Journal of Mobile Multimedia, Vol.3, No.1 (2007) 065-087


# MULTIMEDIA APPLICATIONS FOR HANDHELD DEVICES: ANALYSIS OF REQUIREMENTS FOR DEVELOPMENT PLATFORMS AND APPLICATION AUTHORING TOOLS

DAPHNE ECONOMOU, DAMIANOS GAVALAS, MICHAEL KENTERIS AND KATY MICHA

*Department of Cultural Technology & Communication*

*University of the Aegean, Greece*

*d.economou@ct.aegean.gr, dgavalas@aegean.gr, ctm04007@ct.aegean.gr, k.micha@ct.aegean.gr*

Received July 23, 2006
Revised October 17, 2006

This paper explores requirements application authoring tools should satisfy for the development of cultural applications tailored for deployment on Personal Digital Assistants (PDAs) and mobile phones. The paper reviews the use of mobile technologies in the context of cultural organizations and tourism. It identifies and evaluates the development and design facilities provided by state-of-the-art multimedia application development tools for PDAs and mobile phones: Macromedia Flash Lite, Navipocket and Java 2 Micro Edition. It describes the way these tools have been used in the implementation phase of two projects that have been developed at the Cultural Heritage Management Lab (CHMLab), at the Department of Cultural Technology and Communication, University of the Aegean. These projects focus on the use of PDAs and mobile phones for providing cultural and tourist information, keeping the visitors' interest and attention, as well as promoting various cultural organizations' and tourist facilities. Based on these two case studies the paper extracts a set of PDA and mobile phone application requirements. The paper concludes with a set of suggestions related to the way application authoring tools should be exploited in order to gratify application and designer needs for developing operational and profitable cultural and tourist applications.

*Keywords:* Application authoring tools, development platforms, requirements, cultural and tourist multimedia applications, Navipocket, J2ME
*Communicated by*: I. Ibrahim & K. Liu

## 1 Introduction

There has been a considerable amount of research on the use of multimedia technologies in the fields of cultural organizations and tourism for the provision of cultural interpretational information [14, 27, 33, 38, 40]. Mobile devices have also been gaining increasing acceptance as a means to provision cultural multimedia applications due to their physical characteristics and suitability in these fields. This is evident by a number of research prototypes and commercial projects that have been reported in literature [3, 5, 10, 14, 25, 28, 33, 34]. Currently, the tools used to develop multimedia applications for mobile devices are light versions of state-of-the-art multimedia application authoring tools, which are not tailored to adequately satisfy user, designer and mobile device applications requirements.

The aim of this paper is to identify a coherent set of requirements for the implementation of cultural applications on PDAs and mobile phones. This exercise builds upon the evaluation of current state-of-the-art application authoring tools and on experience gained by using such tools in two case studies developed in our laboratory. The first case study focuses on the use of PDAs for the provision of interpretative cultural information in a museum environment, whilst the second is a mobile tourist guide for the city of Mytilene, Greece. The case studies have been developed using Navipocket authoring tool and J2ME development platform, respectively.

The remainder of the paper is structured as follows. Section 2 reviews projects related to the use of mobile devices in cultural organizations and tourism. Section 3 discusses issues related to the design of PDA and mobile phone cultural and tourist applications. Section 4 describes the main features of typical tools for multimedia application development for PDAs and mobile phones. Section 5 presents the two case studies and Section 6 identifies a set of requirements for cultural application authoring tools and development platforms. Finally, Section 7 draws conclusions about this work.

## 2 A review of the use of mobile devices in Cultural organizations and Tourism

### *2.1 The use of mobile devices in Cultural organizations*

Multimedia and new technologies provide unique opportunities to museums as they bring new ways of communication and interpretation. Technological solutions known by now, like projection systems and info-kiosks, successfully connect artefacts to information. However, these solutions are limited to the museum physical space. Audio-guides are successful examples of connecting artefacts to information in a portable way, which is not limited by the physical environment, but it is restricted to only using audio.

PDAs technology allows the dynamic presentation of information, without being limited by, or being encroached by the aesthetics of the galleries. The introduction of handheld computers in museums for enhancing visitor experience was inspired by Zaurus, Psion, and Newton [2], that provided the museum visitor the opportunity to access multimedia, text and audio while walking through the exhibitions. Moreover, in contrast to audio-guides, users could now follow a non linear path of exploring the information provided. The “pocket curator” in 1995, was a system that made personalized experience possible by providing interactive audio and text about biographical information and interpretive annotations for 15 works of art (Fernando Quintero). In 1997 the Smithsonian Institute launched a travelling exhibition named “America's Smithsonian,” which used handheld computers iGo to lead an interactive tour, which provided information on 90 objects of the exhibition using narration, and text graphics. In 1999 the Smithsonian Institute included the handheld computer Rocket e-Book at the “On Time” exhibition, which supported hypertext files, sound, and B&W graphics. Museum futurists showed great interest in wireless networks and location-aware technologies. The project Hyper Interaction with Physical Space (HIPS) was one of the first projects that experimented with mobile computing and location awareness in museums [14]. The Hippie/Hips project was based on the use of infrared beacons strategically installed to track location of visitors of an exhibition [31]. The visitors’ choices were traced in the form of user profile, from which more exhibition suggestions were recommended. Also the MIT lab is known for its interest in wearable computing through the work of Sparacino in interactive exhibit design [34]. Tate Gallery gives emphasis on museum interpretation. Tate Modern Gallery launched two pilot

multimedia tours whose mission was to "to test both applications of wireless technology in the gallery, and to access a wide range of approaches to content design" [33]. In Greece, the first attempt of using PDAs to aid the museum visit is the "Fables on pocket PC" project for the Museum/Library Stratis Eleftheriadis Teriade in Lesvos, Greece [28]. This project uses PDAs to provide enriched multimedia interpretative information for the collection of "Fables" by Jean de La Fontaine, which have been illustrated by Marc Chagall and are exhibited in part of the Teriade Museum.

