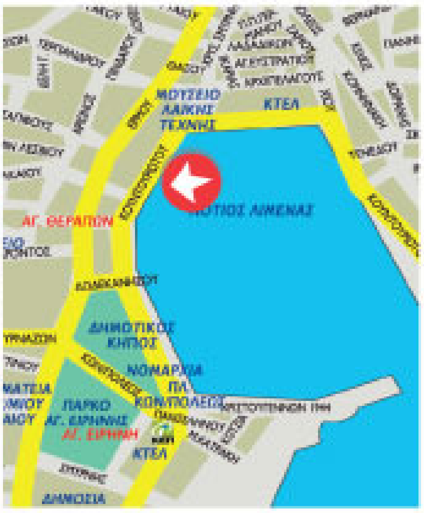
UNIVERSITY
OF THE AEGEAN

Department of Cultural
Technology & Communication

October 2010

Michael Kenteris

Internet Multimedia Applications for Mobile Devices: The Case of Electronic Tourist Guides



A thesis submitted as a partial fulfilment for
the award of the degree of Doctor of Philosophy
Department of Cultural Technology & Communication
University of Aegean

PHD Thesis

Internet Multimedia Applications for Mobile Devices:

The Case of Electronic Tourist Guides

Michael Kenteris

Advisory committee

Damianos Gavalas, Assistant Professor, Department of Cultural Technology and Communication, University of the Aegean (Advisor)

Βαγγελιώ Καβακλή, Assistant Professor, Department of Cultural Technology and Communication, University of the Aegean (Member of the coadvisory 3 member commitee)

George Tsekouras, Assistant Professor, Department of Cultural Technology and Communication, University of the Aegean (Member of the coadvisory 3 member commitee)

John D. Garofalakis, Associate Professor, Department of Computer Engineering and Informatics, University of Patras

Charalampos Konstantopoulos, Lecturer, Department of Informatics, University of Piraeus

Vasilios A. Siris, Assistant Professor, Department of Informatics, Athens University of Economics and Business

George M. Giaglis, Assistant Professor, Department of Management Science and Technology, Athens University of Economics and Business

Διδακτορική Διατριβή

Διαδικτυακές Πολυμεσικές Εφαρμογές Για Κινητές Συσκευές:

Η Περίπτωση Των Ηλεκτρονικών Τουριστικών Ξεναγών

Μιχάλης Κεντέρης

Επταμελής Εξεταστική Επιτροπή

Δαμιανός Γαβαλάς, Επίκουρος Καθηγητής, Τμήμα Πολιτισμικής Τεχνολογίας & Επικοινωνίας Πολιτισμικής Τεχνολογίας & Επικοινωνίας, Πανεπιστήμιο Αιγαίου (επιβλέπων)

Βαγγελιώ Καβακλή, Επίκουρη Καθηγήτρια, Τμήμα Πολιτισμικής Τεχνολογίας & Επικοινωνίας, Πανεπιστήμιο Αιγαίου (μέλος 3μελούς συμβουλευτικής επιτροπής)

Γιώργος Τσεκούρας , Επίκουρος Καθηγητής, Τμήμα Πολιτισμικής Τεχνολογίας & Επικοινωνίας, Πανεπιστήμιο Αιγαίου (μέλος 3μελούς συμβουλευτικής επιτροπής)

Ιωάννης Γαροφαλάκης, Αναπληρωτής Καθηγητής, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Πατρών

Χαράλαμπος Κωνσταντόπουλος, Λέκτορας, Τμήμα Πληροφορικής, Πανεπιστήμιο Πειραιά

Βασίλειος Σύρης, Επίκουρος Καθηγητής, Τμήμα Πληροφορικής, Οικονομικό Πανεπιστήμιο Αθηνών

Γεώργιος Γιαγλής, Αναπληρωτής Καθηγητής, Τμήμα Διοικητικής Επιστήμης, Οικονομικό Πανεπιστήμιο Αθηνών

ABSTRACT

Mobile Tourist Guides (MTGs) have attracted considerable research interest during the past decade resulting in numerous standalone and web-based mobile applications. MTGs enhance the tourism experience of users, even more so by incorporating features like interactive maps and location-based services. However, there are issues that still hinder the wide use and market success of such technologies. Not only are there usability issues when designing for mobile devices, like device capabilities, screen size and input methods; one must also take into account the compatibility issues among web and mobile web platforms and the cost of use of such technologies for tourists. Furthermore, the mobile phone is still evolving which brings about problems in the making of standards. This calls for solutions which can compensate for the constraints of mobile browser capabilities and the lack of device standards compliance.

One of the first objectives of the research work described in this thesis has been to record and evaluate the tools which can be used to create full-fledged mobile multimedia applications and evaluate research and commercial mobile applications used by tourists for information provision, navigation, guidance or entertainment. Through this evaluation we have derived a set of design guidelines for a tourist platform which can be used by tourists on and off the web in either a static or mobile environment.

Using those guidelines as stepping stones, we have developed a research prototype, a multi-platform tourist framework that uses the web to promote cultural information to mobile tourists. This framework allows for tourists to tag content of personal interest; following this, the users are prompted to build a personalized mobile application which can run on their mobile phone either in standalone mode or using the web to get access to additional services.

Among other things, particular emphasis has been given to the personalisation of services through designing and implementing an innovative recommendation system used to assist tourists in choosing places to visit. Unlike existing systems, our "recommender" system exploits specific information, behaviours, ideas, evaluations, assessments, ratings, and so on of other tourists with similar interests, which then provides ground for the cooperative production of tourist content and travel recommendations. In addition, we extend this notion of travel "recommender" systems utilizing context-aware collaborative filtering techniques for deriving improved recommendations.

We also propose the use of Wireless Sensor Network (WSN) installations around tourist sites with the aim of providing mobile users convenient and inexpensive means for uploading tourist information and ratings about Points of Interest (POIs) via their mobile devices. User ratings uploaded through WSN infrastructures are weighted higher in order to differentiate between users that rate POIs using the mobile tourist guide application in direct proximity of the POI and others using the web, away from the POI.

A final contribution this doctoral research attempts to fulfil deals with the problem of deriving personalized recommendations for daily sightseeing itineraries for tourists visiting any destination. Our approach considers selected POIs that a traveller would potentially wish to visit and derives a near-optimal itinerary for each day of their visit; the places of potential interest are selected based on stated or implied user preferences. Our method enables the planning of customized daily personalised tourist itineraries considering user preferences, time availability for visiting sights in daily basis, opening days of sights and average visiting times for these sights.

ACKNOWLEDGEMENTS

Given the limited space of this thesis there is not enough room to acknowledge my feelings for those who have supported me on this quest for knowledge. Therefore, in short, I would like to express my deepest gratitude to my supervisor Dr. Damianos Gavalas; his support made it possible for me to complete this thesis, build a credible bibliography and gain a tremendous amount of professional, academic and practical knowledge.

I would also like to thank, Dr. Daphne Economou, who supported me and helped me in implementing the case study which comprises the backbone of my research. I am also thankful to Dr. Evangelia Kavakli and Dr. George Tsekouras for being members of my advisory committee.

Since the birth of the idea to return to the University, back in 2004, a lot has changed in my personal life; I have got married to a wonderful wife and have had two baby daughters. I state this to emphasize the extent to which I owe a debt of gratitude to my wife. Without her unlimited support it would not have been possible for me to bring my research to a close.

I would also like to acknowledge the Municipal council of Mytilene and ETAL SA, Lesvos, Greece for partially funding the case studies included in this research.

Special thanks go to my parents who have always offered among other things, their moral support throughout the years of my entire educational journey.

To my wife Mari,

and my daughters Joanna and Maritina,

TABLE OF CONTENTS

List of figures

List of Tables

Acronyms

.NET CF	.NET Compact Framework
aGPS	Assisted GPS
API	Application Development Environment
CDC	Connected Device Configuration
CIL	Common Intermediate Language
CLDC	Connected Limited Device Configuration
CLR	Common Language Runtime
CSS	Cascading Style Sheets
GIS	Geographical Information Systems
GPRS	General Packet Radio Services
GPS	Global Positioning System
GSM	Global System for Mobile communications
HTTP	Hypertext Transfer Protocol
ICT	Information and Communications Technology
IDE	Integrated Development Environment

IEEE	Institute of Electronic and Electrical Engineers
IRDA	Infrared Data Association
ITU	International Telecommunications Union
J2ME	Java 2 Micro Edition
JAVA ME	Java Mobile Edition
JCP	Java Community Press
JIT	Just In Time (Compiler)
JSE	Java Standard Edition
JSP	Java Server Pages
JSR	Java Specifications Request
JVM	Java Virtual Machine
LAN	Local Area Network
MIDP	Mobile Information Device Profile
MIT	Mobile Internet Toolkit
MSA	Mobile Service Architecture
MTG	Mobile Tourist Guide
MSA	Mobile Service Architecture

OS	Operating System
PC	Personal Computer
PDA	Personal Digital Assistant
POI	Point Of Interest
RIA	Rich Internet Application
SDK	Software Development Kit
TIDP	Tourist Itinerary Design Problem
TOP	Team Orienteering Problem
TOPTW	Team Orienteering Problem with Time Windows
TSP	Travelling Salesman Problem
TSPTW	Travelling Salesman Problem with Time Windows
UML	Unified Modelling Language
UMPC	Ultra-Mobile PC
UMTS	Universal Mobile Telecommunications System
WAP	Wireless Application Protocol
WLAN	Wireless Local Area Network
WLAN	Wireless Local Area Network

WML	Wireless Markup Language
WSN	Wireless Sensor Network
XHTML	eXtensible Hypertext Markup Language
XML	eXtensible Markup Language

CHAPTER 1

INTRODUCTION

1.1. The Landscape of Mobile Tourism

The new era of mobile communications is having an enormous global impact for innovative networked applications and information services. According to the ITU [78], there are approximately 4.6 billion mobile phone subscribers worldwide which represents over 50% of the world's population and globally, mobile devices outnumber PCs and regular phone subscriptions by almost four to one, and credit cards and TVs by two to one [5]. In many countries, the mobile phone has become an electronic wallet and the main means for accessing the web. In the next few years it is anticipated that more people will log onto the Internet via a mobile device rather than from a PC.

