Authoring tools & development platforms: Requirements for mobile devices-enabled cultural applications

Daphne Economou, Damianos Gavalas, Michael Kenteris

Cultural Heritage Management Laboratory (CHMLab)
Department of Cultural Technology and Communication
Harilaou Trikoupi & Faonos St., GR – 81100
University of the Aegean, Mytilene, Lesvos, Greece
d.economou@ct.aegean.gr, dgavalas@aegean.gr, ctm04007@ct.aegean.gr

Abstract

This paper identifies authoring tools requirements for the development of cultural applications tailored for deployment on mobile devices: Personal Digital Assistants (PDAs) and mobile phones. To address this issue it recognizes 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, Java 2 Micro Edition and Microsoft .Net platform for the Mobile Web. Secondly, it describes the use of these tools for the implementation of three projects that have been developed at the Department of Cultural Technology and Communication, University of the Aegean, in Greece, providing cultural and tourist information. Based on these three case studies the paper extracts a set of PDA and mobile phone-enabled application requirements and concludes with a set of suggestions related to the way authoring tools should be exploited in order to gratify application and designer needs for developing functional, fast & easy to deploy and profitable cultural applications.

Keywords: Multimedia authoring tools, development platforms, mobile cultural applications, requirements gathering, Flash Lite, Navipocket, J2ME, .NET Mobile Platform.

1. Introduction

Mobile devices have gained increasing acceptance as platforms for executing cultural multimedia applications due to their physical characteristics and suitability in these fields. However, currently available tools for developing multimedia applications for mobile devices are light versions of state-of-the-art multimedia authoring tools, which are not tailored to satisfying 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 based on user, application and designer needs (in section 5). This exercise builds upon the evaluation of current state-of-the-art authoring tools and development platforms (in section 3) and on experience gained by using such tools in three case studies developed in our laboratory (in section 3.2.2). The first case study involves a museum guide and news reader application executing on mobile phones (in section 4.1). The second focuses on the use of PDAs for the provision of interpretative cultural information in a museum environment (in section 4.2), whilst the third is a mobile tourist guide research prototype (in section 4.3). The case studies have been developed using Flash Lite and Navipocket authoring tools and J2ME development platform, respectively. The paper closes with conclusions about this work (in section 6). A table that summarizes the main features of the four reviewed technologies may be found at the end of the paper and is expected to be rather useful resource for mobile application designers and developers that are open to mobile software development framework.

2. Cultural Mobile Applications

Multimedia and the Internet provide unique opportunities to cultural organizations (Museums, Libraries and Archives, visitor centers, exhibition centers) as they bring new ways of communication and interpretation. Commonly used technological solutions in the context of cultural organizations, like projection systems and info-kiosks, successfully connect cultural artifacts to related information. However, these solutions are tightly bound to a cultural organization's physical space. Mobile technologies allow the dynamic presentation of multimedia information without being limited by a physical environment. In addition, current mobile devices allow accessing Internet resources. The WWW represents a medium which is well tried and tested on cultural organizations related to information and services provision. 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.

Due to current practical concerns mobile technologies have not been incorporated by most of cultural organizations for information and services provision. However, it is believed that they provide a promising media for enhancing the cultural experience. To address this issue multimedia application development tools are required to satisfy user, application and designer needs.

The following section presents state-of-the-art tools for multimedia application development and services provision for PDAs and mobile phones.

3. Mobile Application Development Tools

Typical tools for multimedia application development and services provision for PDAs and mobile phones are classified in:

- authoring tools
- application development platforms.

3.1 Authoring tools

3.1.1 Macromedia Flash Lite

Macromedia Flash Lite [Adobe Flash Lite] is referred to in this section, although not used in the case studies examined later on in the paper, as one of the most commonly used multimedia authoring tools that enables companies to easily and rapidly deploy content to mobile devices. The explosive adoption of Flash Lite was driven by a variety of causes. The Flash Lite authoring environment 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. In addition, developers already skilled in working with Flash MX can easily switch into using Flash Lite to design applications for mobile devices.

3.1.2 Navipocket

NaviPocket v. 2.4 by OPHRYS SYSTEMS [OPHRYS Navipocket] has been designed to meet the demands of Theme and Leisure Parks, Museums and Cultural sectors in developing multimedia guides. Navipocket 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 CE 2.xx and PocketPC). The current version works with Microsoft PocketPC^{TM} 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”. 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. Navipocket supports the following objects: page, button, text area, bitmap and video. Each object has properties and can be linked with another object.