PDAs afford to be used as guides that provide enriched information in outdoor archaeological sites. PDAs on site fill the gap left by conventional paper guidebooks, info kiosks and audio guides and provide truly mobile devices with navigation, personalization, and interactivity features. Two such exemplar projects that are based on advanced mobile IT to provide new approaches for accessing information at cultural heritage sites are the Archeoguide [37] and the "Nara Palace Site Navigator" [35] projects.

Although the audience's feedback was positive towards the use of PDAs in exhibitions, the museums did not incorporate them because of practical concerns. Issues included: the fragility of the devices; the frequent need to recharge batteries; the current high cost of the devices. Among cultural organizations in Greece, with the exception of Archeoguide mentioned above, there have not been many examples of using PDAs. However, the interest for enhancing the museum experience by providing information and interpretation, as well as promoting various museum facilities with the use of handheld technologies has expanded lately and there has been a considerable amount of research activity on this subject.

### *2.2 The use of mobile devices in Tourism*

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. The Internet represents a medium which is well tried and tested on many successful business models related to services provision.

At the same time, the rapid growth of mobile devices (mobile phones, palmtops, PDAs, etc) user base raises a demand for using such devices for accessing Internet resources. 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; tourists are amongst this technology-oriented public (e.g. [12]). Hence, a growing body of commercial and research initiatives that incorporate electronic tourist guide functionality into mobile devices have been reported [24]. 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 [7]. In the following paragraph, we review several commercial and research projects with respect to mobile tourist guides. 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 [25]. 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 LoL@ (LOcal Location assistant) project was a mobile tourist guide designed for the city of Vienna using the UMTS Mobile Telecommunication system [3]. 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 [5]. 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. 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. Therefore, the usage of mobile phones as mobile tourist guides that do not necessarily rely on network connection to function still needs to be intensively re-searched. The myMytileneCity Guide (developed at our laboratory and presented in 5.2) addresses this issue by allowing the generation of tourist guide applications over the Internet which can later execute on any mobile platform with no network connection requirement.

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

## 3 PDA and mobile phone application requirements

### *3.1 Issues related to the design of PDA and mobile phone applications*

According to Dunlop and Brewster [13] for the design of successful PDA applications various factors should be considered which are mainly related to:

- the technical characteristics of the devices
- the use of the application

These factors charge the PDA application designers with new challenges, such as:

- design for mobility
- design for a wide user community
- design for limited input/output facilities
- design for providing information based on the user location
- design for user multitasking at levels unfamiliar to most desktop users

However, several issues differentiate the requirements for PDAs and mobile phone applications. The major difference between PDAs and mobile phones is currently screen size. For the design of mobile applications, one must consider the limited screen space the mobile phone has to offer [17]. The up springing of high end mobile devices has seen the use of larger screen sizes, which however still represent the minority of the market share of mobile phone owners. Designers of mobile phone applications have to consider three main categories of screen sizes when designing applications [17], unlike PDAs where standards in screen sizes do not vary considerably. In addition, most available mobile phones are not yet fully multimedia enabled.

Furthermore, the number of dedicated mobile phone applications are still few. Thus, users can not easily draw on past experiences in the use of such applications [13]. 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. Hence, applications targeting mobile phones need to be highly optimized and customized to meet a broad range of user and devices requirements.

### *3.2 Issues related to the design of PDA and mobile phone applications for cultural organizations and tourism*

The design of PDA applications for cultural organizations (e.g. museums) should address the organizations' requirements and provide the visitors a pleasant experience [11]. Woodruff et al. [40] studied the visitor behavior using PDAs in museums and they identified certain issues related to the user interaction with the PDA application:

- the visitors should be provided with visual feedback for their selection
- the information presented should be short and the system should support audio presentation of the information
- the audio information should not interfere with the interaction between the visitors in the museum

In terms of the interface design, the PDA applications should pursue criteria similar to web sites development [11]. In addition, the Canadian Heritage Information Network (CHIN) adds some practical guidelines [8] for the graphic design of the interface:

- each screen node of the PDA application should fit the size of the PDA screen
- the navigation should be structured hierarchically
- backtrack and easy access to the home page should be supported

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 [32].

As far as mobile phones applications are concerned most of research work and prototypes are restricted to tourist applications. This is mainly due to the physical restrictions of this class of devices.

Kray and Baus in 2001 completed a survey of available tour guide applications [24]. Many issues were retrieved relating to mobile phone applications design:

- the use of Location-Based Services (LBS) is recommended to take into account user and context related information
- adaptation capabilities should be enabled (e.g. the system should adapt to the lack of network connectivity)
- the user interface should support multimodal communication, natural language and multilingualism.

## 4 Application authoring tools and application development platforms

This section presents the development and design facilities provided by some typical tools for multimedia application development and services provision for PDAs and mobile phones, which are classified in:

- application authoring tools
- application development platforms

Application authoring tools offer features that accelerate the development process requiring some or no intensive programming effort. While application development platforms require immense amount of effort, which is balanced by the flexibility in application configuration and development they offer.

### *4.1 Application authoring tools*

In this section two application authoring tools are reviewed, namely, Flash Lite by Macromedia and NaviPocket v. 2.4 by OPHRYS SYSTEMS. The latter tool attracts attention as it has been used in one of the case studies examined later on in the paper, while the former tool is reviewed due to its popularity for designing and developing multimedia application for mobile devices by the relevant industry players.

#### 4.1.1. Macromedia Flash Lite

Macromedia Flash Lite [1] 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 1) 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.

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.

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.

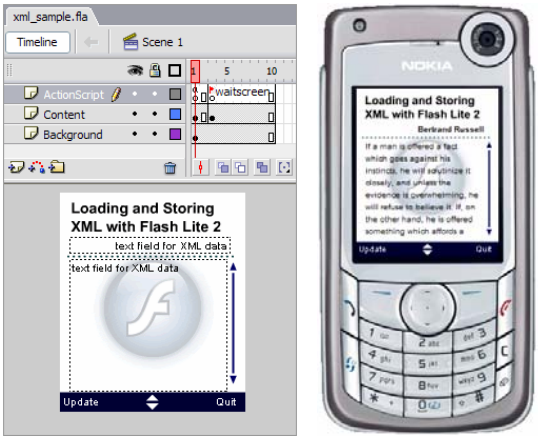



Figure 1. Screens of a Flash Lite authoring environment and the result of its execution shown by an emulator.