The combination of wireless devices, Internet connectivity and service provision technologies reveals tremendous potential in terms of emergent mobile applications. We are witnessing the introduction of a broader spectrum of mobile devices such as PDAs, laptops, cellular phones and Blackberries, as well as a wide range of communication support technologies for different types of networks such as Bluetooth, Wi-Fi, GPRS, ZigBee, and WiMax. In addition, a variety of new devices such as the Apple iPad and other portable media centres are being connected to the Internet. In the near future hundreds of millions of people will be carrying pocket-sized, networked computers. This will require innovation and extraordinary advances in the emerging area of mobile software engineering. Although the current state-of-the-art has established some elements to support mobile software engineering, proper engineering methods to design, build, evolve and deploy applications are underdeveloped or entirely missing and engineering approaches to new modes of mobile technology and user interaction pose significant challenges.

The convergence of information technology, the Internet and the telecommunications industry have also generated massive changes in the field of tourism. Gone are the days that tourists walked up and down on the high street in order to book the best holiday package they could find. Now, tourists use ICTs to search for destination information regarding prospective places to visit and the industry makes use of such technologies to offer information to tourists. In addition, with the advent of Web 2.0 technologies such as social networks, wikis, blogs etc, tourists often seek out information from other tourists. There are numerous websites on hand which offer this sort of personal information space to tourists such as tourist blogs, tourist diary notes and communication technologies to connect with family, friends and others.

In parallel, the mobile phone sector is showing a large increase in users with phones equipped with personal navigational systems and along with that a large increase in the usage of the mobile web. Inherently, an increase in mobile phone usage for services other than voice calls and text messaging is also evident. This is mostly due to the mobile phone transcendence from a traditional voice communication device to an instrument facilitating multimedia communications. The vision of nomadic users having seamless, worldwide access to a range of tourist services is expected to be a reality within only a few years. Hence, the concept of ‘mobile tourism’, wherein users access tourist content through mobile devices, has recently emerged [25].

Undeniably, mobile devices present many unique characteristics that make their use in the mobile tourism paradigm (namely as mobile tourist guides) particularly attractive: These characteristics are….

(a) Ubiquity and convenience: Use on the go, anywhere, anytime,

(b) Positioning: Using technologies like GPS, users may receive and access navigational information and services specific to their location [170],

(c) Personalization: Unlike PCs, handheld devices are typically operated by a single user, thereby enabling the provision of personalized services through wireless web portals [73].

There have been many research initiatives initiated in the area of mobile tourism. Many of which have identified indoor as well as outdoor services. In museums for example a number of services have already been operational for some time now. A few outdoor examples are location-aware and location-based services. Such services consider the location of users and use this to adapt the information offered.

1.2. Research Questions & Motivation

Mobile devices are designed to be used in many situations and for various applications, i.e. mobile computers, mobile phones, digital cameras, portable media centres, etc, and are most suitable in fields that require the assistance and convenience of certain aspects common when using a conventional computer, in environments where carrying one would not be practical.

However, mobile devices have a large number of limitations, which may vary, depending on the device used. Several limits inherent in mobile devices need to be carefully evaluated by tourist service providers. Some of the problems which may be encountered include:

- Small display size, which normally results in a lack of windows support for multitasking and small/hard to use keyboards resulting in different navigational methods from those known to the majority of PC-based users,

- Limited computing power and restricted energy capacity,

- Limited coverage and/or bandwidth. Admittedly worldwide coverage is increasing which leaves fewer places without coverage. However, many areas are still not covered by 3G network infrastructures.

- High cost of wireless connections/roaming charges; typically when roaming applies, i.e. in most tourist circumstances charges which accumulate for the tourist are far from insignificant.

- Fragmentation. The inability to "write once and run anywhere". The mobile market faces the problem wherein mobile applications must be ported to multiple end-devices with divergent characteristics.

In summary, small screens mean fewer visible options for users at any given time, requiring users to rely on their short-term memory to build an understanding of an online information space. This makes almost all interactions rather difficult. It is also tricky to use multiple windows or other interface solutions that can support advanced behaviours. Mobile devices also have awkward input, especially for typing. It is hard to operate GUI widgets without a mouse: Items such as menus, buttons, hypertext links and scrolling all take longer and are more error-prone, whether they are touch-activated or manipulated with a small trackball. Text entry is particularly slow and is prone to error, even on devices with dedicated mini-keyboards. Often getting the next screen takes forever, sometimes even with a supposedly faster 3G service [125].

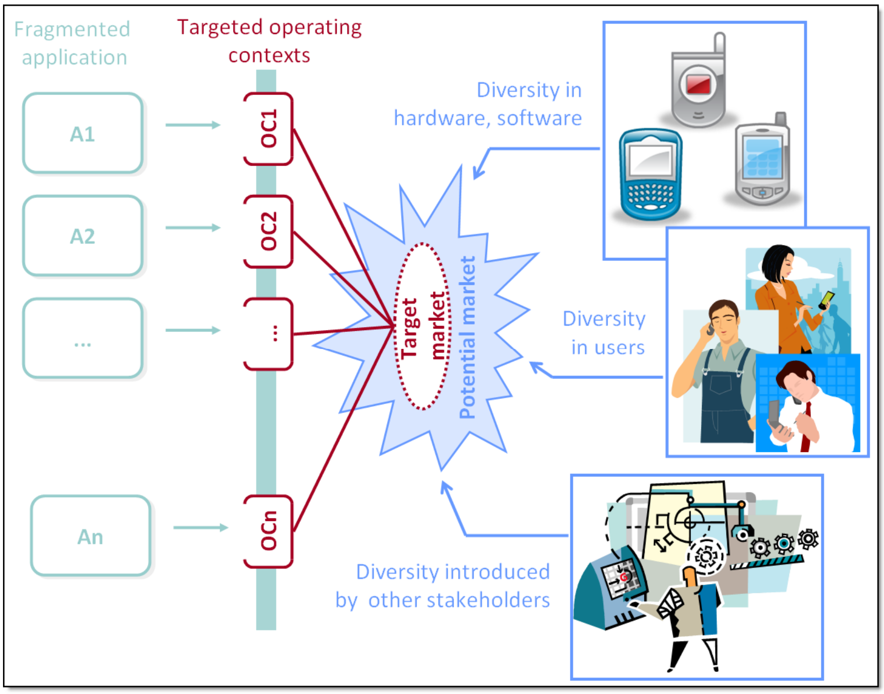

Fragmented application
Targeted operating contexts
A1
A2
...
An
OC1
OC2
...
OCn
Target market
Potential market
Diversity in hardware, software
Diversity in users
Diversity introduced by other stakeholders

Figure 1-1. Fragmentation of mobile applications (Source: [41])

Admittedly, although the capabilities of mobile devices are increasing steadily, the 'resource gap' between mobile and stationary devices will still be there. As with stationary devices, the idea of porting an application to different operating contexts, i.e. different platforms like PC, Mac, Linux, etc, is more or less straightforward. Yet, because of the large number of mobile devices available, serious issues concerning porting mobile applications to the developers target market arise. In the mobile market this problem is often called 'device fragmentation' (see figure Figure 1-1) and does not only originate from the differences in device diversity, i.e. screen parameters, memory size, processing power, input mode, etc; it may also originate from other parties such as software platform diversity (i.e. different available platforms), implementation diversity (i.e. the need to test the mobile application on each handset physically due to small bugs and quirks in the system), user-preference diversity (e.g. language, styles, accessibility requirements, etc) and environmental diversity (e.g. branding by carrier, compatibility requirements of the carrier, backend APIs, gateway characteristics, opened ports, restrictions on access to outside the network, locale, local standards etc,); all the abovementioned parameters play contribute to the fragmentation problem [41].

Most application developers build separate applications for the stationary device and for each of the mobile platforms they wish to target. This results in large development overheads and the consumption of many man hours. Moreover, today we can clearly see the mobile phone sector churning out new phones in the market place supporting new operating systems, new platforms and overall new business paradigms. A recent example of this can be seen with the Mobile Application Stores coming into the spotlight these days. Examples of this can include the iPhone and developers' applications accessed via itunes [11], Nokia offering developers' application to be downloaded via OVI [135], Google releasing the Android platform and Market for mobile developers [10], Samsung releasing the new bada operating system [13], Microsoft realising the Windows market place for mobile [177], and the also BlackBerry and the Research In Motion's store, etc. Even these recent developments bring forward the need for solutions from the software industry which can compensate for the fragmentation of devices in conjunction with the constraints of mobile browser capabilities. This in turn brings about problems of porting mechanisms to mobile phone devices which need to be investigated.

As for the tourism sector, mobile tourism applications have attracted considerable research interest during the past decade resulting in numerous commercial and research applications [26], [8], [35], [45], [134]. Most of the existing approaches in the field of mobile tourism basically fall within two main categories:

- Tourist guides as pre-installed applications, namely rigidly defined content (in text, visual and auditory format) that cannot be customized according to user preferences (e.g. [26]).
- Mobile devices used to access mobile web portals and browse tourist information of interest (e.g. [8], [35], [45], [134]).

The first approach implies a static application be ported to a mobile device resulting in large file sizes and the problems of porting to mobile devices. The second approach implies the use of a mobile or wireless network to access Internet resources. The main disadvantage of a connection service is the requirement for constant connection (airtime) of the mobile device with a mobile network to offer access to web content. Thus, users are charged for the wireless connections (if there is no flat fee for use, either pay-per-minute or pay-per-packet billing policy applies, which is typical where roaming applications apply). Most importantly, whenever a user is out of the coverage area of the mobile network (i.e. 'has no signal') he/she cannot access any service.