Authoring tools like Navipocket and Flash Lite accelerate the delivery of advanced applications and content services. However, they are not open source, they do not support dynamic content maintenance and they require MS Windows compatible devices for the development of multimedia projects, and for the run-time.

3.2 Development platforms

3.2.1 J2ME

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

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

On the other hand, MIDlets programming is not straightforward as it requires Java development skills. The development of J2ME applications is far more complex compared to creating content using developer-friendly authoring tools like Flash Lite or Navipocket.

3.2.2 Microsoft .Net platform for the Mobile Web and the ASP.NET mobile controls

Microsoft's entering the mobile market has been characterized by the release of a proprietary operating system, namely, Microsoft Windows Mobile 2003 [Microsoft Windows Mobile (2006)] and the provision of developer support to program mobile devices. Specifically, a subset of the rich .NET Framework, called Microsoft .NET Compact Framework, provides a runtime engine preloaded in the device's memory in order to facilitate mobile application deployment.

The ASP.NET mobile controls [ASP.NET Mobile Controls, 2006] (formerly known as the Microsoft Mobile Internet Toolkit, MMIT) represent a mobile application development platform, recently released by Microsoft. 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) development environment. The main asset of MIT is that it provides server-side mobile controls (including user interface elements such as list, command, call, calendar, etc.) with rich device identification mechanisms; developers simply utilize ASP.NET pages (for no particular target device) which automatically identify the device that posted a request1 and render the appropriate content.

Summarizing, the main strengths of ASP.NET mobile controls are: no need to perform browser checks and deliver the appropriate content based on the target device

[1] 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.

(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. Case Studies

In this section three case studies are presented. The first involves a Flash Lite-based application executing on mobile phones, the second evaluates Navipocket as an authoring tool for the development of a cultural multimedia application on a PDA and the last evaluates J2ME as a development platform for the implementation of a tourist guide on a mobile phone.

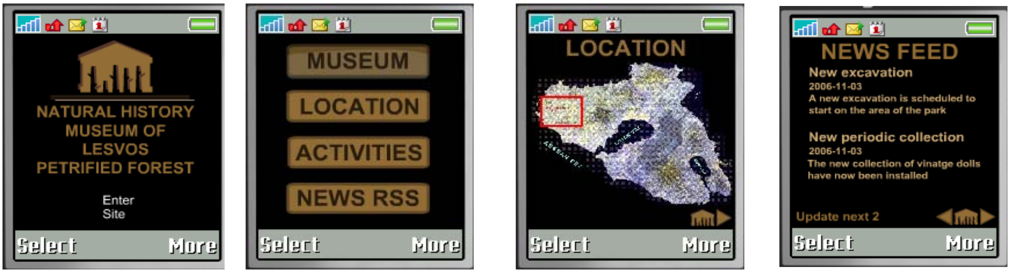
4.1 The mobile phone used as museum guide and news reader

Presented here is a prototype mobile application used to promote museum services via a mobile phone. As a case study we have chosen the natural history museum of the Petrified forest situated in Lesvos, Greece. The original concept of this project was to build a lightweight, robust application to promote the museum including its location, descriptive content of its collections and a latest news section to mobile devices. The case study originated via a promotional campaign of the museum; the museum's executives seek for distributing multimedia-enabled informative content via their internet web site targeting mobile devices. Flash Lite was chosen as the development platform, mainly because the developers were already skilled in working with Flash MX Professional and therefore could easily switch to Flash Lite to design applications for mobile devices, thereby enhancing dramatically the production curve.

At this stage, the museum case study does not incorporate any location-based services; this does not though represent a critical omission since this application targets mass deployment for promotional use away from the museum and not for use as a guide system during a museum visit.

Several usability aspects have been taken into account in the design of user interfaces, e.g. (a) no scroll bars are used since they were found difficult in use for small screen sizes; (b) menu buttons are selected via navigational phone buttons and not via users softkeys; (c) the main functions buttons incorporate the same design for all navigational functions. Figure 1 illustrates several screenshots of the application developed in Flash Lite.

The application includes a news reader facility, whereby the news section is periodically updated using HTTP to the news feeder of the museum's website. The size of the application is directly related to the content size. The final application size is 454 KB when incorporating images and a short video and 256 KB when omitting the clip gallery and the video.