### 4.1.2. Navipocket

NaviPocket v. 2.4 by OPHRYS SYSTEMS has been designed to meet the demands of Theme and Leisure Parks, Museums and Cultural sectors (gardens, zoos, aquariums, etc.) in developing multimedia guides. Navi-pocket is an authoring tool which allows the creation of multimedia applications on electronic message minders of PDA type. It is a software unit aimed for portable systems (PDA or TabletPC-type) supporting an embedded OS (Version 1 functions under Microsoft Windows EC 2.xx and PocketPC). The current version works with Microsoft PocketPCTM 2002 and Windows Mobile 2003. A PalmOS version will be available soon.

The product is a complete set of an “Editor”, a “Simulator” and a “Run-time”. The software benefits the developed applications offering speed of execution, economy of resources, and facility of use without stylus. The editing and simulation programs allow a “content” designer to create all the files that will be operated by NaviPocket (see Figure 2). Within the Editor Module, the user creates a set of pages. These pages are in text format and are built according to an object-oriented model. Each page is an object which can contain other objects. Navipocket supports the following objects: page, button, text area, bitmap and video. Each object has properties and can be linked (hypertext-type dynamic link) with another object.

Application authoring tools like Navipocket and Flash Lite offer the great advantage of expediting the delivery of advanced applications and content services. However, both tools are not open source, they do not support dynamic content maintenance and they require MS Windows compatible devices for the development of multimedia projects, as well as for the run-time.

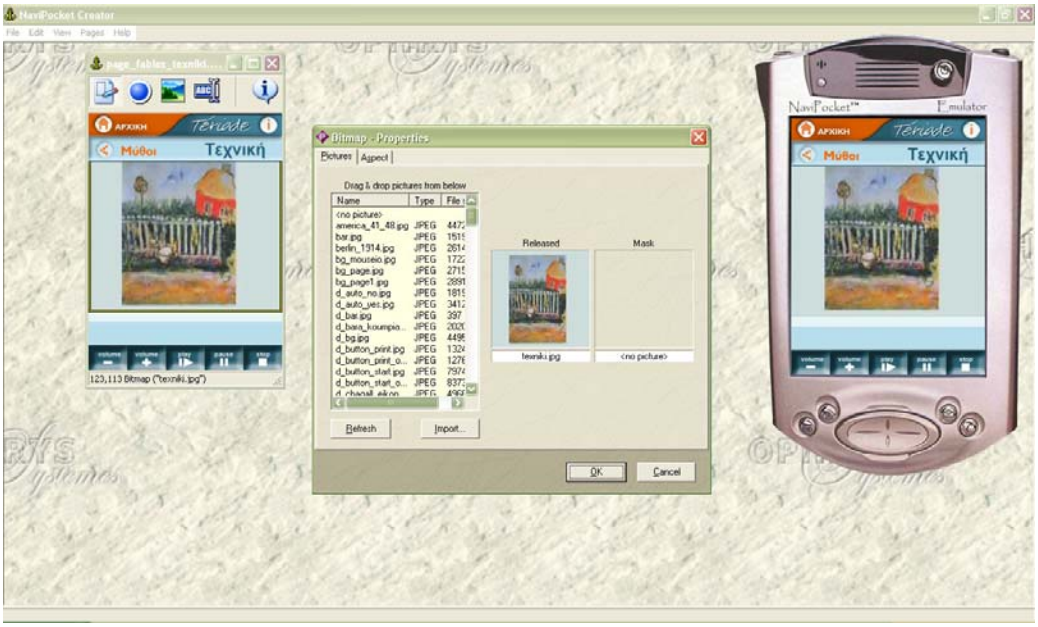



Figure 2. The Navipocket "Editor" and "Simulator" environments.

### *4.2. Application development platforms*

In this section two application authoring platforms are reviewed, namely, Microsoft Technologies for Mobile Application Development and Deployment and Java 2 Micro Edition. The latter tool attracts attention as it has been used in one of the case studies examined later on in the paper, while the former platform is reviewed due to its popularity for developing application for mobile devices.

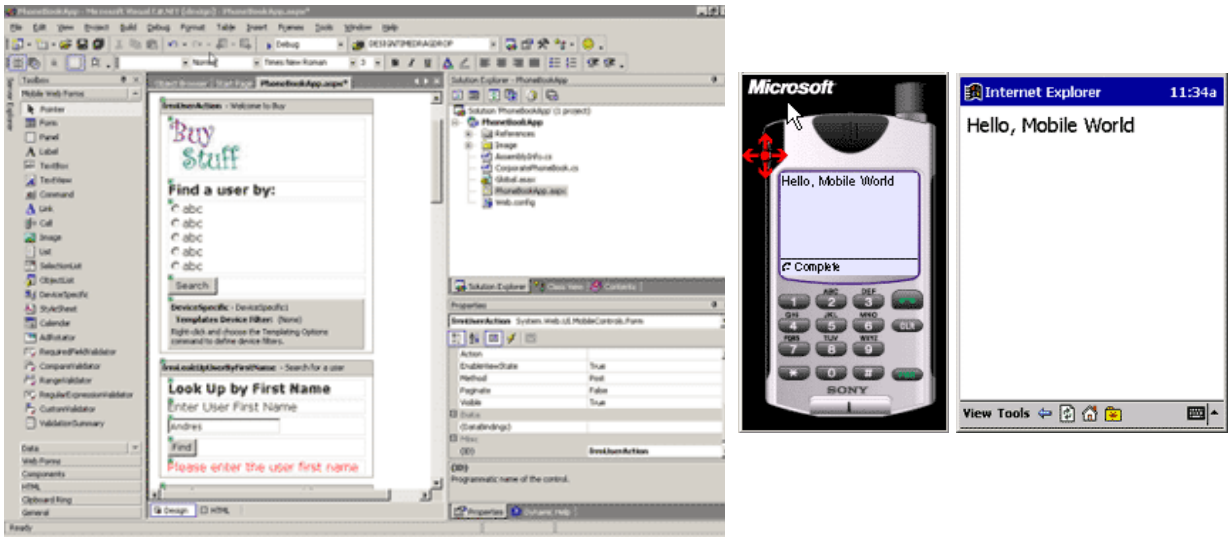
#### 4.2.1. Microsoft .Net platform for the Mobile Web

The ASP.NET mobile controls [4] (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)[a] development environment (see Figure 3). 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

[a] 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.

mechanisms; developers simply utilize ASP.NET pages (for no particular target device) which automatically identify the device that posted a requestb and render the appropriate content.