We argue here that there is a need to research platforms which enable the creation of portable tourist applications with rich content that can match users' personal preferences. The users should be able to download personalized applications optimized for specific device models either directly to their mobile device or first to a PC and then to a mobile terminal. Thereafter, network coverage is not further required as the applications execute in standalone mode and may be updated when the user returns online. This issue is investigated throughout this thesis.

Certainly, personalization has been recognized by researchers as a critical factor of efficiency, added value and commercial success in tourism [144]. Personalization systems originally found success in e-commerce sites providing recommendations for products and offering information to consumers to aid decisions on product or service purchases. In mobile tourism, personalization has mainly been addressed in the context of guides providing content recommendations that either match individual user preferences (typically consolidated in 'user profiles') or depend on the user's current location. We argue that existing recommender systems fail to take into consideration the context of the mobile tourism user, i.e. mobile tourism recommender systems should also exploit information, behaviours, ideas, evaluations, assessments, ratings, etc, by other tourists with similar interests.

These systems can further provide the foundation for the cooperative production of tourist content and other travel recommendations.

In addition to personalization, it is often the case that tourists are confronted with the decision of which POIs to visit while at their destination. Even if the user eventually chooses which POIs to visit, they then have to decide in which order to visit these POIs and how to get there. This is sometimes even more laborious when the area has many POIs to visit and the tourist has a limited time to visit them all. Solutions such as guide books might not be the correct choice given that the information included could be outdated or incompatible with user preferences. Mobile tourist guides may be used as tools to offer solutions to these types of problems [34], [107]. Based on a list of personal interests, up-to-date information for the sight and information about the visit (e.g. date of arrival and departure, accommodation address, etc); a mobile guide can suggest near-optimal routes that include visits to a series of sights. It may also recommend the order of each sight's visit along the route [171]. The issue of near-optimal personalized daily itineraries for a mobile tourist guide is investigated in this thesis.

Last but not least, the formulation of usability guidelines as well as the usability testing of mobile applications represents emerging research areas that face a variety of challenges due to unique features of mobile devices. We argue that usability tests should be performed both in a laboratory environment and through field trials. Experimental testing is more suitable for standalone mobile applications, i.e. those without network connectivity requirements [186]. In contrast, field studies allow the use of mobile applications in a realistic environment and take dynamic mobile context into consideration, which is difficult to simulate in laboratory experiments [94].

In this context, this doctoral research has contributed to the growing body of literature related to mobile tourism, the understanding of open research issues and challenges and building a solid know-how in developing practical mobile tourist applications.

1.3. Overview and Style of Thesis

This thesis is organised in the following fashion. This chapter, which comprises the introduction to the thesis. Chapters 2 and 3, which present state-of-the-art work (literature review) on authoring tools, development platforms and existing mobile tourist applications. Chapters 4, 5 and 6 constitute the core material of the thesis, describing our innovative research work results. Finally, Chapter 7 summarises the thesis, presents the conclusions and points toward directions for future work.

Chapter 2 comprises two main parts. First, it investigates the general requirements regarding the design of PDA and mobile phone applications and then focuses on specific application-related issues for cultural organizations and tourism. Following that, it presents the development and design facilities provided by some typical tools for multimedia application development and service provisions for PDAs and mobile phones.

Chapter 3 focuses on the evaluation of research and commercial applications used by tourists to retrieve information, navigation and guidance using some form of mobile device. Following the evaluation, a number of design guidelines are extracted for the design and the development of a complete mobile tourist framework.

Chapter 4 presents the design and implementation aspects of a multiplatform tourist framework which, amongst other services, allows for the tourists to use the web in order to 'build' unique mobile standalone guides that run on any mobile device offering tourist information but also providing additional personalized tourist services. The optional use of a network connection to enhance the mobile tourist guide application by offering personalized content and services is also presented in this Chapter.

Chapter 5 describes extensions upon the mobile tourist framework. In particular, we extend the notion of travel recommender systems utilizing collaborative filtering techniques whilst also taking into account contextual information (such as the current user's location, time, weather conditions and places already visited by the user) for deriving improved recommendations in pervasive environments. Thereby, we introduce the concept of 'context-aware collaborative filtering'.

Chapter 6 describes the problems that tourists typically deal with in selecting tourist sights to visit as well as the visiting order while at a destination, for each day of stay. In this Chapter, we define and model the 'tourist itinerary design problem' and propose a heuristic algorithm running on the web and mobile platform as a solution. This algorithm is compared to alternative approaches reported in the literature; one of those algorithms that directly compares to our work has been implemented for testing and evaluation purposes.

Chapter 7 brings together the various strands of the thesis and summarises the main findings. It explains the significance and contributions of this thesis and also identifies areas in which the work presented herein could be developed further. Bibliographical references follow at the end, along with an Appendix listing the author's publications in international journals, conferences and edited books.

CHAPTER 2

MOBILE DEVELOPMENT PLATFORMS:

DEVELOPING CULTURAL AND TOURIST APPLICATIONS

2.1. Introduction

A considerable amount of research has been dedicated to the use of multimedia technologies in the fields of cultural organizations and tourism for the provision of cultural interpretational information [54], [103], [138], [180]. Mobile devices have also been gaining increasing acceptance as a means to provide cultural multimedia applications due to their physical characteristics and suitability to these fields. This is evident by a number of research prototypes and commercial projects that have been reported in literature [8], [15], [35], [54], [98], [113], [138], [144]. Currently, most tools used to develop multimedia applications for mobile devices are light versions of state-of-the-art multimedia authoring tools, which are not tailored to adequately satisfy user, designer and mobile device applications requirements.

Mobile applications are distinct from other application domains with respect to the type of content provided, the way this content is made available to the user and the way these types of applications are developed. Cultural and tourist applications require rich multimedia content with a highly interactive interface and should be customized for a wide range of devices. To minimize user reluctance and enable affordable usage, the access to content and services should not require a constant network connection. Also, cultural applications pose a requirement for multi-modal media provision (e.g. graphics, audio, video) which are not always applicable to other application fields of mobile computing (e.g. mobile commerce, calendars, news, etc); some of these requirements though, may be projected onto other fields as well, such as mobile education.

Developers have many different tools to choose from when targeting the mobile device sector. Developing mobile applications may prove a complicated task because, on one hand, there is a never-ending stream of innovations and new mobile technologies being released on a regular basis and, on the other hand, it can take a painfully long time for these new technologies to actually become usable for the end customers [174]. Undeniably, because of the large number of mobile devices available, each having unique features, problems in porting to a fragmented range of devices arise. Most mobile software houses have to develop separate applications for each of the targeted mobile device platforms, resulting in large development overheads and outlay of man hours. This in turn has raised issues of development platforms and different variations in porting to mobile phone devices.

The aim of this chapter is to provide an overview of the mobile applications development landscape and identify a coherent set of requirements that application development platforms should satisfy in order to allow for the effective, efficient and economical development of cultural applications on mobile devices. Also, to highlight the main characteristics as well as the relative merits and shortcomings of the most popular mobile platform development options for mobile devices. This study deals with two parties, the cultural application "end user" and the cultural application "developer". The former determines the mobile cultural application needs, while the latter refers to authoring tools and application development platform requirements.

The remainder of this chapter is structured as follows: Section 2.2 outlines the evolution of the mobile phone in respect to HCI and mobile device characteristics. Section 2.3 discusses issues related to the design of cultural and tourist mobile device applications. Section 2.4 describes the main features of typical tools for multimedia application development for mobile devices. Section 2.5 presents a comparison of mobile application development environments. Section 2.6 concludes our thoughts on what holds for the future of mobile development platform. Section 2.7 identifies a set of requirements for cultural mobile application development platforms. Finally, Section 2.8 concludes this chapter.

2.2. Types and Evolution of the Mobile Phone

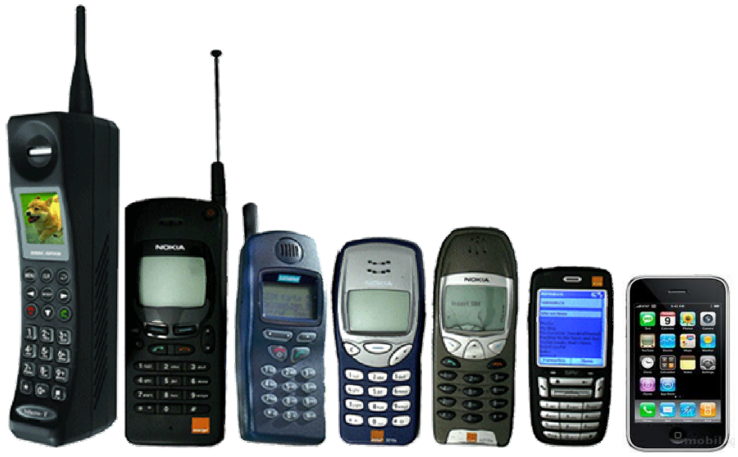


Figure 2-1 Evolution of the mobile phone (Source [31])

Since the early 1990's the mobile phone has evolved to such a degree that computing and communication is now possible from the smallest, pocket-sized mobile smart devices, whilst delivering excellent visuals and speed. Figure 2- depicts the evolution of the mobile phone starting from the literal size of 'brick'; a non feature phone with little or no features all the way up to smartphones which are small in size, yet rich in hardware capabilities and software features. The term 'feature phone' is used to describe a low-end mobile phone with limited –compared to smartphones- computing ability, but more powerful than a "dumb phone". The term 'smartphone' refers to a mobile phone that offers advanced computing capabilities and connectivity than a contemporary basic "feature phone".

