has been released by the Japanese NTT DoCoMo in 1999.

³ Recently, a number of WAP and i-mode official and unofficial services have emerged providing tourist, context-aware [21] and location-based services [45], for instance, the i-mode’s i-area [23] service.

demand for using such devices for accessing Internet resources. The use of mobile devices as means for supporting tourist and cultural activities is not an entirely new idea. In fact, mobile audio guides are being used in museum environments for years [15], while the use of personal digital assistants (PDAs) to enhance the museum visit experience has also been recently proposed [37].

Wireless access through mobile devices adds to the Internet connection the element of “portability”, i.e., connection with no time or geographical limitations, by devices with high penetration to the public [16]; tourists are amongst this technology-oriented public. Hence, a growing body of commercial and research initiatives that incorporate electronic tourist guide functionality into mobile devices have been reported [34]. However these technologies in general have had a limited success; this is due to the lack of an in-depth study of the special characteristics of tourism, which can draw implications for the design of mobile tourist applications [6]. In the remainder of this section, several commercial and research projects with respect to mobile tourist guides are being reviewed. Client–server interactions and location-based services are the key features shared among all the reviewed projects.

The Guide project was about the city of Lancaster, UK [10]. It used a series of wireless access points to locate a user and derive information for a point of interest via a browser-based interface. The TellMaris is a Nokia Research center prototype [35]. It was one of the first mobile systems to use 3D graphics in combination with 2D Maps and has been developed for the city of Tonsberg in Norway. The LLocal Location assistant (LoL@) project was a mobile tourist guide designed for the city of Vienna using the UMTS Mobile Telecommunication system [1]. This system was one of the first projects based on next generation mobile phone technology. The REAL project was a hybrid pedestrian navigation system which helps the user to find information by generating a graphical route description [2]. Its location tracking system is based on GPS/compass signals outdoors and infrared beacons indoors.

All reviewed projects assume some type of network connection as a basis of location-based services provision: either a wireless [2, 10] or a mobile network infrastructure [1, 35]. Also, most of these projects (with the exception of LoL@) have not been designed for mobile phone usage; rather they were programmed for specific PDA platforms. Furthermore, all the abovementioned projects are based on the assumption that the user initiates the search, browsing and download of tourist information (“pull” model). That is, under no circumstances the delivery of information with no user intervention is enabled (“push” model).

In conclusion, the usage of mobile phones as offline mobile tourist guides requires further research. The

myMytileneCity Guide (developed at our laboratory and presented herein) addresses the shortcomings of existing systems. Its main design and implementation objectives are the following:

- the system should allow the generation of tourist guide applications over the Internet, which can later be executed on non-networked mobile terminals
- the personalized tourist applications should execute on any mobile device (mobile phones, smart phones, PDAs, palmtops, etc.) independently of their hardware platform or operating system
- the applications should enable the dynamic delivery of tourist content with minimal user intervention, given that the user has expressed interest on the corresponding content type.

The following section discusses issues related to the design of PDA and mobile phone applications, with emphasis on their role in tourism.

3 Requirements for mobile devices-enabled applications for tourism

The design of successful mobile devices’ applications involves factors related to the technical characteristics of the devices, the use of the applications [14] and the business model [22].

These factors charge the application designers with new challenges, such as: design for mobility; design for a wide audience with various levels of competency in the use of new technologies, that do not necessarily have a history of experience with similar applications to draw on in learning a new one; design for limited input (diminutive and impractical keyboard)/output facilities (small screen size, limited color and font size support); and design for user multitasking at levels unfamiliar to most desktop users [5, 47]. Unlike PDAs, most of the mobile phone applications are downloaded over-the-air (OTA), which implies lack of user manuals, long installation time and possibly high download cost. Mobile phone devices in general have less processing capabilities than PDAs and are not yet fully multimedia enabled (which are important issues for developing effective and attractive applications).

In terms of the interface design and usage, mobile devices’ applications should pursue criteria similar to web sites development [12]. The design of an aesthetically pleasing interface is important, however, the success of the system is based on accessing information in an intuitive and easy way [40]. Canadian Heritage Information Network (CHIN) adds some practical guidelines for the graphic design of mobile cultural guides’ interface [9] which stretch that: each screen node of the application

should fit the size of the mobile device screen; the navigation should be structured hierarchically; and backtrack and easy access to the home page should be supported.

Kray and Baus in 2001 completed a survey of available mobile tour guide applications [34]. Many issues were retrieved related to mobile phone applications design. According to our criteria the most important in order to satisfy user and application need is: the provision of a tourist guide which can be executed on non-networked mobile terminals; the personalization of the tourist application to be execute on any mobile device and the support of dynamic delivery of tourist content given that the user has expressed interest on the corresponding content type. These requirements have been considered in the design of the electronic tourist guide which is described in detail in Sect. 5.

4 Java 2 Micro Edition (J2ME)

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 [25] by Sun Microsystems, in 1999. J2ME is a framework for developing applications executed on resource-constrained devices. However, it also offers strong wireless networking support and enables programming applications able of accessing a broad range of web content formats; hence, J2ME may also be considered as an (alternative to WAP and i-mode) wireless Internet technology.

Java 2 Micro Edition follows a modular design which aims at simplifying the support for a broad range of small

devices. Thus, it introduced the concepts of *configuration* and *profile* (see Fig. 1). J2ME configurations define the minimum features of a Java Virtual Machine and a minimum set of libraries for a “horizontal” family of devices, i.e., devices with similar processing and memory limitations, user interface requirements and connection capabilities. The configuration currently supported by all J2ME-enabled mobile devices is the connected limited device configuration (CLDC).

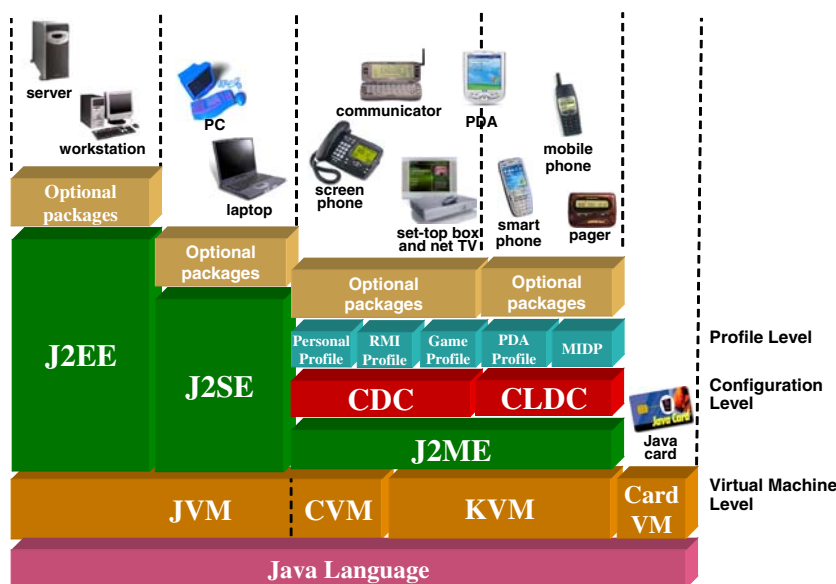
Profiles are implemented on the top of configurations (a configuration is the base for one or more profiles). Typically, a profile includes libraries specialized in the unique characteristics of a particular class of devices. Java applications are built on the top of a particular configuration/profile pair and based on their specified class libraries. Currently, the only available profile specified on the top of CLDC is mobile information device profile (MIDP). The Java applications developed over MIDP profile (and CLDC configuration) are called MIDlets, usually packaged in *.jar files.