(a) (b)

Figure 3. (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.

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.

#### 4.2.2. Java 2 Micro Edition

WAP [39] and i-mode [30] technologies have recently emerged to enable the use of wireless infrastructures for Internet access and realize the vision of wireless Internet. Lately, a number of WAP and i-mode official and unofficial services have been deployed providing tourist [12], context-aware

[b] 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.

[16] and location-based services [36]. The main disadvantage of i-mode/WAP-based services is their requirement for constant connection (air-time) 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') he/she cannot access any service.

The need for defining a computing platform that could execute Java applications and be supported by small electronic devices led to the development of Java 2 Micro Edition (J2ME) [19] by Sun Microsystems, in 1999. J2ME is a framework for developing applications executed on resource-constrained devices. It also offers strong wireless networking support enabling applications to access a broad range of web content formats.

J2ME 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. 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 con-figuration currently supported by all J2ME-enabled mobile devices is the CLDC (Connected Limited Device Configuration).

Profiles are implemented on 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 top of a particular configuration/profile pair and are based on their specified class libraries. Currently, the only available profile specified on top of CLDC is MIDP (Mobile Information Device Profile). 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 indicate its suitability for developing cultural -and more specifically- 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. 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).
- 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 [15].

- 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 [15].

On the other hand, J2ME technology has several weaknesses that should be carefully considered. First, J2ME applications have increased requirements on device resources: storage, processing power and memory. Second, MIDlets programming is not straightforward as it requires Java development skills. Evidently, the development of J2ME applications is far more complex compared to creating content using developer-friendly application authoring tools like Flash Lite or Navipocket.

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. All these issues have been considered in the design and implementation of the mobile tourist guide prototype presented in Section 5.2.

## 5. Case studies

In this section two case studies are presented. The former evaluates Navipocket as an authoring tool for the development of a cultural multimedia application on a PDA and the latter evaluates J2ME as a development platform for the implementation of a tourist guide on a mobile phone.

### *5.1. The use of a PDA to provide interpretative information for a set of illustrations*

In this section, we present a PDA case study, the “Fables on pocket PC”. This project focuses on the study of real requirements for the development of museum multimedia applications for PDAs that enhance visitor’s experience and provide information and interpretations about the museum collections. The project uses the Mu-seum/Library Stratis Eleftheriadis Teriade in Lesvos, Greece, as a case study and in particular the collection of “Fables ” by Jean de La Fontaine, which have been illustrated by Marc Chagall and are exhibited in part of the Teriade Museum .

In its galleries Teriade Museum exhibits the only illustrations of the “Grant Livres”, the “Verve” and medieval manuscripts (see Figure 4b). The illustrations are disconnected from the story they supplement (see Figure 4a). The only information that connects the exhibited illustrations to their originating source is presented in textual form on a few supporting panels and is limited to referencing the book and the artist that created them. This makes it difficult for the visitor to understand the collection and appreciate its importance. Then again the Teriade Museum appreciates the visitors’ requirements of being provided with more information about the collection, and connecting the illustrations to the stories, however, this should not superimpose the physical site.

The use of PDAs technology suits adequately to the Teriade Museum requirements, as it offers a very rich way of:

- connecting the stories to the illustrations,
- providing general and interpretative information about the Fables’ illustrations that allows the visitor to understand the collection,

- helping visitors to focus their attention on specific items of the collection,
- allowing the visitors to create their own personal opinion about the collection
- at the same time provide the visitors a pleasant experience

without encroaching on the aesthetics of the gallery space. In addition, the system allows the visitors to select content (text and images) that meets their personal interest and take it with them after their visit at the museum in print (as a kind of souvenir).

The content which was included in the application was especially selected in order to reflect the information which is of value to certain stakeholders. This set of stakeholders consists of: the Teriade Museum curators, museum management, art historians who might provide various interpretations for the collection and visitors. This stakeholder-oriented method of content requirements analysis evolved out of the Soft Systems approach [9].


(a) (b)

Figure 4. (a) A snapshot of two pages of the "Fables" by Jean de La Fontaine that contains the story and the illustration by Marc Chagall. (b) One of the walls where the collection of "Fables" illustrations are exhibited at the Teriade Museum exactly the way that Teriade set it up. The stories are disconnected from their illustrations and no interpretational information about the collection is presented.

The multimedia content format which was finally included in the application was videos made of narration and animated pictures. Text was not used as visitors cannot afford the time to read the text while moving through the gallery [38]. Furthermore, PDA devices are too small to provide long textual descriptions, therefore audio was chosen as the preferred medium [33]. The videos which were included did not exceed 1 minute duration, as long descriptions are tiring for the visitor [33, 40]. Each screen node of the PDA application fits the size of the PDA screen, so the user won't have to scroll to view all the content. These design solutions address basic interaction with PDA applications requirements, like keeping presented information short and providing audio information in a way that it does not interfere with the interaction between the visitors in the museum (see Section 3.2).

Content is structured hierarchically and backtracking is supported, so the user can easily return to the home page and navigate to other sections. The user can navigate through the content by choosing manually the sections of their interest, or the content to be presented can be automatically selected by the software, since it makes use of infrared based technology which determines the visitors' position in the gallery. The user can choose to switch between manual and auto guidance at any time while using the system. These design solutions address CHIN's suggestions for developing applications for Cultural Heritage (see Section 3.2).

The application targets a wide audience (wide variety of age groups and familiarity to the use of the technology). Thus, one of the application objectives is to have an easy to use interface, so that the interaction with the application should be intuitive even for users who are not particularly familiar with the use of new technologies. Adaptation capabilities techniques could have been used to address different users' particular requirements. However, these solutions were beyond the purpose of this study, and it would require network connectivity and increase the development and installation cost.