The main differences between older mobile phones and newer mobile phones whether they are feature or smart phones are:

- Old phones used to have small display screen and big keypad, whereas new phones have big display screen and smaller keypad.
- Old phones had only black and white display, whereas newer phones have colourful display.
- Older phones have text display, whereas newer phones have graphical display

Today's smartphones and feature phones may be thought of as handheld computers integrated within a mobile telephone. While most feature phones are able to run applications based on platforms such as Java ME [163] or BREW[1], a smartphone allows the user to install and run more advanced applications based on a specific platform (see Section 2.4).

Feature phones are often compact and their keypad and overall operation is generally straightforward. All allow to store frequently used numbers and to send/receive text messages. Many are equipped with cameras and support for wireless Bluetooth headsets for hands-free communication. Many can access high-speed data networks to enjoy music and video-based services. Other capabilities might include a touch screen, a QWERTY keyboard, a full browser, a multi-megapixel camera, memory-card storage for music and pictures and more options for custom ring tones, games, and other services.

A smart phone can typically handle multiple e-mail accounts (including corporate types) and Office documents. Their advanced operating systems provide them access to a bunch of applications: productivity tools, shopping,

[1] BREW is an acronym that stands for Binary Runtime Environment for Wireless. From a software developer's perspective, Qualcomm's BREW can be viewed as a set of APIs that enable developers to create software applications for wireless devices (wireless phones for now), and a means of selling and delivering applications to end-users

multimedia, maps, games, travel, news, weather, social, finance, references, etc.

2.3. Mobile Application Design Requirements

2.3.1. Issues related to the design of mobile applications

According to Dunlop and Brewster [47], various factors should be considered in the design of successful mobile applications mainly related to the device's technical characteristics and the use of the application.

These factors confront mobile application designers with new challenges, such as:

- Mobile context, which can be defined as "any information that characterizes a situation related to the interaction between users, applications and the surrounding environment" [43],
- Design for mobility,
- Design for multimodality (input with voice and touch with relevant spoken output and onscreen visual displays) and user multitasking at levels unfamiliar to most desktop users,
- Design for a large user community with a wider range of skills and expertise,
- Design for limited input/output facilities, which means restrictive data entry methods and small screen size with different display resolutions,
- Limited processing capability and power,
- Design for providing information based on the user's location.

Nevertheless, several issues differentiate the requirements for mobile applications. The major difference between mobile devices as a general rule is the current screen size. For the design of mobile applications, one must consider the limited screen space the mobile device has to offer [76]. More recent releases of high-end mobile devices have larger screen sizes[2]; however these still represent a minority of the market. Specifically, designers of mobile phone applications have to consider three main categories[3] of screen sizes when designing applications [76], unlike PDAs where standards in screen sizes do not vary considerably.

Another issue regarding designing applications for mobile devices is related to limitations of computational power and memory capacity. This is an immense problem - particularly for multimedia applications that require fast processing speed and a large amount of memory for graphic support. Due to the limited processing capability of mobile devices, developers may have to disable some functions (e.g., high resolution images and dynamic frame movement). The computational constraints of mobile phones have been addressed by the evolution of high end smart phones e.g. iPhone 4 and of the advent of the Ultra-Mobile PC (UMPC) [167], which offers consumers a small, thin and light ultra-mobile device with PC-like capabilities, Internet access and anytime connectivity. These devices have enough processing power to support audio, video, and gaming, in addition to having rich support for browsing the internet as well as other communication and networking applications. They can also feature inbuilt GPS devices, webcams, Bluetooth, Wi-Fi, etc. these devices are expected to be supported by a wider range of developmental tools as they do not suffer from the typical constraints of current phones.

Additionally, the number of dedicated mobile phone applications being used on a daily basis is still limited. Consequently, users cannot easily draw upon past experiences of such applications [47]. Unlike the PDA, most mobile phone applications are downloaded over-the-air (OTA), which implies lack of user manuals, long installation time and -possibly- high download cost. Mobile

[2] The iphone 4 has a screen resolution of 960-by-640 pixels

[3] The 3 most popular screen sizes are 128-by-160 pixels, 176-by-220 pixels and 240-by-320 pixels.

phone devices in general have less processing capabilities than PDAs; therefore applications targeting mobile phones need to be highly optimized and customized to meet a broad range of user and devices requirements.

2.3.2. Issues related to the design of mobile applications for cultural organizations and tourism

The design of mobile applications for cultural organizations (e.g. museums) should address the organizations' requirements and should have as a goal to provide visitors a unique and pleasant experience [39]. Woodruff et al. [180] studied the visitor behaviour using PDAs in museums and identified certain issues related to user interaction with the PDA application:

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

In terms of the interface design, mobile applications should pursue criteria similar to web site development [39]. In addition, the Canadian Heritage Information Network adds some practical guidelines [30] for the graphic design of the interface:

- Each screen node of the PDA application should fit the size of the mobile device
- 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 [137].

Requirement gathering studies performed at the CHMLab[4] in close collaboration with cultural organizations identified a list of design factors that need to be taken into consideration in order to develop effective and efficient cultural mobile applications that enhance the visiting experience [50]. The mobile application developer should:

- Provide a means of promoting the cultural content (cultural artefacts) and connecting it with related information (e.g. for an artist that might have created the exhibited artefacts, the techniques used to create the artefacts, the artist's inspirations or even information for parts of an exhibition that are presented in another museum, etc.) without bounding the user to the cultural organization's/exhibition's physical space;
- Apply multimedia techniques to provide interpretative information about a collection in order to be able to effectively assign meaning related to them;
- Guide the user by implying translucently a path that the user should follow to view an exhibition;
- Focus the visitors attention;
- Personalize the provided information to the visitors interests and experience;
- Promote services and receive visitors' feedback in order to improve services;
- Support pre- and post-visit interaction with visitors;

[4] Cultural Heritage Management Lab, Dept of Cultural Technology and Communication, University of the Aegean

- Support the formation of interest groups.

As far as mobile phone applications are concerned, most of the research work and prototypes are restricted to tourist applications. This is mainly due to the physical restrictions of this class of device as explained in section 2.3.1.

Kray and Baus in 2003 completed a survey of available tour guide applications [96]. Many issues were retrieved relating to mobile phone application 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.

2.4. Mobile Application Development Platforms

In reality, there are many different mobile application development platforms to choose from, including native environments (Symbian, OpenC, iPhone, Palm OS, etc), runtimes (Java ME, .NET Compact Framework, Python, Lazarus, BREW, Flash Lite, Android, etc) and web runtimes (widgets, widsets, plusmo, bondi etc) [58]. Those are typically incompatible with each other and in most cases will only work on one particular platform. In order to maximise reach and revenue for their applications, developers must decide carefully which platforms they will support.

The contemporary mobile application development platforms give the opportunity to application developers to implement applications using tools such as Java, Open C, Python, Flash Lite or Web technologies (such as XHTML/CSS, JavaScript, Mobile Ajax and mobile widgets) to create highly functional mobile applications. These application development platforms can be categorised as such:

- Platforms supporting devices by multiple manufacturers,
- Platforms supporting devices by one manufacturer

Platforms supporting devices by multiple manufacturers include:

- Java ME
- Symbian platform
- Android platform
- .NET Compact Framework
- Qt (framework) Qt
- BREW
- Windows Mobile
- Palm OS
- Flash Lite
- Microbrowser (Web) based applications

Where as platforms limiting support to devices by one manufacturer are:

- Blackberry
- Nokia OVI
- Samsung Bada
- iPhone OS

In this section, we briefly discuss the most important characteristics of runtime environments for handheld devices (mainly mobile phones and PDAs), which currently enjoy the largest developers and deployment base: Java ME, .NET Compact Framework, Flash Lite and Android.

2.4.1. Java Platform, Micro Edition (Java ME)

The need for defining a computing platform that could execute Java applications and be supported by small electronic devices led to the development of the Java Platform, Micro Edition (Java ME) [82] by Sun Microsystems now a subsidiary of Oracle Corporation, in 1999. Java ME 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. There are over 2.1 billion Java ME enabled mobile phone devices worldwide [81] which inevitably leads to a very large developer database.

Java ME follows a modular design which aims at simplifying the support for a broad range of small devices. Thus, it introduced the concepts of 'configuration level' and the 'profile level' (see Figure 2-). The former defines 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 latter includes libraries specialized in the unique characteristics of a particular class of devices. The configuration and profile currently supported by all JAVA ME-enabled mobile devices is the CLDC (Connected Limited Device Configuration) and the MIDP (Mobile Information Device Profile), respectively. Java applications developed over CLDC/MIDP are called MIDlets, usually packaged in *.jar files.

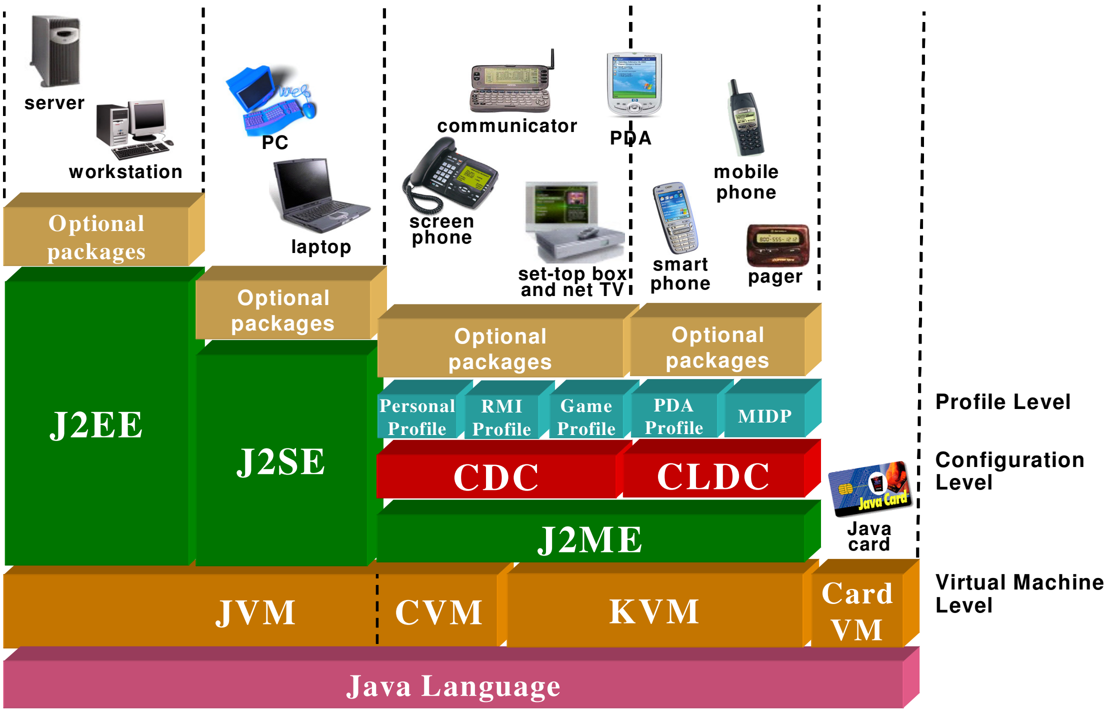


Figure 2-2. Java technologies and their respective target devices (adapted from [184]).