MIDlets are typically downloaded on the fly from a web server 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.

J2ME platform presents many advantages that signify 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

Fig. 1 Java technologies and their respective target devices (adapted from [48])



hardware or the operating system), etc. In addition, apart from simply browsing content (as in the case of WAP/i-mode), 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) [48].

- Java 2 Micro Edition 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, since graphics are typically generated locally, network bandwidth demand is reduced [19].
- Java 2 Micro Edition 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 [19].

On the other hand, several weaknesses of 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 to update the selected tourist content [16].

5 A J2ME-based mobile tourist guide application: prototype implementation

In this section, a prototype mobile tourist guide application is presented, the myMytileneCity guide. This 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, XML-family technologies have been chosen to enable compatibility with web standards

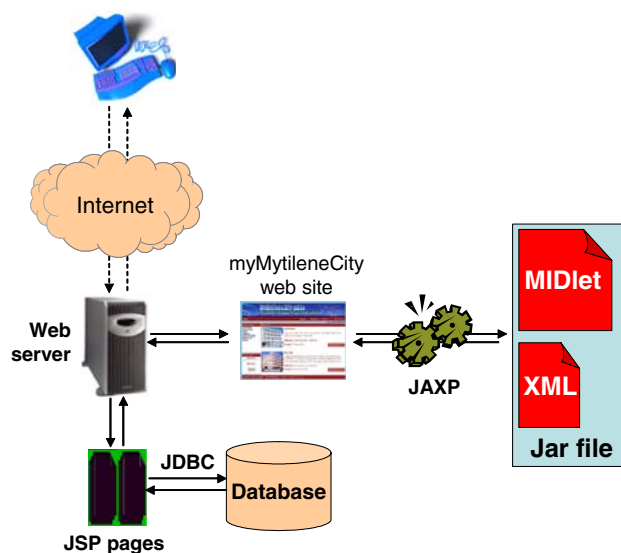


Fig. 2 Dynamic generation of a J2ME application through remote interaction with myMytileneCity web site

and interoperability (current trends promote XML-based languages for content development and data interchange on the Internet).

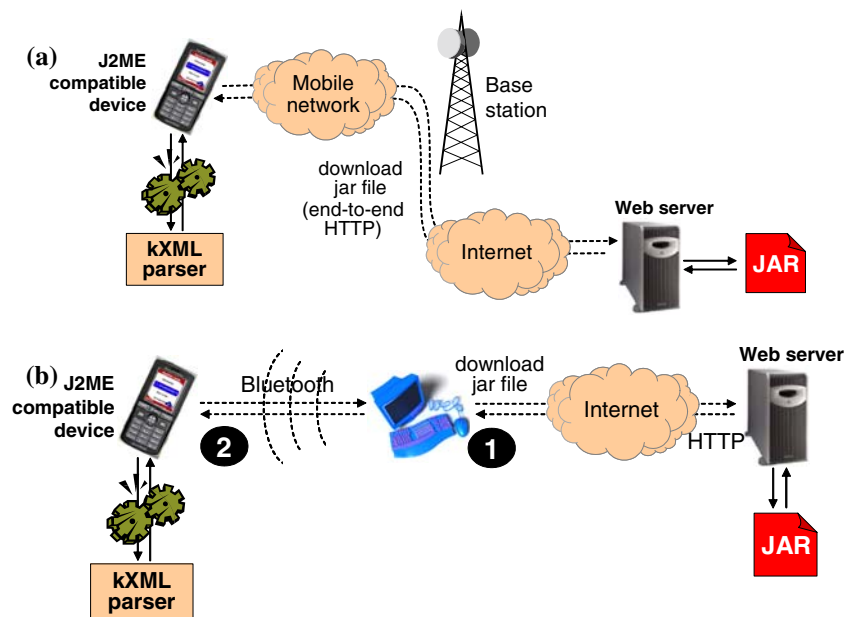
Some initial ideas and experiences related to the implementation of the application prototype presented herein also appear in [31]. Also, interested readers may find in the Appendix several UML sequence diagrams that explain our proposed application's design and functionality.

The design of the myMytileneCity guide follows a two-steps approach: (a) the creation of a J2ME-enabled tourist application; (b) the download and installation of the tourist application. These are portrayed in the following two subsections.

5.1 Creation of the J2ME-enabled tourist application

In the first step (illustrated in Fig. 2), the user interested in a particular tourist destination visits a web site including information about restaurants, accommodation, sightseeing, events, night life, etc. The user loads 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) [26] and saved on the file server to be incorporated in the custom mobile application. Following that, the system automatically generates a jar file bundle that includes the MIDlet application (to be executed on the user's mobile phone), graphics and the selected

Fig. 3 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)



XML-based tourist content,⁴ the application is built using the ANT build [18] console command (an open-source Java-based build utility). It should be emphasized that the generated jar file is optimized for the specific user mobile device's profile (see Fig. 4c).

The web site comprises dynamic pages enabled by JavaServer Pages (JSP) server-side programming technology [27] (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 Java DataBase Connectivity (JDBC) MySQL driver. The personalized tourist application is temporarily stored in binary format in the MySQL database upon user “check out”⁵ (when checking out, the user is asked to fill in a form with his/her credentials and the jar file is stored in the corresponding row of the USERS_SUITCASE database table).

5.2 Download and installation of the tourist application

In the second step, the user downloads the generated jar file (temporarily saved on the web server) to his/her mobile device. Herein, the user is given two options (see Fig. 3):

- directly download the jar file to his/her mobile device OTA (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
- 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.⁶

In both cases, the user first needs to sign-in in order to be given access and retrieve the jar file from the MySQL database; following that, the file is automatically removed to free server storage resources.

Upon completion of the jar file download to the J2ME-compatible device, the latter is installed and loaded by the local AMS module⁷ (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, a user-friendly MIDlet menu that allows easy browsing of selected content has been designed; the latter is included

⁴ It is noted that only a portion of the web page content is included into the XML file. In particular, each content item is accompanied by a full and a short description. The former appears on the corresponding JSP page while the latter is retrieved from the backend database and appended to the XML file in the case that the user selects this item.