New excavation
2006-11-03
A new excavation is scheduled to start on the area of the park

New periodic collection
2006-11-03
The new collection of vinatge dolls have now been installed

Figure 1. *Screenshots of the museum guide application taken from a mobile phone emulator.*

The prototype's development phase highlighted several assets of Flash Lite tool, such as the acceleration of development and deployment speed (especially for developers familiarized with Flash MX Professional), the advanced content development and UI design tools and the satisfactory support for every class of mobile devices. On the other hand, Flash Lite poses a heavyweight run-time environment resource overhead, it includes restrictive facilities for developing entirely new content and services and has limited networking support.

4.2 Fables on pocket PC

The "Fables" prototype for the Museum/Library Stratis Eleftheriadis Teriade in Lesvos, Greece, is the first attempt in Greece of using PDAs to aid the museum visit [Micha & Economou (2005)]. This project uses PDAs to provide enriched multimedia interpretative information for the collection of "Fables" by Jean de La Fontaine (see Figure 2), which have been illustrated by Marc Chagall and are exhibited in part of the Teriade Museum.

The development process of the "Fables" prototype demonstrated that Navipocket offers a number of advantages. Navipocket is available free of charge, its runtime environment requires low storage and memory overhead, it offers rapid multimedia content and UI creation and it requires effortless familiarization of developers with the authoring environment.

However, several drawbacks of Navipocket have also been revealed. Navipocket depends on specific operating systems and devices hardware's, the tools for the design of expressive UI it provides are restrictive, the multimedia formats supported are limited which results to large-sized applications, dynamic content update is not supported, and it lacks of specialized libraries for implementing extra functionality (custom solutions for specific customers could be supported by OPHRYS on demand).

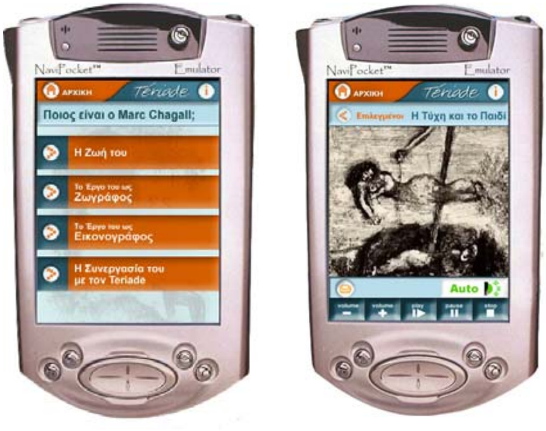


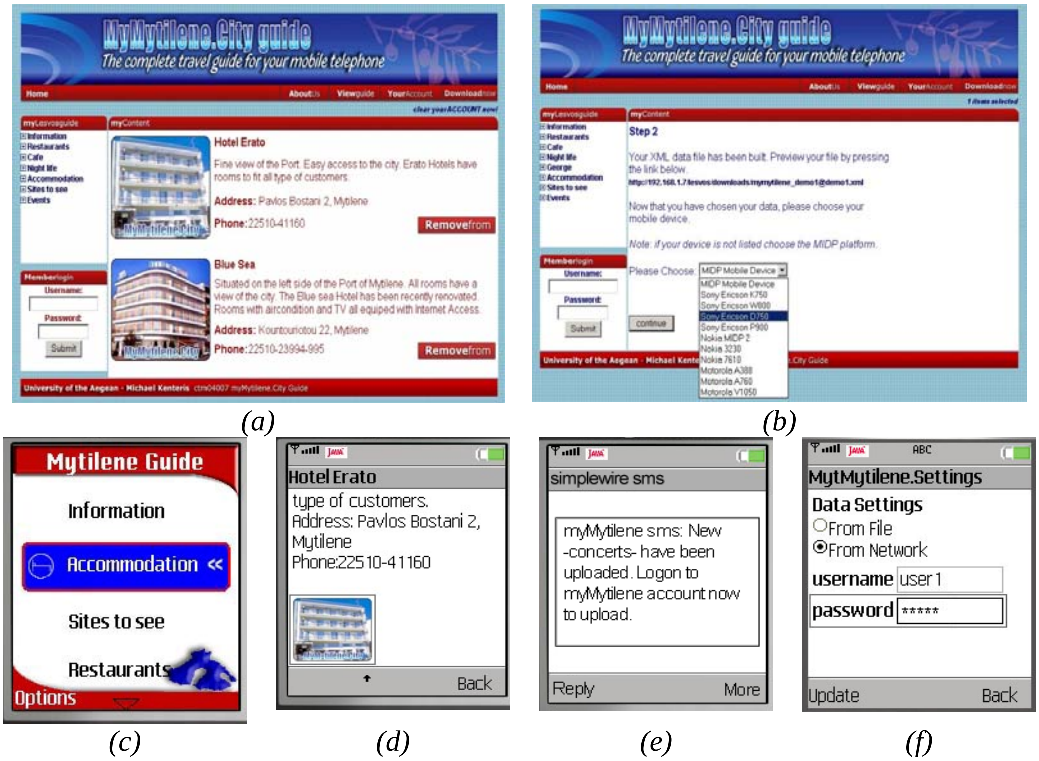
Figure 2. *The layout of the “Fables” prototype application*