MIDlets are typically downloaded on-the-fly from a web server and are 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 which download information on-demand. The communication of MIDlets with web servers is carried out over the Internet's HTTP 1.1 protocol.

JAVA ME platform presents many advantages that point to 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 mark-up language).

Java ME 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 mark-up language requires the use of specialized parsers.

Java ME 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 back-end infrastructure. This is in contrast to mobile microbrowser supported applications that require constant connection with the mobile network [67].

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

Most importantly, the over-the-air download of new Java ME applications (jar files) is costly, slow and consumes network resources. With respect to the latter disadvantage, it is evident that Java ME-based mobile tourist guide applications should -ideally- be downloaded only once; synchronization with the back-end server should thereafter be considered only when the user wishes to update the selected tourist content.

Since Java ME is designed to be cross-platform, the Java ME specification and implementation are two separate processes. Through the Java Community Process (JCP), a formalized process which allows interested parties to get involved in the definition of future versions and features of the Java platform, a committee of mobile solution providers decides the new Java

ME standard APIs. After the API specification (JSR[5]) is developed, each company can develop its own implementation preserving vendors' incentives to differentiate and innovate.

2.4.2. Microsoft .Net platform for the Mobile Web

The .NET Compact Framework (.NET CF) [115] is a subset of the full .NET platform designed by Microsoft for applications on Windows Mobile. .NET CF provides a runtime engine (Common Language Runtime, CLR) preloaded in the device's memory in order to facilitate mobile application deployment. The CLR provides interoperability with the underlying device OS allowing the integration of native components into mobile applications. In principle, the .NET CF runtime is analogous to the JVM. Instead of writing native code for the underlying OS, .NET developers write managed code which targets a managed execution environment. Microsoft originally designed and developed the .NET platform with support for multiple languages and operating systems (aimed at reaching an extended base of developers and reusing existing libraries). However, the .NET CF development tool (VS.NET), currently supports only two major .NET languages (C# and VB.NET) while OS support is restricted to Windows platforms which only represents a small part of today's mobile device population.

The core components are a subset of the full .NET Framework, bringing roughly ~30% of the classes and functionality. Some classes exist in both .NET and .NET CF, however the .NET CF version does not necessarily support all the class members (properties, methods, or events) of the full version. Many classes are not implemented at all or are partially implemented. Unique .NET CF classes (device specific and third-party extensions) are also provided, while UI design is based on a rich subset of .NET Windows Forms. The software stack of the .NET CF platform is illustrated in Figure 2-.

[5] The JCP involves the use of Java Specification Requests (JSRs), the formal documents that describe proposed specifications and technologies for adding to the Java platform. A final JSR provides a reference implementation which is a free implementation of the technology in source code form.

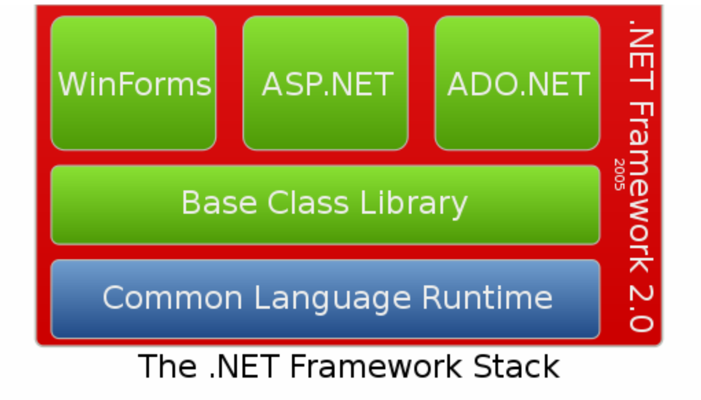


Figure 2-3. .NET CF platform software stack.

2.4.3. Adobe Flash Lite

Adobe Flash Lite is referred to in this section as one of the most commonly used multimedia authoring tools created to enable easy and rapid content deployment to mobile devices. There has been large adoption of Flash Lite by Original Equipment Manufacturers (OEMs), operators and developers which is growing worldwide [3]. The Flash Lite authoring environment provides 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 are already skilled in working with Flash Professional and can easily adopt in 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. The software stack of the Flash Lite .NET CF platform is illustrated in Figure 2-.

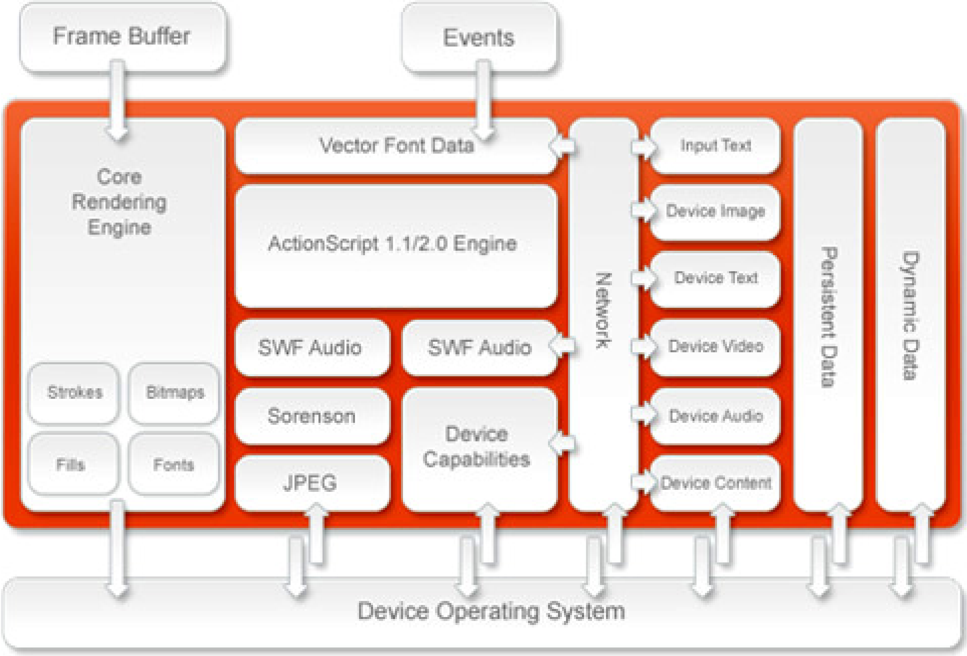


Figure 2-4. FlashLite platform software stack (Source: Adobe).

The key features of Flash Lite as of Adobe's product page [4] are:

- FLV support — including H.264, On2 VP6, and Sorenson video codecs;
- A more complete web experience on mobile devices by providing access to content and video created with Adobe Flash;
- Delivery of high-definition video content and rich applications to Internet connected TVs and TV connected consumer electronic devices in the digital living room;
- Optimized performance for mobile and digital home devices;
- Object-based extension mechanism for accelerated UI and application design;
- Integrated authoring environment for mobile devices;
- Multiplatform support.

2.4.4. Android Platform

Google's Android 1.0 [9] was officially launched been launched in 2008, devoted to advancing open standards for mobile devices. Current version of the Android Platform in the writing is version 2.2. Android is an Apache free-software platform with open-source licence for mobile devices based on Linux. It essentially comprises a software stack for mobile devices that includes an OS, middleware and key applications.

Android applications are primarily written in Java and compiled into a custom byte-code (dex). Each application executes on its own process, with its own instance of Dalvik virtual machine. Dalvik runs dex files, which are converted at compile time from standard class and jar files; dex files are more compact and efficient than class files.

Developers have full access to the same frameworks and APIs used by the core applications as well as Google-developed software libraries. The Android's software architecture is designed so as to simplify components' reusability: any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). The Android SDK enables authoring applications with rich functionality: just like the iPhone, it can handle touchscreen, accelerometer, 3D graphics and GPS; collaboration among applications (e.g. email, IM, calendar, social networking); location-based services. The software stack of the Android platform is illustrated in Figure 2-.