⁵ In our implementation, the jar file is maintained for a period of 48 h. If the file is not downloaded within this period, it is automatically removed.

⁶ Tests undertaken by members of our laboratory have shown that the majority of mobile phone users are reluctant in trying to upload a file to a mobile phone. In the future, we plan to create a stand alone application which via a bluetooth communication port will seamlessly push the jar file to the users' mobile phones.

⁷ Application management software (AMS) controls the management (start, pause, termination) of MIDlets execution as well as their installation/un-installation; AMS runs on the same device as MIDlets and is typically provided by the device's manufacturer.

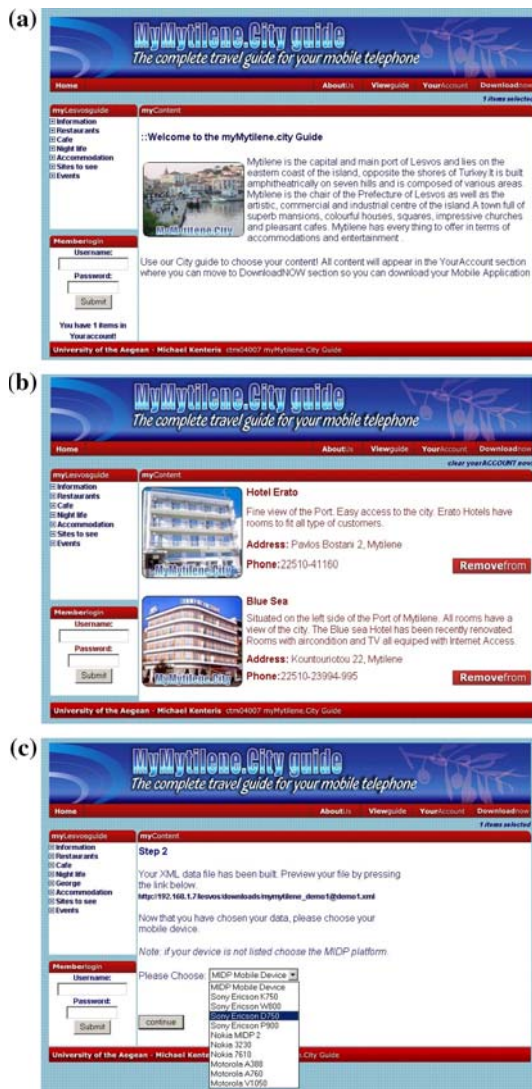
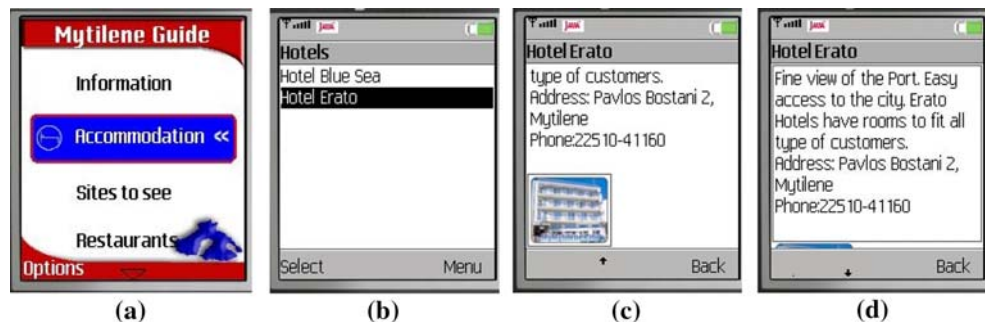


Fig. 4 Screenshots from the myMytileneCity web site: **a** the main page; **b** contents of user’s account (includes two lodging preferences); **c** selection of mobile device’s profile when ‘checking out’

within the downloaded jar file (in XML format) and rendered for display by the “lightweight”, open-source kXML parser [36].

Fig. 6 Screens of a mobile phone emulator executing the myMytileneCity guide application



```
<?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>bluese.jpg</cimage>
    </sub>
  </content >
</city>
```

Fig. 5 An excerpt of XML code describing the users’ accommodation preferences

The following figures present various parts of the prototype usage. Figure 4 shows screenshots of the web site used by visitors to select tourist content of interest and define their mobile device’s profile. 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 (through a specialized forum) to share their opinions and experiences with respect to destinations they have already visited, make recommendations to other users, etc. Post-visit reports and voting for best-to-see attractions features are also available, as suggested in [6].

Figure 5 demonstrates an excerpt of generated XML code, describing the users’ selected content (this code corresponds to the user preferences shown in Fig. 4b). Finally, Fig. 6 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. It is noted that a 2D digital map of Mytilene city with two zoom levels is accessible from any screen of the mobile application.

The Sun Java Wireless Toolkit [44] integrated development environment (IDE) 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 [24].⁸

5.3 “Pushing” tourist information to the end user

The tourist application also incorporates a “push” content service. This service involves automatic user notification and dynamic update of the applications with content inserted into the database after the user has pre-selected his initial content preferences. The application update is completed with minimal user intervention.

To illustrate, let us assume that a user is particularly interested in a specific content type, e.g., music festivals. While surfing the myMytileneCity web site, it is likely that no content exists for this content category. The user then subscribes (registers) to the “push content” service. In particular, the user fills in a form the content categories he/she interested in and also the exact period of visit (arrival and departure dates).

Upon the registration's completion, the server forks a dedicated thread (monitoring agent, MA) that periodically checks the database for new content. The MA is stationary and its lifetime depends on the ‘interest period’ defined by the user (i.e., the departure date from the tourist destination). On the event of having new information added by the web site administrator matching the subscribed content type (e.g., the arrangements for a music festival have just been finalized), the MA creates XML code with the description of the added content item(s), stores the XML file in the database and notifies the mobile subscriber through an SMS message. Although the added content's description is included in the SMS, the user is given the option to update his/her application with the new content items. In the latter case, the XML file may be downloaded either directly or through a PC (as discussed in Sect. 5.2); of course, the application's Java class files and the graphics

do not need to be downloaded again, hence, the cost of update is minimal. The XML file is removed from the server's file system when downloaded or after a specific timeout period elapses. The SMS messaging has been implemented using the Simplewire's Java SMS SDK [43].⁹ The steps involved in the process of pushing tourist content are illustrated in Fig. 7.

5.4 Application overhead and performance evaluation

To properly evaluate the performance and overhead that our tourist application prototype places upon mobile devices' resources, we have measured its storage and memory footprint as well as the application's download delays.

As shown in Fig. 8, the tourist application is built on the top of CDLC/MIDP and comprises the application logic (MIDlet class), utility classes, graphics and the content (XML file). The application evaluation parameters appear in Table 1.