For the design of the application graphical user interface (GUI) and navigation system various issues of usability have been taken into consideration [29, 32] (which have been stated in Section 3.2). The application interface consists of five parts as it appears in Figure 5:

- part 1 remains static in all pages and includes two buttons, the left one leads to the home page, and the right provides information for the Teriade Museum
- part 2 describes the current section the visitors views and includes a button that returns to the previous page
- part 3 includes either navigation buttons that lead to the various sections and subsections of the application, or a video related to the subsections selected
- part 4 includes buttons that allow switching between auto or manual guide
- part 5 includes control buttons for the video and audio

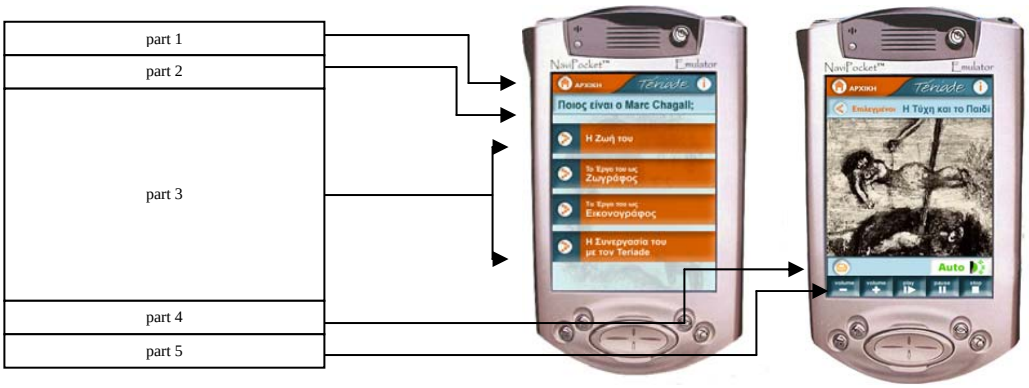



Figure 5. The layout of the "Fables" prototype application.

Buttons appear consistently in the aforementioned positions indicating the hierarchical structure of the information in the application. All navigation and selection button have been clearly labelled with the title of the section or subsection the user wishes to visit and they are decorated with icons demonstrating the user navigation in the hierarchical structure of the information (e.g. > arrows indicate movement in a subsection of the current section, while < arrows indicate navigation one level up or back). Colour has been used consistently in buttons and the icons. Icons are being highlighted providing visual feedback of their selection.

One hundred illustrations of the Fables were digitized and edited for the creation of the videos. For this proto-type only 5 videos have been created. For the narration a male and a female voice have been

used, in order not to tire the visitor. Appropriate background sound has been incorporated to the videos.

The videos have been created with the use of Flash MX 2004 (by Macromedia) and then imported as MPEG in NaviPocket v. 2.4 by OPHRYS SYSTEMS, which was used for the implementation of the application (see Section 4.1.2). The final application is 104 MB (although it is only a prototype and not the final application), due to the large size of graphics and videos that support the multimedia application. These files are stored locally on the PDA storage or an SD card and executed by the Navipocket "Run-time". This size is prohibitive for limited-resource devices like PDAs and mobile phones. This shows that the current state-of-the-art application authoring tools are not adequately tailored to meet the specific requirements of handheld devices as argued in the introduction of this paper.

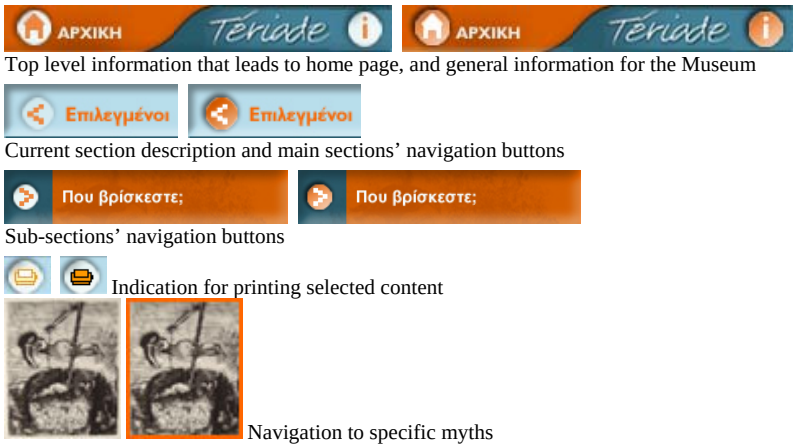



Figure 6. All the above buttons show have two states, the first one is the up state and the second one the down (the selected) state of the button

The development process of the "Fables" prototype demonstrated several aspects regarding the Navipocket authoring tool functionality. In particular, Navipocket seems to offer a number of advantages:

- low application development cost (the software is available free of charge)
- low storage and memory overhead of the Run-time environment
- rapid multimedia content and UI creation
- effortless familiarization of developers with the authoring environment.

However, several drawbacks of Navipocket have also been revealed:

- restrictiveness of the tools provided for the design of expressive UI
- typically large-sized applications, mainly due to the lack of support for a variety of multimedia formats (e.g. if Macromedia Flash files were supported, the conversion to MPEG format would have been avoided and the overall application size would have been significantly reduced)
- dynamic content update is not supported

- dependency on specific operating systems and devices hardware
- lack of specialized libraries for implementing extra functionality (custom solutions for specific customers could be supported by OPHRYS on demand).

### *5.2. The use of a mobile phone as a city tourist guide*

In this section, we present a mobile tourist guide case study, the myMytileneCity guide (an electronic guide for the city of Mytilene, Lesvos Island, Greece). The prototype implementation is entirely based on Java, on both the web server 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 and interoperability (current trends promote XML-based languages for content development and data interchange on the Internet).

The design of the tourist guide application follows a two-step approach: On the first step, the user interested in a particular tourist destination visits a web site including information about restaurants, accommodation, sights, events, night life, etc. The user appends information of interest to his/her '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 Java API for XML Processing (JAXP) [20]. 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.