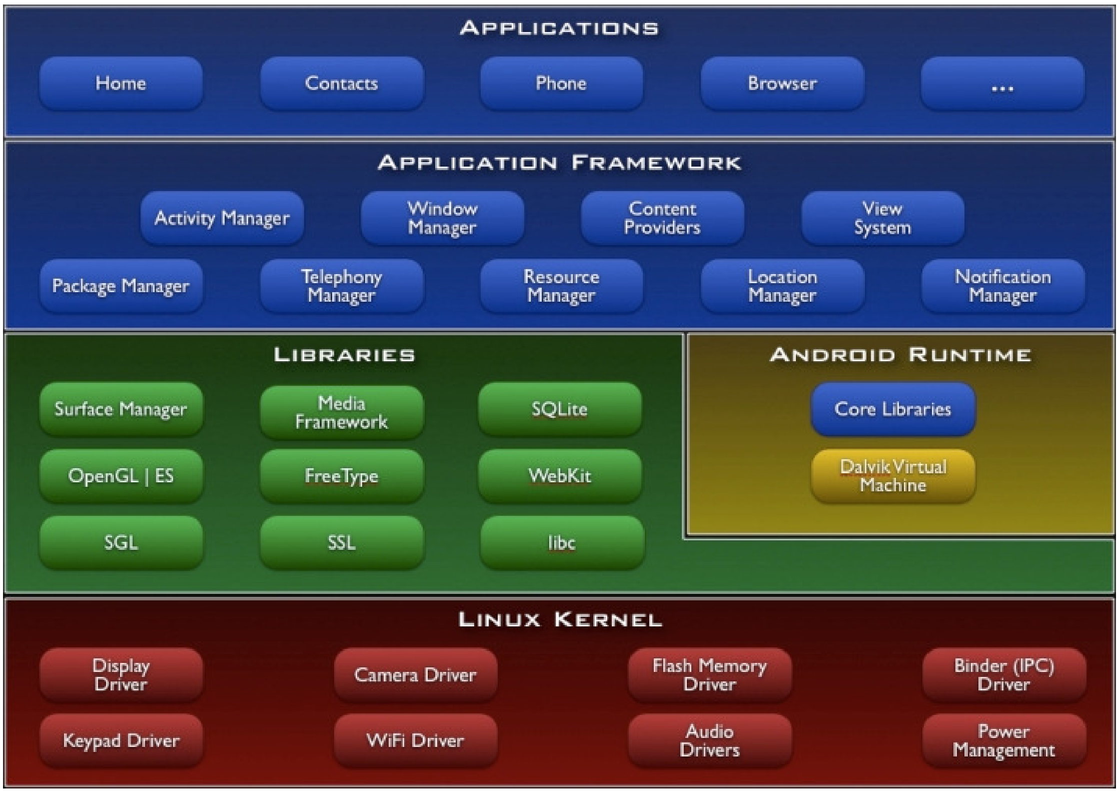


Figure 2-5. Android platform's software stack.

2.5. Comparison of Application Development Platforms: Software Architecture, Development, Functionality, Porting and Market Success

Java ME is the dominant mobile software platform with respect to its installation and developers base. However, the 'write once, run anywhere' axiom of the Java language does not apply to the world of Java ME [163] choice of profiles, configurations and Java ME APIs (specified in JSRs), developers are required to provide slightly different application versions (not all phones support the same set of JSRs, while JSR implementations vary across devices), which often results in dozens of executables for a given title. This is commonly referred to as device *fragmentation*, which considerably increases operational costs as it affects the development-testing-signing-delivery-maintenance cycle. Fragmentation restricts Java applications device reachability and suggests Java ME only for vertical applications, i.e. for limited targeted devices with similar capabilities and Java APIs support.

By targeting individual OS platforms, mobile application developers may use a large set of well-defined and mature JSRs (e.g. Java applications targeting the Symbian platform can reach ~70% of the world's smartphones). There are currently more than 80 JSRs providing the MIDlet developers with a rich and diverse set of additional technologies; however, MIDlets programming is not straightforward as it requires serious Java development skills. Examples of commonly available JSRs that extend MIDP 2.0 on the Symbian platform include the Bluetooth API (JSR 82), the Wireless Messaging API (JSR 205) and the Mobile 3D Graphics API (JSR 184).

.NET CF is comparable to Java ME with respect to providing: (a) a managed runtime environment to manage security, memory usage and runtime optimization, (b) rich libraries and components to allow developers to reuse software modules (advanced UI components, network connectivity, data management, XML web services, etc), and (c) familiar APIs from the full .NET framework (e.g. the Windows Forms controls) which allow as smooth transition of desktop developers to mobile development.

The use of a runtime system for intermediate (managed) code (similarly to Java ME) implies relatively low execution performance. Unlike Java though, it is designed to be language-agnostic and simply specifies the Common Intermediate Language (CIL) instructions. Therefore the .NET supported languages (i.e. C# and VB.NET) compile to the same CIL and can be executed by the .NET CF runtime.

.NET CF demonstrates improved API level consistency and compatibility with the full .NET platform; this design approach though has had an unseen cost in terms of memory footprint. .NET CF represents a fast-paced implementation driven by a single powerful vendor; developers have specific hardware specifications to program against and can assume the availability of certain native software (e.g. Windows Media Player). Therefore it offers satisfactory integration with device-specific functionality (telephony, SMS messaging, accessing the SIM card, Bluetooth, etc) and does not exhibit the

Java ME’s fragmentation problem. On the other hand, .NET CF targets a limited set of -Windows- end devices and lacks free development tools (Visual Studio is shipped with a license cost).

Flash Lite platform is a reasonable choice for graphics-intensive phone and PDA applications. It has enjoyed increasing adoption by mobile technologies industry players, since the developers already skilled in working with Flash for desktop applications can easily switch into using Flash Lite to design mobile applications. The main assets of Flash Lite include rapid development (easy to learn, fast migration from Flash applications, rich designer/developer tools), rich media support (images, video, sound, animation), a relatively broad runtime installation base and small deployment files (more animation and graphics packed into the same file size when using SVG). As of Flash Lite 2.x compressed SWF may be used, while Flash Lite 3.0 added support for the popular native Flash video (flv). Currently, Flash Lite is mostly suitable for creating animations, casual games, mobile web-based Flash applications, front-end interfaces, device-specific content (e.g. wallpapers, screen savers, etc), however not for developing full-fledged standalone applications mainly due to the lack of powerful mobiles-oriented APIs (e.g. compared to the Java ME platform).

On the other hand, Flash Lite exhibits relatively poor graphics performance (partially due to the complex processing required for vector graphics), while it’s extensive toolset (Adobe CS4, Adobe Device Central) is shipped with a licence fee. Its low-level device integration can be perceived as limited; ‘outside the box’ low-level device APIs are offered by third parties (e.g. KuneriLite toolkit extends Flash Lite capabilities in the Symbian platform) allowing the development of powerful, innovative applications at the expense of raising fragmentation issues and increasing memory footprint.

Android supports a relatively large subset of the JSE 5.0 library (implying reduced migration cost from Java desktop applications) along with several third-party libraries; similarly to Java ME, application development is powered by popular Java IDEs (e.g. NetBeans, Eclipse). An important asset of Android is its inherent support for modular service-oriented applications and inter-application communication (MIDP 3.0 of Java ME also supports inter-MIDlet communication).

New platform releases introduce many new features for users and developers (accounts synchronization, improved media playing performance, database and geolocation API support, etc) but raises fragmentation concerns (soon there will be phones on the market running Android 1.0, 1.5, 1.6, 2.0, 2.1 and 2.2) as applications may have trouble working smoothly across all the OS versions. The platform's openness and the growth of targeted devices' stack aggravate the fragmentation problem**.**

Table 3-1 evaluates the reviewed platforms with respect to software architecture, application development, capabilities and constraints, developer communities and market success and development tools.

	Java ME	**.NET CF**	**Flash Lite**	**Android**
Software Architecture and Technical Issues				
Footprint	Around 128 KB for storage of KVM and the associated libraries	1.55MB on Windows Mobile-based Pocket PC 2000/2002; 1.35MB on Windows Mobile for Pocket PC 2003 or Windows CE .NET Devices	450KB for the core library of Flash Lite 2.1; 374KB for Flash Lite 3.1	3MB
Runtime memory requirement	< 0.5 MB	~ 0.5 MB	2-4 MB	The minimum requirement for Android handsets is 32 MB of RAM
Memory management	Automatic memory management provided by the 'traditional' garbage collector, which de-allocates memory occupied by objects that are no longer used by the program.	Automatic memory management provided by the CLR. The CLR garbage collector manages the allocation and release of memory for an application.	Garbage collection executed automatically every minute or whenever memory usage of the application increases by 20% or more.	Automatic memory management handled by Dalvik's garbage collector (garbage collections might be noticeable through performance drops)
Devices support	All devices supporting CLDC/MIDP (practically lacks support only for Windows Mobile-based Pocket PCs)	Pocket PC 2000, Pocket PC 2002, Windows Mobile 2003-based Pocket PCs and SmartPhones, embedded systems running Windows CE .NET 4.1 and later	Supported by major mobile phone and PDA manufacturers (e.g. Fujitsu, Hitachi, LG, Mitsubishi, Motorola, Nokia, Panasonic, Samsung, Sanyo, Sharp, Sony Ericsson ,…)	Mostly dominated by HTC (Magic, Hero, Tattoo). Also, T-Mobile (G1, Pulse), Motorola Dext, Samsung Galaxy i7500, Acer Liquid, Sony Ericsson Xperia X10. Android 2.0-compatible handsets already announced (by Motorola, Sumsung).
UI components	High-level LCDUI components (e.g. Form or List), Low-level LCDUI (for controlling every pixel of the UI), support for SVG (defined in JSR 287). J2ME Polish allows design along with animations and effects specified in external CSS-like files.	Windows Forms controls (vary for Pocket PCs and SmartPhones)	Nokia Flash Lite Feather Framework (FL 2.x), Sony Ericsson Adobe XD UI Components (FL 1.1/2.x)	The UI is built using View and ViewGroup objects. DroidDraw tool serves for rapid UI design. J2ME Polish enables conversion of Java ME MIDlets' UI to Android-compatible UI.
Development	Java (CLDC/MIDP)	C#, Visual Basic .NET	ActionScript 1.0,	Java (Android SDK)