The final application size is approximately 240 KB, including the kXML parser libraries and the jar file (assuming an XML file with ten content items and the accompanying graphics). However, the graphics files occupy the largest part of the application (~150 KB); the graphics overhead may probably be reduced through more careful image processing, without sacrificing their display quality. In any case, the overall size is considered as an acceptable dimension for current J2ME-enabled devices [4].

It is noted that post compilation, the personalized tourist application is -transparently to the user- compacted using the freely available obfuscator ProGuard [41].¹⁰ The obfuscation process keeps the application functionally equivalent. However, more compact jar files means smaller storage requirements, faster transfer of applications across networks, faster loading, and smaller memory footprints.

⁸ 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.

⁹ The Java SMS SDK is a 100% pure Java implementation that provides easy, high-level control of the Simplewire wireless text-messaging platform. The Java SMS SDK was designed to be as developer-friendly as possible by hiding the intricacies of the XML format required to communicate with the Simplewire Wireless Message Protocol (WMP) servers. The Java SMS SDK makes it possible to send an SMS message off with as little as two lines of code.

¹⁰ ProGuard is a free Java class file shrinker, optimizer, and obfuscator. It can detect and remove unused classes, fields, methods, and attributes. It can then optimize bytecode and remove unused instructions. Finally, it can rename the remaining classes, fields, and methods using short meaningless names. The resulting jars are smaller and harder to reverse-engineer.

Fig. 7 Steps undertaken in the process of pushing tourist content: **a** looking for tourist content, currently not available; **b** expressing interest for specific content items; **c** receipt of an SMS notification, **d** logging in the myMytileneCity system to download the updated content

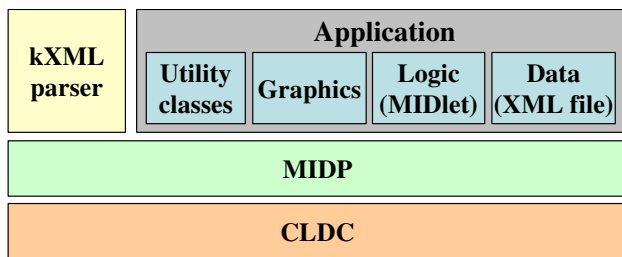
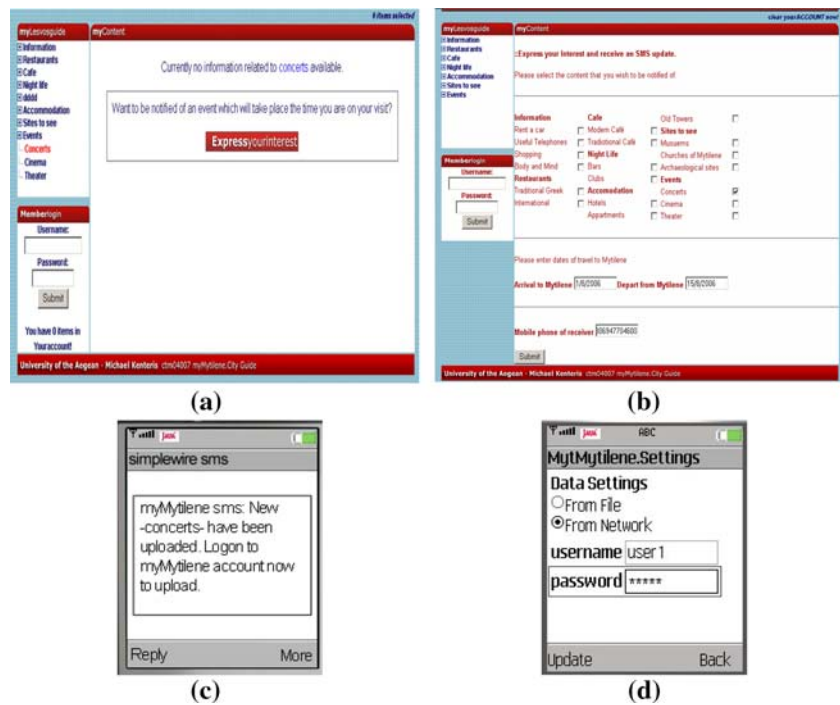


Fig. 8 Framework and application modules

6 Experiences gained from prototype development and evaluation of J2ME platform

The experience gained throughout the development process of the tourist application prototype has revealed several interesting aspects of J2ME platform. In particular, J2ME seems to offer a number of advantages:

- low application development cost (practically free)
- lightweight storage and memory footprint for both the runtime environment and the applications
- potential for developing innovative and portable applications powered by Java programming language, which can handle any content type and are not restricted by the functionality of an authoring tool
- very large deployment base (virtually all modern mobile devices are shipped with support for CLDC/MIDP)

Table 1 Bytecode sizes and memory footprint of application modules

Parameter	Value
Bytecode size of CLDC/MIDP	~0.8 MB ^a
Memory footprint of CLDC/MIDP	~230 KB
Bytecode size of kXML parser	22 KB
Memory footprint of kXML parser	~100 KB [33]
Application size (bytecodes, graphics & XML file with 10 content items)	220 KB
Application memory footprint	20–30 KB
Download time (over-the-air)	62 s ^b
Download time (through bluetooth)	6 s
Download cost (over-the-air)	€ 1.32 for the application, less than € 0.05 for an XML file with ten content items ^c
Download cost (through a PC and then bluetooth)	Practically free

^a This is the footprint of a typical MIDP 2.0 implementation including a CLDC implementation and the Kilo Virtual Machine (KVM)

^b All download times that appear in Table 1 represent the average times measured amongst ten downloads

^c Vodafone Hellas (Greece) currently charges 0.006 €/KByte through a GPRS phone. If the user chooses to download the whole jar file (application logic and content) over-the-air, he/she will be charged for ~220 KB (~€ 1.32). If the user returns to the web to update the application and adds ten new content items (average XML file size 7.5 KB) the download will cost less than € 0.05)

- capability for dynamic customization and over-the-air update of existing applications content and functionality
- current billing policies of mobile network operators allow the cost-effective download even of full applications; furthermore, the transmission rates of GPRS and 3G networks significantly reduce download times [42]
- support for disconnected operation, i.e., ability to run applications in standalone mode even when the mobile terminal is out of any network's coverage area
- large community of developers, which assists the exchange of development experiences (e.g., through developer forums)
- wide availability of add-on application libraries, which accelerate the implementation of custom services (e.g., parsers, APIs for implementing location-based services and applications that involve pushing content to mobile terminals, etc.).

Yet, several inherent weaknesses of J2ME have also been brought into the spotlight:

- although experienced in Java programming, the application's development team met difficulties in familiarizing with the main features and particularities of J2ME platform, which seriously affected their learning curve and decelerated the application development
- several development phases, which are typically straightforward when using an authoring tool (e.g., UI's layout design), required serious programming effort
- the inclusion of special features within J2ME applications require the usage of specialized (optional) APIs, which certainly increase the overall applications' overhead; such examples are the Location API [28] (allows to build applications that adapt according to the user's physical location), the Mobile 3D Graphics API [29] (use of lightweight 3D graphics), etc.