4.3 myMytileneCity: A mobile tourist guide

myMytileneCity [Kenteris et Al. (2006)] is an electronic guide implementation for the city of Mytilene, Greece. This research prototype first enables (through a dynamic web site) the creation of mobile tourist applications with rich content that matches user preferences. The users may then download these customizable applications either directly to their mobile device or first to a PC and then to a mobile terminal (e.g. through bluetooth). Thereafter, network coverage is not further required as the applications execute in standalone mode and may be updated when the user returns online. Our prototype has been developed on the top of J2ME. Representative screens of the myMytileneCity web site and mobile application are shown in Figure 3.

The prototype development revealed that J2ME offers advantages like: low application development cost (practically free of charge); lightweight storage and memory footprint for both the runtime environment and the application; potential for developing any type of content and powerful services not restricted by the functionality of an authoring tool.

On the other hand, several inherent weaknesses of J2ME have also been brought into the spotlight: although experienced in Java programming, our 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 (e.g location-awareness) require the usage of specialized (optional) APIs, which certainly increase the overall applications’ overhead.



Hotel Erato

Fine view of the Port. Easy access to the city. Erato Hotels have rooms to fit all type of customers.

Address: Pavlos Bostani 2, Mytilene

Phone:22510-41160

Blue Sea

Situated on the left side of the Port of Mytilene. All rooms have a view of the city. The Blue sea Hotel has been recently renovated. Rooms with aircondition and TV all equiped with Internet Access.

Address: Kountouriotou 22, Mytilene

Phone:22510-23994-995

University of the Aegean - Michael Kenteris ctm04007 myMytilene.City Guide

(a)

Step 2

Your XML data file has been built. Preview your file by pressing the link below.

http://192.168.1.7/lesvos/downloads/mymytilene_demo1@demo1.xml

Now that you have chosen your data, please choose your mobile device.

Note: if your device is not listed choose the MIDP platform.

Please Choose: MIDP Mobile Device

MIDP Mobile Device
Sony Ericson K750
Sony Ericson W800
Sony Ericson D750
Sony Ericson P900
Nokia MIDP 2
Nokia 3230
Nokia 7610
Motorola A388
Motorola A760
Motorola V1050

University of the Aegean - Michael Kente… City Guide

(b)

(c)

Hotel Erato

type of customers.
Address: Pavlos Bostani 2,
Mytilene
Phone:22510-41160

(d)

simplewire sms

myMytilene sms: New -concerts- have been uploaded. Logon to myMytilene account now to upload.

(e)

MytMytilene.Settings

Data Settings
○ From File
● From Network

username user1
password *****

(f)

Figure 3. *(a, b) Screenshots from the myMytileneCity web site (selected user's content items and selection of mobile device's profile when 'checking out'), (c, d, e, f) Representative screens of an emulator executing the myMytileneCity guide mobile application.*

5. Authoring Tools and Development Platforms Requirements

It is evident, that the development tools reviewed in Section 3 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 defined 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 mobile 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 at the end of this paper summarizes the features of the available development tools (Flash Lite, Navipocket, J2ME and .NET Mobile Platform) within respect to the above listed set of requirements.