languages			ActionScript 2.0	
Packaging	jad (application descriptor in text) and jar (application)	Cabinet (cab) file installers	swf files	Android package (.apk) files
Deployment methods	Over-the-air (OTA), Bluetooth/IR, WAP Push	OTA, bluetooth	bluetooth, physical cable, OTA	OTA, bluetooth
Server-side technologies (*)	Java Servlets, JSP	ASP.NET Mobile Controls	Flash Media Server (uses ActionScript 1 for server-side logic)	Java Servlets, JSP
Persistent storage and database support	Record Management System (RMS) and Perst Lite from mObject	Local database support is provided for SQL Server Mobile Edition; on the server side support provided for SQL Server	Persistent storage through ‘shared objects’; on the server side support for interaction with PHP scripts and usage of MySQL DB	Android APIs contain support for SQLite DB
Sound handling & supported formats	mp3 and whatever format is supported by the device	Supports only PCM-encoded files (uncompressed sound). Support for well-known formats (wav, mp3, etc) offered by third parties (e.g. Resco Audio for .NET CF)	Sound files embedded within the swf file. Supports mp3, aiff, au, wav, etc; no support for simultaneous playback of multiple-sounds	3gp, mp3, mp4
2D/3D graphics handling & supported formats	All MIDP versions support the displaying of rasterized images (in PNG format only) MIDP 3.0 adds support for GIF images. Support for SVG since MIDP 2.0 (JSR 226). Support for mobile 3D graphics on m3g format: Mobile 3D Graphics 1.0 (JSR-184) or Mobile 3D Graphics 2.0 (JSR-297).	BMP, JPG, GIF and PNG image formats are supported. Does not support SVG. Direct3D mobile applications available for Windows Mobile 5.0.	Graphics are vector-based (bitmap support is also included). Does not provide low-level 3D graphics API; possible to use a sequence of images exported from a 3D tool.	Supports PNG, JPG, and GIF. Does not support SVG. Supports 3D graphics via the OpenGL API.
Application development				
Learning curve	Moderate (developers need to familiarize with several APIs which are not part of the Java SE	Average (significant overlap with the .NET platform APIs)	Steep (reuse of the same ActionScript code)	Average (significant overlap with the Java SE platform APIs)

	platform)			
Developer Community base	Large community	Relatively large	Relatively large	Fair-sized and fast growing community
Debuggers availability	Excellent	Excellent	Good	Debugger integrated in Eclipse, Standalone debugging monitor also available
Cross-Platform Deployment	Execution on any device supporting CLDC/MIDP, yet, inconsistent implementations across vendors necessitating separate builds	Windows Mobile, Symbian-based devices (via third party tools)	Excellent (supported by Top 5 mobile manufacturers, best web compatibility)	Android only, because of Dalvik VM
Deployment Speed (packaging, installing, testing)	Slow (due to the fragmentation problem)	Relatively fast	Relatively fast	Relatively fast
Capabilities				
Functionality	Varies by handset - dependent on available included JSR's. No high-resolution pictures, No cell ID, limited file access	Limited audio support	No support for accessing native components	Touchscreen, accelerometer, GPS, cell ID, inter-application communication
Event Model	Event-handling mechanism based on 'Command' objects	GUI events bound to methods through multicast delegates	Uses the powerful ActionScript's event model (movie clip and object events)	Inherits the Java event model; uses a special class (Intent) to enable applications to respond to external events (e.g. phone call)
Phone Data Access	Varies by handset - dependent on available JSR 75, the PDA Optional Packages.	Full	None	Full

Runtime Speed	Average due to Java bytecode	Average due to CLR managed code	Below Average (interpreted language)	Average due to Java bytecode
Developer communities and market success				
Developer Community and Support	Extensive	MSDN	Extensive	Recent, growing
Market penetration	Extensive (also the basis of the Danger Sidekick Platform)	Average	Average	Based on the fact that it is supported by 34 major software, hardware and telecoms companies it has great potential to gain wide acceptance
Distribution and Licensing	None (Java Signed)	None (licensing may be provided by third-parties)	None	None
Development tools				
IDE available	Eclipse, NetBeans	Visual Studio .NET 2008, 2010	Adobe Flash CS4, Adobe Device Central	Eclipse, NetBeans (with Android plugin)
Emulator available	Free emulator, Sun Java Wireless Toolkit	Bundled with IDE	Bundled with IDE	Free emulator
Development Tool Cost	Free	Free (but only basic tools)	Varies: Free but limited with MTASC (Motion-Twin ActionScript 2 Compiler) IDE	Free

Table 3-1. Comparison of programming platforms with respect to software architecture and technical issues, application development, capabilities, developer communities & market success, development tools.

2.6. Current Trends

In the current state of practice, devices vary in terms of display size, processing and memory capabilities, supported runtimes and APIs, etc, making it almost impossible to write a single version of a mobile application that can run on a broad range of mobile devices available. Fragmentation increases the production effort in almost all aspects of the software life cycle, driving up the cost, lengthening the time-to-market and narrowing the target market. Measures such as better standardization (e.g., less optional APIs, more detailed specifications), stricter enforcing of the standards (e.g., using API verification initiatives, Technology Compatibility Kits) can help in this regard. Major players in the mobile application industry (such as platform vendors, device manufacturers, and operators) have a critical role to play in the war against fragmentation.

Java ME is undoubtedly the platform with the broadest deployment base and still maintains the largest market share, yet, it is the platform mostly affected by fragmentation, facing the possibility of being displaced by alternative platforms. Sun Microsystems has published techniques (set of guidelines) [164], aiming to reduce the practice of generating distinct executables for each phone. Some tools for resolving Java ME device fragmentation (e.g. NetBeans Mobility Pack for CLDC) are already available but there is still a long way to go. Along the same line, the Mobile Services Architecture (MSA) has emerged as an industry standard that aims to reduce fragmentation and provide a consistent Java ME platform for developers to target. In addition to specifying what component JSRs must be present on a compliant device, the MSA also clarifies behavioural requirements in order to improve the predictability and interoperability of the JSRs. The MSA defines two stacks: a full stack that comprises 16 JSRs (JSR 249), and a subset of eight JSRs (JSR 248). JSR 248 is being pushed ahead of JSR 249 so developers can make the earliest possible start on MSA-compliant applications that will run on the highest-volume mobile devices. JSR 248 has recently been approved, yet, its adoption by OEMs remains to be proved.

The competitiveness of Java ME against platforms that target graphics-heavy applications (e.g. Flash Lite) will also largely depend on the provision of technologies that will enable the production of expressive, feature-rich content to mobile devices. Along this line Sun Microsystems has recently released JavaFX Mobile [84], a new platform and language with Rich Internet Applications (RIA)-friendly features, including a declarative syntax of the JavaFX Script language for GUI development. JavaFX Mobile allows developers to build expressive interfaces while re-using existing back-end Java code; namely, it allows various team members with no programming experience (such as designers and graphic artists) to create graphics-intensive front-ends for mobile applications. However, JavaFX mobile success will mainly be decided by the support offered by OEM, i.e. the integration of its binaries and runtime on mobile handsets.

.NET CF will probably maintain its developers' base as long as Windows handhelds remain in the picture, as it comprises a powerful platform for programming and accessing native components of Windows-compatible PDAs and smartphones. However, its market share is not likely to increase since its porting to popular phone OSs is cumbersome (it requires the implementation of a 'Platform Adaptation Engine' as an interface between the CLR and the OS) [60].

The release history of Flash Lite indicates that Adobe has concentrated more on multimedia support than defining a powerful API for developing applications with rich functionality. Despite the effort to establish Flash Lite as a gaming platform, it lacks APIs or classes specifically targeting game development (e.g. Flash Lite 3.0 does not support the BitmapData object which is part of Flash 8 and really useful for game development) and needs to improve its sound handling. Furthermore, comparative studies indicate that Flash Lite exhibits lower performance and frame rate, while it is more memory consuming than Java ME [163]. On the other hand, Flash Lite appears as a natural choice for designing user interfaces and graphically rich applications. In that sense, it allows designers into mobile development space. A promising evolution path for Flash Lite seems to be its synergy with different application platforms, bringing together the best of diverse worlds. Recently, the Capuchin Project [139] defined a Java ME API as a bridge between the Java ME and Flash Lite enabling the use of the latter as the front-end and the former as the back-end of applications;

namely, Flash tools may be used for GUI design while still having access to all the phone services available to Java ME.

Android has received an enthusiastic welcoming among manufacturers and developers, however, some handset manufacturers are taking longer than expected to integrate Android. Hence, its market share is not growing as rapidly as anticipated; yet, it seems to expand its developers' community -mainly- against Java ME. The future of Android will also largely depend on providing technologies for simplifying the design of multimedia-rich applications (Sun Microsystems announced that JavaFX Mobile will be available on Android OS) and -most importantly- its success in confronting the porting (fragmentation) problems; the latter is too early to estimate, given Android's narrow installation base.

Being a relatively young software platform, Android is struggling with a small number of available applications; therefore, Google has spent funds to attract developers and prepare a critical mass of applications before the first Android phone release. Taking advantage of an opportunity to run a large number of existing Java ME applications may also add value to Android. Along this line, porting services are provided to convert existing Java ME titles to the Android platform (e.g. by Tira Wireless and J2ME Polish).

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

2.7. Requirements for Cultural Application Development Platforms

It is evident, that the application development platforms reviewed in Section 2.4 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 cultural applications for small devices. The set of requirements presented herein have been identified through the experiences gained during the implementation of case-studies at our laboratory, as well as the compilation of the results from usability tests performed upon them. These exercises indicated that the requirements identified in this section are not unique to cultural applications alone. This means that these results can also be generalized to other types of applications for small devices. These requirements are listed below:

- rapidity of application's development and deployment (rapid development implies minimal implementation effort and cost)
- 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 usable 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)
- restrictions 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
- 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.