7 Usability considerations, usability evaluation and testing of the mobile application prototype

The formulation of usability guidelines as well as the usability testing of mobile applications represent emerging research areas that face a variety of challenges due to unique features of mobile devices, limited bandwidth, unreliability of wireless networks, as well as the changing context. Traditional guidelines and methods used in usability testing of desktop applications are not directly applicable to a mobile environment [49]. The following two subsections present the usability attributes taken into consideration in the design of

the mobile application and the results of the usability tests performed on myMytileneCity guide with the assistance of real users.

7.1 Design and usability qualities taken into consideration

myMytileneCity application has been designed taking into account several usability guidelines collected from relevant studies associated to three main characteristics of mobile applications small screens, limited input and mobility:

- the application is designed for mobility by providing an intuitive interface with simple dialogues and short and concise information so that interaction with the application requires minimal effort and does not distract the user's attention from other activities (walking, talking, etc.)
- the interface is appealing to a wide range of users with various skills and expertise
- the application's presentation follows a hierarchical multi-level structure that helps users to easily search for/browse/understand specific information of their interest. This design approach has been dictated by usability studies which have shown that a combination of summary and keywords of each document at first is more efficient for users to locate relevant information from a list of documents than displaying entire documents directly [8].
- menus have been designed in order to help the user to easily reach the desired information [11]. Menus and buttons are clearly and consistently labeled to help the user navigation, learnability and memorability. To minimize cognitive load, long lists of choices have been avoided and backtrack and easy access to earlier pages/home page is supported [3, 9]. Finally, the menus' structure facilitates users to finish tasks with minimum interaction with the mobile device (e.g., scrolling and button clicks).
- content page information is typically fitted on one screen to avoid scrolling.

In addition to developing an interface which provides easy interaction and access to information, factors that can determine the success or the failure of an application, emphasis has been put in developing an aesthetically pleasing interface.

7.2 Usability evaluation and testing of the mobile application prototype

Usability tests have been performed both in experimental environment and with field studies. Experimental testing is

more suitable for standalone mobile applications, i.e., those without network connectivity requirement [49]. In contrast, field studies allow the use of mobile applications in a realistic environment and takes dynamic mobile context into consideration, which is difficult to simulate in laboratory experiments [32]. During experimental studies, participants have been asked to accomplish specific tasks using the mobile application (i.e., it is easy to measure usability attributes and interpret results, while it makes it possible to use video or audio recording to capture participants reaction, including emotions) [8].

Twenty participants in total participated in the tests; 12 were males and 8 were female. The ages of the participants varied from 20 to 53 years, average age being 33.5 years. All had visited the city of Mytilene before, but only a few had seen or used a printed tourist guide or a map of Mytilene. However, the participants were quite experienced with printed guides and maps in general and 13 of them claimed they used tourist guides often. The participants were all experienced mobile phone users and nine of them had used in the past standalone mobile phone applications (e.g., calendar, address books, games, etc.), yet, none had previous experience with electronic tourist guides usage.

For the field studies the participants' personal mobile phones have been utilized as test tools (after downloading and installing the myMytileneCity guide application on them). While for the experimental studies a phone emulator have been used. Each usability test session consisted of three parts, an introduction to the scope of the session and the application, the main testing task continued by an interview.

During the introduction, the project and the application were introduced to the participant. Some demographic information about them have been collected like, age, sex, education, as well as information about their prior knowledge regarding use of mobile phones, conventional and electronic tourist guides and maps. Then the test procedure was briefly introduced to the participant.

The test part included four tasks (repeated twice), assigned to all participants:

1. select content items from the myMytileneCity web site and then download the mobile application directly to the mobile device through a mobile network and the Internet
2. repeat the first task, but trying to first download the application to a PC and then to the mobile device (through bluetooth)
3. access a couple of specific content pages of the mobile application
4. use the application's 2D map to find the way from a place to another (see Fig. 9). It is noted that no GPS

was available to highlight the participant's current position on the map.

During usability tests, the evaluators have recorded several quantitative usability attributes measurements for each individual participant. Out of the generic usability attributes identified in [17] and [49], we have measured those that fit in the nature of our electronic guide application:

- *effectiveness*: the percentage of tasks completed
- *efficiency*: time needed to solve tasks in comparison to a pre-defined "task completion time goal"
- *learnability*: the improvement in task performance in the second trial.

Figure 10 summarizes the results of the quantitative usability attributes measurements recorded throughout the usability tests.

There were also a number of qualitative usability attributes that have been evaluated through interviewing the participants:

- *user satisfaction*: the attitude of users toward using the application
- *simplicity*: the degree of comfort with which users find a way to accomplish tasks
- *comprehensibility*: how easily users can understand content presented on the mobile device
- *perceived usefulness*: to what extend the application has met its implementation objectives
- *system adaptability*: how the system adapts to the user requirements.

Each test session ended with an interview (following a verbal protocol), in which participants qualitative remarks about the application and the test session were recorded.

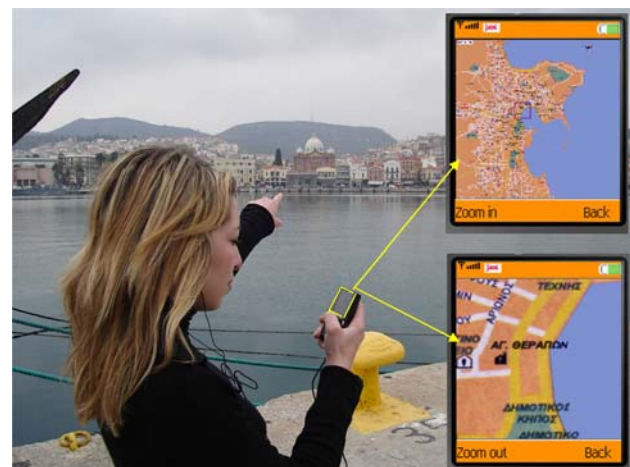


Fig. 9 A participant navigating with myMytileneCity guide in the city of Mytilene using the two-levels 2D city map

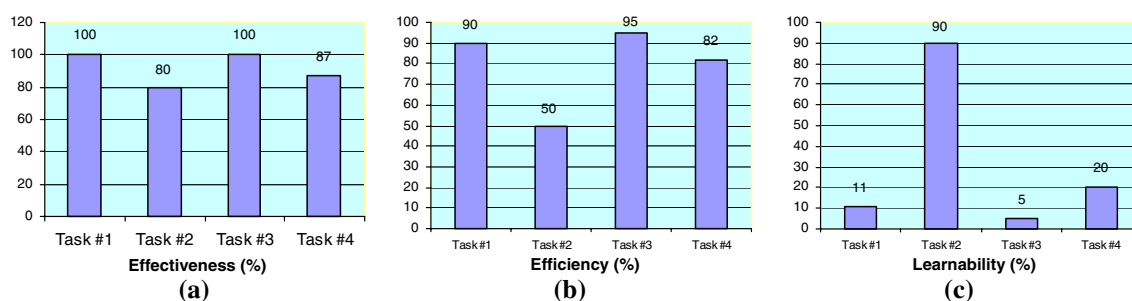


Fig. 10 Measurement of quantitative usability attributes: **a** effectiveness; **b** efficiency; **c** learnability

Finally the participants were also asked to fill out another short questionnaire (to record their remarks and assist in assessing the qualitative usability attributes) which was collected in the end of the session.