The synopsis of Table 1 proves that the choice of the appropriate development technology is not a straightforward task, since the four 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:

Table 1. *Features of mobile applications authoring tools (Flash Lite, Navipocket) and development platforms (J2ME and .NET Mobile).*

	Flash Lite	Navipocket	J2ME	.Net platform for the Mobile Web
Development and deployment speed	Relatively fast	Very fast	Slow	Relatively fast
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	ASP.NET and .NET mobile controls programming skills are required, yet, no markup language authoring skills are needed
Content development and UI design tools	Very advanced	Restrictive	Not integrated, requires additional design automation tools, e.g. J2ME Polish [J2ME Polish]	Very advanced, through the ASP.NET mobile controls
Targeted mobile devices	PDAs, smart phones, mobile phones	PDAs	PDAs, smart phones, mobile phones	Pocket PCs, PDAs, smart 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	4,4 MB footprint for the .NET Compact Framework
Applications connectivity	Feasible, requires programming effort	Not enabled, customized according to customer requirements	Feasible (through HTTP), requires programming effort	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[2]	Execution on any device supporting CLDC/MIDP	Targets devices with Microsoft operating systems
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 [kXML parser] for analyzing XML content)	The appropriate content format (HTML, WML, cHTML) is generated depending on the target device (Microsoft should first announce and incorporate support for specific markup languages versions within .NET mobile controls)

[2] 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.

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	Capacity to develop rich content and new powerful applications, inherited by .NET framework
Support for dynamic application update	Applications may synchronize with the backend infrastructure to dynamically update content	Not supported	Applications may synchronize with the backend infrastructure to dynamically update content	Applications may 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)	Not supported
Cost	~€10 for Flash Player, ~€700 for Flash Professional and ~€4800 for streaming support	Free license given by OPHRYS; free license also for the new Navipocket release, however purchasing the OPHRYS PDA is required	Free	Free license for the .NET Compact Framework, ~€700 for Visual Studio 2005 Professional
Support for location-based services	Not supported	Not supported	Yes (precise location identification though the optional 'Location API' [JSR 179])	No inherent support, approximate location identification through mobile operators networks, increased number of SDKs for developing location-based services (e.g. Microsoft's MapPoint [Microsoft Map Point Web service])
Support for 'push model'	Not supported (possible only through WAP)	Not supported	Yes (in MIDP 2.0)	Not supported
Support for disconnected operation	Yes	Yes	Yes	Yes
Developer community base	A starting community	Very limited	Large community of developers	Large community of developers
Developer libraries	Not many	Only custom libraries for specific customer needs	Large scale of libraries for developers to choose from	Large scale of libraries for developers to choose from

- 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.

6. Conclusions

This paper reviews state-of-the-art technologies for developing mobile applications that enhance the visiting experience in cultural organizations and support tourists traveling experience. Based on this review it suggests authoring tools requirements for developing cultural applications on PDAs and mobile phones based on user, application and designer needs. The requirements gathering process is based on three case studies, that focused on the use of PDAs and mobile phones for providing cultural and tourist information. These case studies were based on the use of Flash Lite and 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.

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

7. References

Adobe Flash Lite, last visit: 8/12/2006, http://www.adobe.com/products/flashlite/.

ASP.NET Mobile Controls (2006), last visit: 30/11/2006, http://www.asp.net/mobile/.

Java 2 Platform Micro Edition (J2ME), last visit: 30/11/2006, http://java.sun.com/j2me/.

J2ME Polish. Last visit: 15/11/2006, http://www.j2mepolish.org/.

JSR 179: Location API for J2ME, last visited in 10/7/2006: http://jcp.org/en/jsr/detail?id=179.

Kenteris M., Gavalas D., Economou D. (2006) *Novel Method for the Development of Personalized Mobile Tourist Applications,* in Proc. CSN’2006: the 5th IASTED International Conference on Communication Systems and Networks, pp. 208-212.

kXML parser, last visited in 10/9/2006: http://kxmlrpc.objectweb.org/.

Micha K., Economou D. (2005) *Using Personal Digital Assistants (PDAs) and to enhance the museum visit experience*, in Proc. PCI-05: the 10th Panhellenic Conference on Informatics, LNCS 3746, pp. 188-198.

Microsoft Map Point Web service, last visit: 30/11/2006, http://msdn.microsoft.com/mappoint/.

Microsoft Windows Mobile (2006), last visit: 30/9/2006 http://www.microsoft.com/windowsmobile/.

OPHRYS Navipocket, last visited in 10/7/2006, http://www.ophrys.net/audioguide fran%E7ais/navipocket.html.