The web site comprises dynamic pages enabled by JavaServer Pages (JSP) server-side programming technology [21] (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 MySQL JDBC (Java DataBase Connectivity) driver.

On the second step, the user downloads the generated jar file (temporarily saved on the web server) to its mo-bile device. Herein, the user is provided two options:

1. 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
2. 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.

Upon completion of the jar file download to the J2ME-compatible device, the latter is installed and loaded by the local Application Management Software (AMS) module (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 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 [26].

The following figures present various parts of the prototype usage. Figure 7 shows screenshots of the web site used by visitors to select tourist content of interest. Figure 8 demonstrates an excerpt of generated XML code, describing the users' selected content (this code corresponds to the user preferences shown in Figure 7b). Finally, Figure 9 illustrates representative screens of a mobile device emulator (main menu, list of lodging preferences, detailed description of a selected hotel).

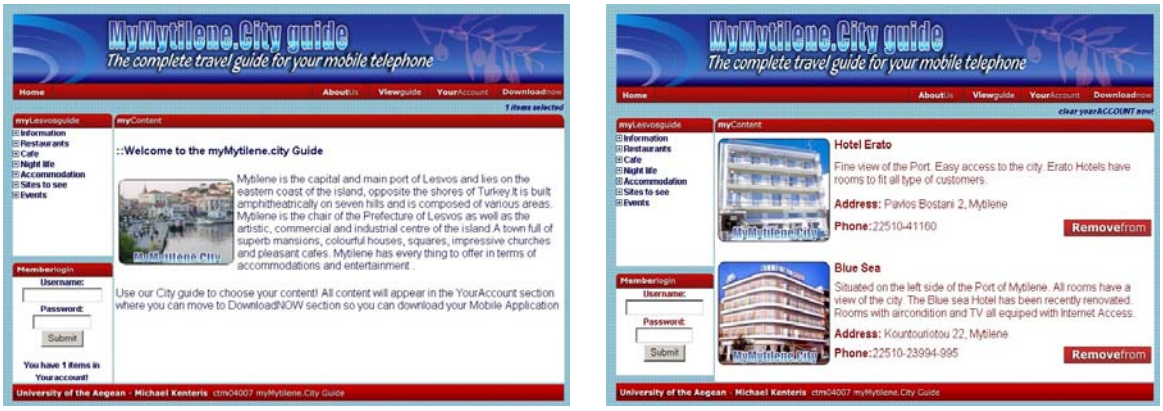



Figure 7. Screenshots from the myMytileneCity web site: main page and contents of user's account (includes two lodging preferences).

The final application size is approximately 520 KB, including the CLDC/MIDP footprint, the kXML parser libraries and the MIDlet jar file (Java classes, XML-based tourist content and the accompanying graphics). This size is considered to be of acceptable volume for today's J2ME enabled devices and contrasts the huge application volume that was generated in Section 5.1 using Navipocket.

The MIDlet application has been designed following the application guidelines for small devices that have been stated in Section 3.2 below:

- the application content (text, images) is kept short
- the content for each screen is designed to fit the size of the mobile phone device
- the content is structured hierarchically and the visitors are provided with visual feedback for their selection
- backtracking is supported
- adaptation capabilities are limited to the user choice and the interface configuration according to the mobile phone device.

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 available free of charge)
- lightweight storage and memory footprint for both the run-time environment and the application

- potential for developing any type of content and powerful services not restricted by the functionality of an authoring tool.

-

```
<?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 >
     ……..
</city>
```

Figure 8. An excerpt of XML code describing the users' accommodation preferences.


Figure 9. Screens of a mobile phone emulator executing the myMytileneCity guide application.

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 [22] (allows to build applications that adapt according to the user's physical location), the Mobile 3D Graphics API [23] (use of lightweight 3D graphics), etc.

The following section identifies a set of requirements for cultural application authoring tools and development platforms.

## 6. Requirements for cultural application authoring tools and development platforms

It is evident, that the development tools reviewed in Section 4 have different features and devices target groups. The scope of this section is to present the requirements for development tools tailored to design and development of multimedia applications for small devices. These requirements are listed below:

- acceleration of application's development and deployment
- reduction of development effort and technical knowledge (e.g. programming skills) required by designers; familiarity of designers with the tool's workspace
- provision of tools for designers and developers that allow a new level of expressiveness, efficiency and interactivity for multimedia content creation and intuitive UI design, personalized according to the user profile (such design could exceed customer expectations and optimize content delivery)
- support for a broad range of mobile devices (ideally, support for PDAs, smart phones and mobile phones)
- restriction on the resource overhead posed by the run-time environment (supporting libraries, APIs, etc.)
- seamless connectivity of applications to services with minimal programming effort
- platform independence of applications from underlying devices hardware and operating systems
- capability for parsing and handling any content type format
- potential for developing entirely new content and services that overcome the restriction set by rigidly de-fined content templates
- capability for dynamic customization and over-the-air update of existing applications content and functionality
- increased deployment base of tools' runtime environments, i.e. management software and media players installed by the major device manufacturers
- minimization of cost for both the designer tools and the runtime environments
- support for location-based services, i.e. availability of resources and services depending on the end user's physical location
- support for 'push model', namely for pushing content to mobile terminals with minimal user intervention the moment an important event occurs
- support for disconnected operation, i.e. ability to run applications in standalone mode even when the mo-bile terminal is out of any network's coverage area
- need for large development community base, which may assist the exchange of development experiences (e.g. through developer forums)

- availability of add-on application libraries, which may accelerate the implementation of custom services.