The feedback from the participants on the myMytileneCity project as well as the application prototype was in overall positive, although some participants also gave critical comments. Participants tended to believe that this kind of tourist guide application would be most beneficial for casual tourists and mainly as a planning and information tool. myMytileneCity guide was generally considered as nice-to-have rather than need-to-have service and participants would like to see it as an entertaining guide tool with some ready-made scenic routes through the city. Its biggest weakness was the absence of GPS support (although it is questionable whether built-in GPS will soon be widely offered by mobile phone manufacturers), but participants also made several other useful improvement and enhancement suggestions, listed at the end of this section.

Participants (even not experienced web users) found it quite easy to select content from the myMytileneCity web site and then download and load the application on their mobile phone. However, most of them pointed out that the over-the-air application download time (~ 1 min) was too long. On the other hand, the over-the-air download was found more easy and straight-forward to deal with compared to downloading the application on their desktop first and then to their mobile phone through a bluetooth connection. In fact, almost half of the participants had never before used the bluetooth connectivity service of their mobile phone.

Participants appreciated the adaptation of the application's content to their personal needs and preferences and the fact that the application was accessible through their personal portable devices that they were already familiar with. They also agreed that those people and companies who get financial benefit from being visible in myMytileneCity guide (e.g., shops, agencies, restaurants) should pay part of the expenses. The amount and quality of the available tourist information was found really important for

the participants and have an effect on their satisfaction and willingness to use the application.

Several participants' suggestions for improving the application's content and introducing new services have been collected. A compilation of these suggestions follows and will certainly be taken into account in future releases of the electronic guide:

- positioning support (GPS), highlighting the user's current position on the map
- exchange of tourist reviews/impressions/suggestions with other tourists, especially with those that share similar tourist interests
- inclusion of emergency contacts in the city map (pharmacies, hospitals, police stations, port and tourist police authorities, etc.) and search facilities to locate the shortest contacts, depending on the current location of the user
- inclusion of 3D city maps to ease the recognition of the surrounding environment
- enhancement of digitized maps so as to provide three zoom levels (instead of two), target highlighting, street names and clickable objects, like city attractions, to provide quick access to important tourist information
- provision of daily/weekly weather forecast reports
- provision of multimodal orientation and navigation features (e.g., synthetic speech in addition to textual and graphical information)
- suggestion of daily, personalized tourist itineraries that will dynamically adapt on current weather conditions.

8 Conclusions and future work

In this article a J2ME-based "mobile tourism" research prototype has been presented. The main design objectives have been:

- to enable the automated creation of portable, personalized tourist applications (optimized for the specific

Fig. 11 Use case for downloading the tourist application to the user’s mobile device from the myMytileneCity web site

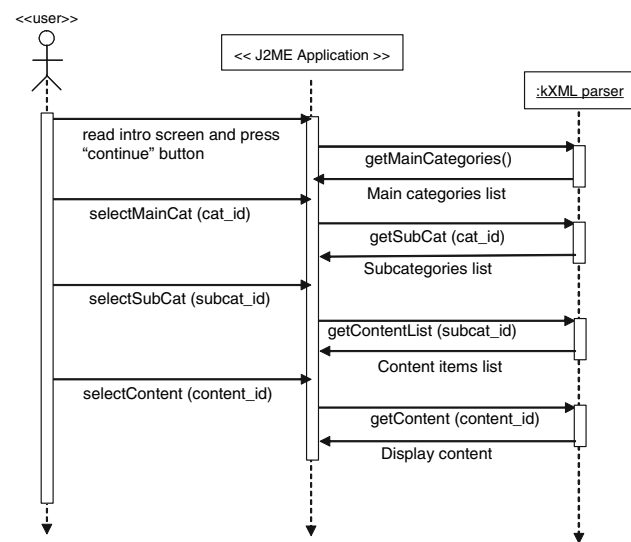
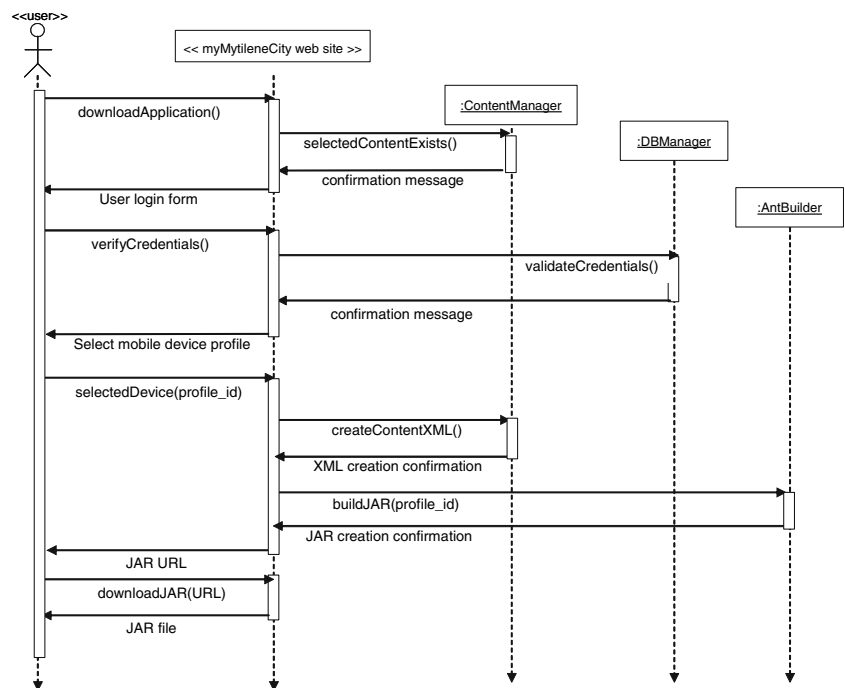


Fig. 12 Use case for the user-tourist application interaction

user mobile device’s model) with rich and customized content

- to minimize the wireless connectivity requirement of the mobile tourist guide application user (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)
- to cater for future dynamic application updates based on a “push model”, wherein new tourist content is pushed to the mobile terminal with minimal user

intervention as soon as it is added by the administrator to the back-end database.

The proposed 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.

Extensive usability evaluation tests performed both in laboratory environments and though field studies revealed a high degree of satisfaction among participants although several critical comments for improving the application’s content and introducing new services have been collected.

Regarding future work, the following directions are going to be studied:

- use of the optional MIDP “Location API” [28] 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)
- design and implementation of algorithmic solutions (that take into account several parameters such as the user profile, the period of stay, the whether conditions, etc.) for suggesting near-optimal, daily tourist itineraries for tourists interested in visiting a specific set of sites (e.g., museums, archeological sites, parks, zoos, etc.).

Fig. 13 Use case for updating the tourist application with new content items selection

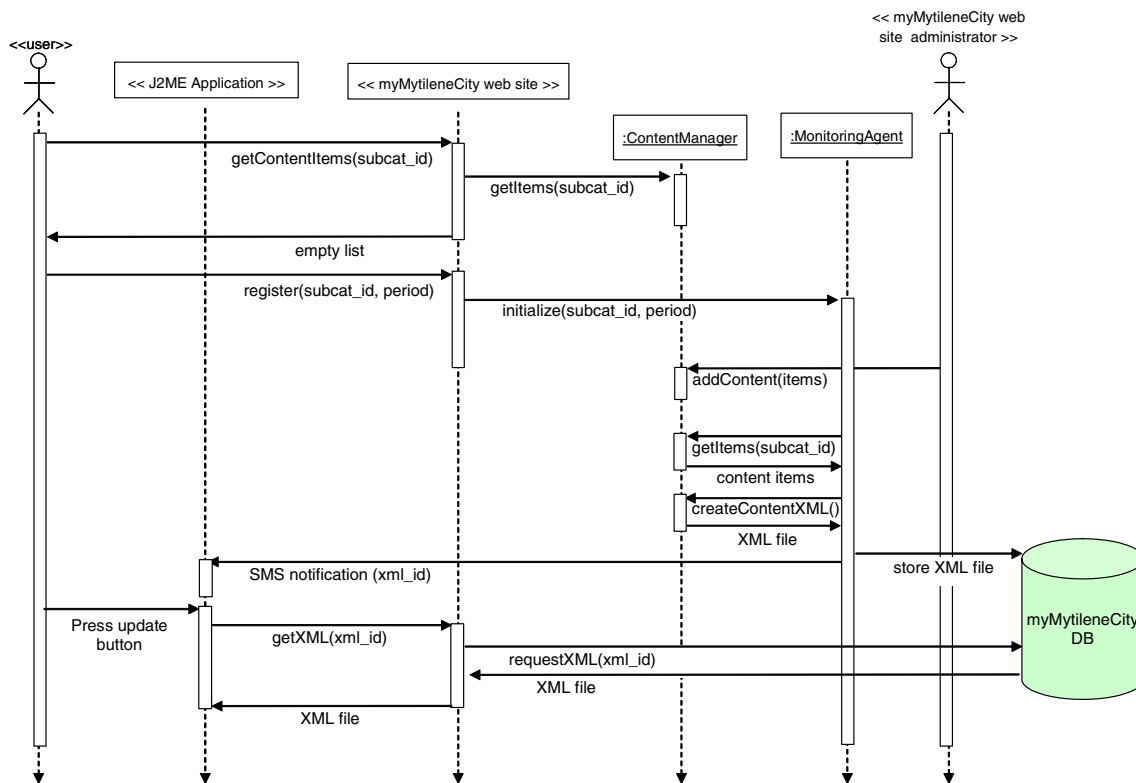
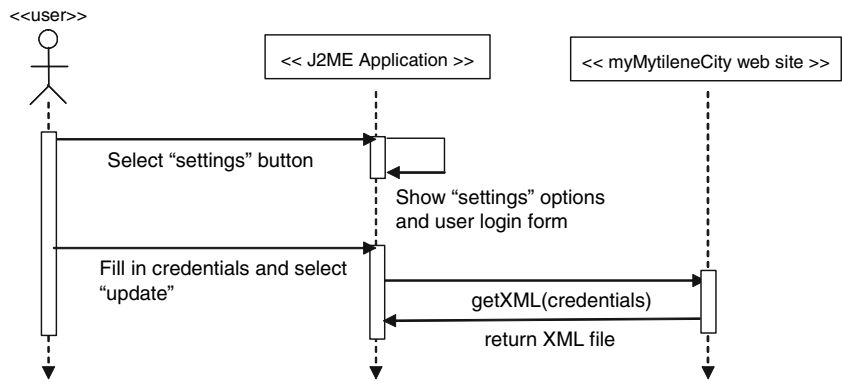


Fig. 14 Use case for pushing tourist content to the mobile tourist application

- design and implementation of a mobile peer-to-peer based framework using the JXME platform [30] for tourist-to tourist communication, in order to bypass the need for downloading content via mobile carriers networks where charges apply.

the use case for downloading the tourist application to the user’s mobile device. Figure 12 depicts the use case for the user—tourist application interaction, while Fig. 13 shows the use case for updating the tourist application with new content items selection. Finally, Fig. 14 presents the use case for pushing tourist content to the mobile tourist application.

9 Appendix: application design

This section includes a series of sequence unified modelling language (UML) diagrams that clarify our tourist application’s design and functionality. Figure 11 illustrates

References

1. Anegg H, Kunczier H, Michmayr E, Pospischil G, Umlauf M (2002) LoL@: designing a location based UMTS application. *Elektrotechnik und Informationstechnik* 191(2):48–51