Table 1 below summarizes the features of the available development tools (Flash Lite, Navipocket and J2ME) within respect to the above listed set of requirements.

| | **Flash Lite** | **Navipocket** | **J2ME** |
|---|---|---|---|
| **Development and deployment speed** | Relatively fast | Very fast | Slow |
| **Technical knowledge required** | Flash developers can instinctively adapt. Users with no prior knowledge require a lengthy training period. | Effortless (2 hours maximum required to understand the UI for an experienced web designer) | Advanced Java programming skills are required |
| **Content development and UI design tools** | Very advanced | Restrictive | Not integrated, requires additional design automation tools, e.g. J2ME Polish [18] |
| **Targeted mobile devices** | PDAs, smart phones | PDAs | PDAs, smart phones, mobile phones |
| **Run-time environment resource overhead** | ~ 6 MB | ~ 1 MB | Up to 100 KB for storage (CLDC/MIDP and kXML), total memory footprint of approximately 128 KB |
| **Applications connectivity** | Feasible, requires programming effort | Not enabled, customized according to customer requirements | Feasible (through HTTP), requires programming effort |
| **Platform independence** | Mobile devices with Flash Lite or Flash Player SDK technology | Requires Windows Pocket PC; executed on PDA platforms[c] | Execution on any device supporting CLDC/MIDP |
| **Accessible content format** | Handles proprietary file formats in addition to either 'external' or integrated multimedia file formats | Handles proprietary file formats in addition to bitmaps and mpeg files | Any (text, XML, WML, cHTML, HTML, XHTML, serialized objects, etc.), but requires specialized parsers (e.g. kXML parser [26] for analyzing XML content) |
| **Potential for developing entirely new content and services** | Development restricted by Flash Lite authoring environment | Development restricted by Navipocket Creator's functionality | Capacity to develop rich content and new powerful applications, inherited by Java programming language |
| **Support for dynamic** | Applications may | Not supported | Applications may |

[c] The release of the latest version of Navipocket Simulator has recently been announced by OPHRYS, planned for July 2006. The new release will only support a PDA manufactured by OPHRYS.

| | | | |
|---|---|---|---|
| **application update** | synchronize with the backend infrastructure to dynamically update content | | synchronize with the backend infrastructure to dynamically update content |
| **Run-time environment's deployment base** | Most major manufacturers | Not supported | Very large deployment base (virtually all modern mobile devices) |
| **Cost** | ~€10 for Flash Player, ~€700 for Flash Professional and ~€4800 for streaming support | Free licence given by OPHRYS; free licence also for the new Navipocket release, however purchasing the OPHRYS PDA is required | Free |
| **Support for location-based services** | Not supported | Not supported | Yes (precise location identification though the optional 'Location API' [22]) |
| **Support for 'push model'** | Not supported | Not supported | Yes (in MIDP 2.0) |
| **Support for disconnected operation** | Yes | Yes | Yes |
| **Developer community base** | A starting community | Very limited | Large community of developers |
| **Developer libraries** | Not many | Only custom libraries for specific customer needs | A large scale of libraries for developers to choose from |

Table 1. Features of the development tools (Flash Lite, Navipocket and J2ME)

The above synopsis shows that the choice of the appropriate development technology is not a straightforward task, since the three reviewed technologies vary significantly in terms of their merits and weaknesses. In particular, the selection of a candidate development technology should depend on user and application needs, such as:

- the technology literacy of developers and familiarity with relevant multimedia based application environments
- the urgency of project completion
- the application requirements regarding network connectivity, dynamic updates, supported services
- the targeted devices
- the project's budget

In conclusion, in order to satisfy application and designer needs for developing operational and profitable cultural and tourist applications, future releases of application authoring tools and development platforms should be directed in combining the strengths of the existing technologies.

## 7. Conclusions

This paper reviews state-of-the-art technologies for developing mobile applications that enhance the visiting experience in cultural organizations and support tourists travelling experience. Based on this review it suggests application authoring tools requirements for developing cultural applications on PDAs and mobile phones based on user, application and designer needs.

To support the requirements gathering process two case studies have been developed, that focused on the use of PDAs and mobile phones for providing cultural and tourist information and promoting cultural content and tourist facilities. These case studies were based on the use of the Navipocket authoring tool and J2ME application development platform, respectively. The prototypes implementation contributed to the evaluation of the main assets and shortcomings of such development technologies.

Specifically it has been concluded that the choice of the appropriate -among Flash Lite, Navipocket and J2ME- technology depends on factors like the developers technical background, the application requirements, the targeted devices and the project's timeline and budget.

Future development of application authoring tools and development platforms should bring together the strengths of existing multimedia technologies in order to satisfy application and designer requirements for developing effective and beneficial cultural and tourist applications.

**Acknowledgements**

The authors would like to thank OPHRYS for the provision of Navipocket software for the implementation of the "Fables" prototype.

**References**

1. Adobe, Macromedia Flash Lite, http://www.adobe.com/products/flashlite/
2. Amirian, S. Hand-held Mobile Computing in Museums, 2001.
3. Anegg, H., Kunczier, H., Michmayr, E., Pospischil, G. and Umlauft, M. LoL@: Designing a Location Based UMTS Application. Elecktrotechnik und Informationstechnik, 191(2). pp. 48-51, 2002.
4. ASP.NET mobile controls, http://www.asp.net/mobile/.
5. Baus, J., Kray, C. and Krueger, A. Visualization of Route Descriptions in as Resource-Adaptive Navigation Aid. Cognitive Processing, 2 (2-3). pp. 323-245, 2001.
6. Brewster, S. Overcoming the lack of screen space on mobile computers. Personal and Ubiquitous Computing, 6 (3). pp. 188 -205, 2002.
7. Brown, B. and Chalmers M. Tourism and Mobile Technology. In Proceedings of the 8th European Conference on Computer Supported Cooperative Work (CSCW'03), 2003.
8. Canadian Heritage Information Network (CHIN): Tip sheets, personal digital assistants (PDA), PDA aesthetics and interface design, Creating and Managing Digital Content, 2004. Last visited in 10/04/2006: http://www.chin.gc.ca/English/Digital_Content/index.html
9. Checkland, P. and Scholes, J. Soft Systems Methodology in Action. Chichester-NY: John Wiley & Sons, 1990.
10. Cheverst, K., Davies, N., Mitchell, K., Friday, A. and Efstratiou, C. 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, 2000.