2. Baus J, Kray C, Krueger A (2001) Visualization of route descriptions in as resource-adaptive navigation aid. *Cogn Process* 2(2–3):323–245
3. Bederson B, Clamage A, Czerwinski MP, Robertson G (2003) A Fisheye calendar interface for PDAs: providing overviews for small displays. In: *Proceedings of the international conference on human factors in computing systems (CHI'03)*, pp 618–619
4. Bisignano M, Di Modica G, Tomarchio O (2005) A JXTA compliant framework for mobile handheld devices in ad-hoc networks. In: *Proceedings of the 10th IEEE symposium on computers and communications (ISCC'2005)*, pp 582–587
5. Brewster S, Dunlop M (eds) (2004) *Mobile human–computer interaction—Mobile HCI 2004*. Lecture Notes in Computer Science, vol. 3160, ISBN: 3-540-23086-6
6. Brown B, Chalmers M (2003) Tourism and mobile technology. In: *Proceedings of the European conference on computer supported collaborative work (CSCW'2003)*, Kluwer, Dordrecht, pp 335–355
7. Brown B, Perry M (2001) Of maps and guidebooks: designing geographical technologies. *ACM SIGGROUP Bull* 22(3):28–32
8. Buyukkokten O, Garcia-Molina H, Paepcke A, Winograd T (2002) Efficient web browsing on handheld devices using page and form summarization. *ACM Trans Inf Syst* 20(1):82–115
9. Canadian Heritage Information Network (CHIN) (2004): tip sheets, personal digital assistants (PDA), PDA aesthetics and interface design, creating and managing digital content. http://www.chin.gc.ca/English/Digital_Content/index.html. Accessed 10 April 2006
10. Cheverst K, Davies N, Mitchell K, Friday A, Efstratiou C (2000) Developing a context-aware electronic tourist guide: some issues and experiences. In: *Proceedings of the 2000 conference on human factors in computing systems (CHI'00)*, pp 17–24
11. Chittaro L, Cin PD (2002) Evaluating interface design choices on wap phones: navigation and selection. *Pers Ubiquitous Comput* 6:237–244
12. Ciavarella C, Paterno F (2003) Design criteria for location-aware, indoor, PDA applications. In: Chittaro L (ed) *Proceedings of human–computer interaction with mobile devices and services, 5th international symposium mobile (HCI 2003)*, pp 131–144
13. Dublin Tourism on the Wireless Web, <http://www.visitdublin.com/wireless/>. Accessed March 2006
14. Dunlop M, Brewster S (2002) The challenge of mobile devices for human computer interaction. *Pers Ubiquitous Comput* 6(4):235–236
15. Economou D, Gavalas D, Varitimiadis S, Samaroudi M (2006) The effectiveness of personal digital assistants (PDAs), audio-guides or paper leaflets to enhance the museum visit experience. In: *Proceedings of the 3rd international conference on museology (ICM'2006)*
16. Gavalas D, Economou D (2007) The technology landscape of wireless Web. *Int J Mob Commun* (in press)
17. Goldman SV, Pea R, Maldonado H, Martin L, White T (2004) Functioning in the wireless classroom. In: *Proceedings of the 2nd IEEE international workshop on wireless and mobile technologies in education (WMTE'04)*, pp 75–82
18. Gawor N (2006) Invoking Apache ant programmatically. IBM Developer works, http://www-128.ibm.com/developerworks/websphere/library/techarticles/0502_gawor/0502_gawor.html. Accessed June 2006
19. Gupta A, Srivastava M (2001) Integrated Java technology for end-to-end m-Commerce. <http://developers.sun.com/techtopics/mobility/midp/articles/mcommerce/>. Accessed February 2006
20. Ho SY, Kwok SH (2003) The attraction of personalized service for users in mobile commerce: an empirical study. *ACM SIGecom Exch* 3(4):10–18
21. Hochstatter I, Küpper A, Schiffers M, Köthner L (2003) Context provisioning in cellular networks. In: *Proceedings of the 8th IEEE international workshop on mobile multimedia communications*
22. Holtsblatt K (2005) Designing for mobile devices: experiences, challenges and methods. *Commun ACM* 48(7):33–35
23. i-area: Location Based Services, <http://www.nttdocomo.com/corebiz/services/imode/iarea.html>. Accessed February 2006
24. J2ME Polish, <http://www.j2mepolish.org/>. Accessed March 2006
25. Java 2 Platform Micro Edition (J2ME), <http://java.sun.com/j2me/>. Accessed November 2005
26. Java API for XML Processing (JAXP), <http://java.sun.com/webservices/jaxp/index.jsp>. Accessed March 2006
27. JavaServer Pages Technology, <http://java.sun.com/products/jsp/>. Accessed March 2006
28. JSR 179: Location API for J2ME, <http://www.jcp.org/en/jsr/detail?id=179>. Accessed March 2006
29. JSR 184: Mobile 3D Graphics API for J2ME, <http://www.jcp.org/en/jsr/detail?id=184>. Accessed March 2006
30. JXME: JXTA Java Micro Edition Project, <http://jxme.jxta.org/>. Accessed June 2006
31. Kenteris M, Gavalas D, Economou D (2006) A novel method for the development of personalized mobile tourist applications. In: *Proceedings of the 5th IASTED international conference on communication systems and networks (CSN'2006)*, pp 208–212
32. Kjeldskov J, Stage J (2003) New techniques for usability evaluation of mobile systems. *Int J Hum Comput Stud* 60:599–620
33. Knudsen A (2003) Cheaper parsing of XML on mobile devices. White paper, <http://www.idi.ntnu.no/grupper/su/fordypningsprosjekt-2003/fordypning2003-Andreas-Knudsen.pdf>. Accessed May 2006
34. Kray C, Baus J (2003) A survey of mobile guides. In: *Proceedings of HCI in mobile guides, in conjunction with the 5th international symposium on human computer interaction with mobile devices and services*, pp 1–5
35. Kray C, Laakso K, Elting C, Coors V (2003) Presenting route instructions on mobile devices. In: *Proceedings of the 2003 international conference on intelligent user interfaces (IUI'03)*, pp 117–124
36. kXML, <http://kxmlrpc.objectweb.org/>. Accessed March 2006
37. Micha K, Economou D (2005) Using personal digital assistants (PDAs) to enhance the museum visit experience. In: *Proceedings of the 10th panhellenic conference on informatics. LNCS*, vol 3746, pp 188–198
38. NTT DoCoMo, <http://www.nttdocomo.com/>. Accessed March 2006
39. Oppermann R, Specht M (2000) A context-sensitive nomadic exhibition guide. In: *Proceedings of the 2nd symposium on handheld and ubiquitous computing (HUC2K)*, pp 127–149
40. Preece J, Rogers Y, Sharp H (2002) *Interaction design: beyond human–computer interaction*. Wiley, Danvers
41. ProGuard, <http://proguard.sourceforge.net/>. Accessed May 2006
42. Shim JP, Varshney U, Dekleva S, Knoerzer G (2006) Mobile and wireless networks: services, evolution & issues. *Int J Mob Commun* 4(4):405–417
43. Simplewire, Java SMS SDK, <http://www.simplewire.com/developers/sdk/java/>. Accessed April 2006
44. Sun Java Wireless Toolkit, <http://java.sun.com/products/sjwtoolkit/>. Accessed March 2006
45. Varshney U (2003) Issues, requirements and support for location-intensive mobile commerce applications. *Int J Mob Commun* 1(3):247–263
46. Wireless Application Protocol (WAP) Forum, <http://www.wapforum.org/>. Accessed March 2006
47. W3C Mobile Web Best Practices 1.0, Basic Guidelines, W3C Candidate Recommendation, <http://www.w3.org/TR/>

- [2006/CR-mobile-bp-20060627/#requirements](#). Accessed 27 June 2006
48. Yuan MJ (2004) Enterprise J2ME: developing mobile Java applications. Prentice-Hall, Englewood Cliffs
49. Zhang D, Adipat B (2005) Challenges, methodologies and issues in the usability testing of mobile applications. *Int J Hum Comput Interact* 18(3):293–308