11. Ciavarella, C. and Paterno, F. Design criteria for location-aware, indoor, PDA applications. In Proceedings of Human-Computer Interaction with Mobile Devices and Services, 5th International Symposium Mobile (HCI 2003), Luca Chittaro (ed.). pp. 131-144, 2003.
12. Dublin Tourism on the Wireless Web. Last visited in 9/3/2006: http://www.visitdublin.com/wireless/.
13. Dunlop, M. and Brewster, S. The challenge of mobile devices for human computer interaction. Personal and Ubiquitous Computing, 6 (4). pp. 235 -236, 2002.
14. Evans, J. and Sterry, P. Portable computers & interactive media: A new paradigm for interpreting museum collections. In Cultural Heritage Informatics 1999: Selected papers from ICHIM 99, D. Bearman and J. Trant (eds.). Kluwer Academic Publishers. pp. 93-101, 1999.
15. Gupta, A. and Srivastava, M. Integrated Java Technology for End-to-End m-Commerce. Last visited in 5/2/2006: http://developers.sun.com/techtopics/mobility/midp/articles/mcommerce/.
16. Hochstatter, I., Küpper, A., Schiffers, M. and Köthner, L. Context Provisioning in Cellural Networks. In Proceedings of the 8th IEEE International Workshop on Mobile Multimedia Communications, 2003.
17. Holtsblatt, K. Designing for mobile devices: Experiences, Challenges and Methods. In Communications of the ACM, 48(7). pp. 33-35, 2005.
18. J2ME Polish. Last visit: 15/3/2006, http://www.j2mepolish.org/.
19. Java 2 Platform Micro Edition (J2ME). Last visited in 30/11/2005: http://java.sun.com/j2me/.
20. Java API for XML Processing (JAXP). Last visited in 10/3/2006: http://java.sun.com/webservices/jaxp/index.jsp.
21. JavaServer Pages Technology. Last visited in 10/3/2006: http://java.sun.com/products/jsp/.
22. JSR 179: Location API for J2ME. Last visited in 10/4/2006: http://jcp.org/en/jsr/detail?id=179.
23. JSR 184: Mobile 3D Graphics API for J2ME. Last visited in 10/3/2006: http://www.jcp.org/en/jsr/detail?id=184
24. Kray, C. and Baus, J.: A survey of mobile guides. In Proceedings of HCI in Mobile Guides. In conjunc-tion with Fifth International Symposium on Human computer Interaction with Mobile Devices and Ser-vices. pp. 1-5, 2003.
25. Kray, C., Laakso, K., Elting, C. and Coors, V. Presenting Route Instructions on Mobile Devices. In Pro-ceedings of the 2003 International Conference on Intelligent User Interfaces, IUI'03. pp. 117-124, 2003.
26. kXML parser. Last visited in 10/3/2006: http://kxmlrpc.objectweb.org/.
27. Lehn, D., and Heath, C. Displacing the object: Mobile technologies and interpretive resources. In Proceedings of the ICHIM 03. Archives and Museum Informatics Europe, 2003. Last visited in 10/04/2006: http://www.ichim.org/ichim03/PDF/088C.pdf.
28. Micha, K. and Economou, D. Using Personal Digital Assistants (PDAs) and to enhance the museum visit experience. In Proceedings of the 10th Panhellenic Conference on Informatics, LNCS 3746. pp. 188-198, 2005.
29. Norman, D. The Design of Everyday Things. New York: Basic Books, 1998.
30. NTT DoCoMo. Last visited in 8/3/2006: http://www.nttdocomo.com/.
31. Oppermann, R. and Specht, M. A Context-Sensitve Nomadic Exhibition Guide. In Proceedings of the 2nd Symposium on Handheld and Ubiquitous Computing, (HUC2K'2000). pp. 127-149, 2000.
32. Preece, J., Rogers, Y. and Sharp, H. Interaction design: Beyond human – computer interaction. Danvers, MA: Wiley, 2002.
33. Proctor, N. and Tellis, C. The state of the art in museum handhelds in 2003. In Proceedings of the Inter-national Conference of Museums and the Web 2003. D. Bearman and J. Trant, (eds.). Archives & Museum Informatics, 2003. Last visited in 10/04/2006: http://www.archimuse.com/mw2003/papers/proctor/proctor.html.
34. Sparacino, F., Larson, K., MacNeil, R., Davenport, G. and Pentland, A. Technologies and methods for interactive exhibit design: from wireless object & body tracking to wearable

computers. In Proceedings of International Conference on Hypertext and Interactive Museums, ICHIM'99, Washington, DC. Sept. 22-26, 1999. Last visited in 10/04/2006: http://ic.media.mit.edu/Publications/Conferences/MuseumTracking/PDF/MuseumTracking.pdf.

35. Tenmoku, R., Kanbara, M. and Yokoya N. A Wearable Augmented Reality System Using Positioning Infrastructures and a Pedometer. In IEEE Proceedings of International Symposium on Wearable Computers 2003. pp. 110-117, 2003.
36. Varshney, U. Issues, Requirements and Support for Location-Intensive Mobile Commerce Applications. International Journal of Mobile Communications, 1(3). pp 247-263, 2003.
37. Vlahakis V., Ioannidis N. and Karigiannis J. ARCHEOGUIDE: Challenges and Solutions of a Personal-ised Augmented Reality Guide for Archaeological sites. In Computer Graphics in Art, History and Archaeology, Special Issue of the IEEE Computer Graphics and Applications Magazine, 22(5). pp. 52-60, 2002.
38. Wilson, G. Multimedia Tour Programme at Tate Modern. In Proceedings of the International Conference of Museums and the Web 2004. D. Bearman and J. Trant (eds.). Archives & Museum Informatics, 2004. Last visited in 10/04/2006: http://www.archimuse.com/mw2004/papers/wilson/wilson.html
39. Wireless Application Protocol (WAP) Forum. Last visited in 8/3/2006: http://www.wapforum.org/
40. Woodruff, A., Aoki, P.M., Hurst, A. and Szymanski., M.H. Electronic guidebooks and visitor attention. In the 6th International Cultural Heritage Informatics Meeting Proceedings. pp. 437-454, 2004. Last visited in 10/04/2006: http://www2.parc.com/csl/members/woodruff/publications/2001-Woodruff-ICHIM2001-VisitorAttention.pdf