The above synopsis shows 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:

- 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

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

2.8. Selection of Mobile Application Platform

Nowadays, the mobile market is fragmented and under constant changing. When this research commenced in 2006, there where only a few development platforms options to choose from, the Symbian platform possessed the largest market share and only a few smart phones with Windows Mobile were in offer. Choices were confined among the Java ME platform, s60, and .Net CF that were readily available. The Android platform was not yet there, while the Windows mobile .Net platform did not posses a significant market share. It was noted that around 80% of all mobile handsets shipped in 2008 supported the Java ME standard [101]. Even today, Java ME still represents the most widely distributed application environment to choose for porting the mobile content and is supported by the largest percentage of operating systems. Figure 2- depicts the current state of mobile phone operating systems. Symbian and RIM amongst others fully support Java ME applications. Assuming some incremental learning, an average Java developer could quickly master coding conventions from smart cards all the way to high powered devices like settop boxes and high end PDAs. As a result, the choice of Java ME (still J2ME back in the 2006 days) as the main development platform (used to implement the prototypes described later in this thesis) has been a straightforward decision.

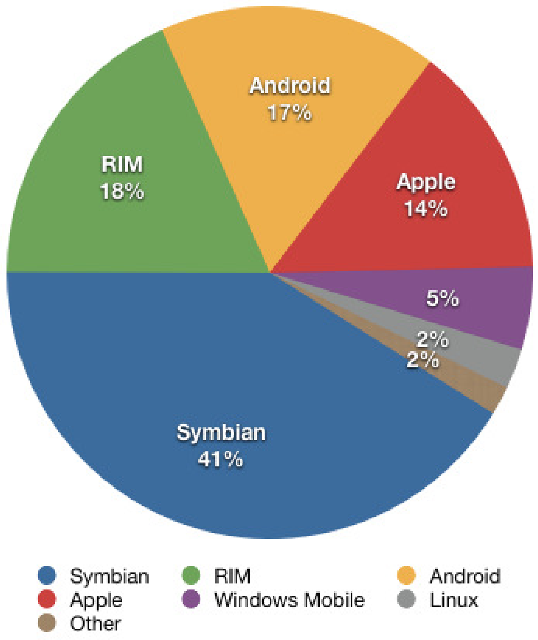


Figure 2-6. Share of 2010 Q2 smartphone sales to end users by operating system, according to Gartner

One of the main reasons of our decision still holds today and is mainly related with the model we chose to research. That is, the use of thin client technology (i.e. web browsers) used to service web users and the use of custom thick clients (i.e. mobile Java application ported to various mobile handsets) to service mobile device users. An example of the above model can be seen in the Google maps platform [98]. On the web, the Google map application is based on HTML and Ajax technology [75]. However, when accessing the web map application via a mobile browser the user is prompted to install a Gmaps Java ME application [61]. This is because client-side technology (e.g. Jacascript) can not be completely supported by mini web browsers [28], hence Google has developed a standalone application to be installed on the mobile device side, mostly using Java ME. Taking this model a bit further, the research workplan involved the dynamic creation of custom mobile applications on the fly, offering a customised solution for each tourist user separately (see Section 4).

2.9. Conclusions

This chapter reviewed state-of-the-art development application platforms to target developing for mobile applications that enhance the visiting experience in cultural organizations and support tourists travelling experience. Based on this review it suggests authoring tools requirements for developing cultural applications on mobile devices based on user, application and designer needs.

Specifically it has been concluded that the choice of the appropriate 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 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.

Parts of this chapter were published in the ACM Mobile Computing and Communications Review (ACM M2CR) journal, the Journal of Mobile Multimedia and a PCI'2007 conference paper (see Appendix A).

CHAPTER 3

THE LANDSCAPE OF MOBILE TOURIST GUIDE PROJECTS

3.1. Introduction

This chapter focuses on the evaluation of research and commercial applications used by tourists to retrieve information, navigation, and guidance using some form of mobile devices. For this evaluation, a large number of relevant projects have been investigated addressing a number of different issues. Our main objective is to extract a set of principles for designing mobile tourist applications, which will be used in the design process of our mobile application discussed later in this thesis. Yet, to obtain design principles for such diverse platform environments, the identification of a set of design criteria is first required. Focus has been given to mobile tourist guides systems running on any hardware architecture with or without a network connection medium.

The remainder of this chapter is organized as follows: Section 3.2 analyzes the design criteria sought out to complete this evaluation. Section 3.3 presents the actual evaluation with respect to those criteria and includes some tables of summarizing data. Section 3.4 comprises a summary of research findings. Section 3.5 below discusses the extracted design principles and concludes this chapter.

3.2. Design Criteria

In the scope of mobile tourist guides, the research carried out over the past decade falls into two main categories [155]: application-led research and technology-led research. Application-led being research led by a domain problem which is evaluated by deploying a solution and quantifying the benefits of this solution and technology based research is motivated by the benefits of the solution, yet challenging technologically wise. In turn, both of these categories have arisen from usability designers and by device technology designers [106].

Until today, evaluation research of mobile guides has mostly been presented by scope of issue [96], [110], [33]. Kray et al. [96] studied map-based navigational guides evaluating guides based upon five basic issues: features offered situational factors, adaptation capabilities, user interaction and architecture. Others like Chen and Kotz [110] took into consideration the issue of context awareness to evaluate mobile guides. As such, our evaluation attempts to address two main questions; what design principles can be used by application-designers for the design of mobile tourist guides; what technological choices do developers have while embarking on this specific domain area. This evaluation tried to answer two fundamental questions bringing about a new sub-set of evaluation criteria which took into consideration our vision for the creation of a nomadic[6] tourist information platform running on readily available mobile technology.

The accumulated findings of the design criteria from both the angle of application designers and technology developers helped to compile our main evaluation criteria which are summarized below in the form of questions:

In regards to application-designers:

[6] The term 'Nomadic Tourist Information Platform' refers to a platform providing access to tourist information and to personal space anytime anywhere.

- What **information models** were developed for mobile guides; do they make use of personal profiling and/or collaborative filtering techniques to offer personalized information and services; could the information model be updated easily; could it support different languages?

- What types of **input/output modalities** were used? Did the projects offer various types of information using multimodality technologies such as 3D graphics or speech?

- What **unique services** were designed and how were they implemented (e.g. using web agents, web services, etc); were these services well accepted by tourists? Did the projects integrate any existing standards-based frameworks or initiatives to support tourist users or were all services propriety?

With respect to technology developers:

- What **architecture** was used; which technology platform was chosen to implement the applications in stake; could these be used in today's mass mobile technology devices?

- What types of network infrastructure was required to support the project on hand; what network infrastructure was used (e.g. WiFi, BT, and 3G); could the application adapt to changing networking environments? What was usage cost of such systems for the end-users?

- What type of positioning technologies and map technologies were used to support indoor and outdoor use; were maps used to support the user; could they be used to support route finding, dynamic itinerary support to users? What types of location-based and context-aware services were offered? How did navigational technology support the user context with respect to information published to them?

In enhancing the above stated questions, five principle groups where extracted and are summarised in Figure 3-1 below. All the projects reviewed in section 3.3 are reviewed in respect to the aforementioned principle groups.

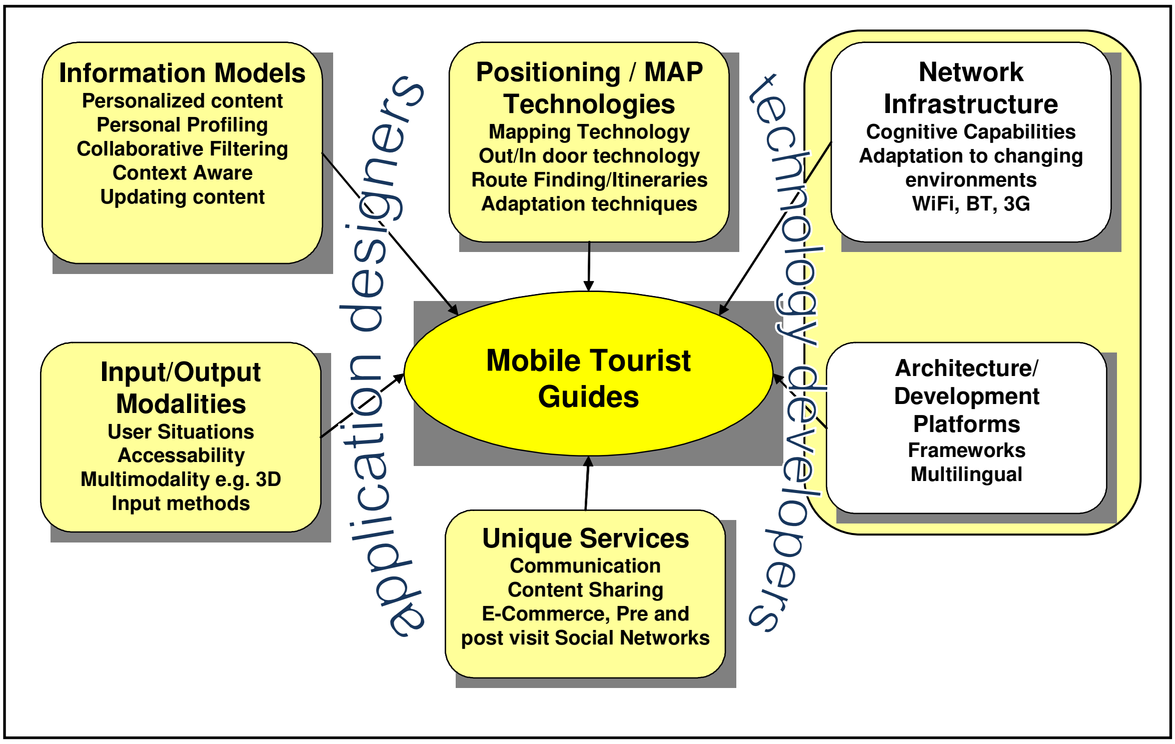


Figure 3-1. Evaluation criteria used to review mobile guide projects.

3.3. *The Evaluation*

The field of mobile tourism has only been around the past decade or so, yet it includes a large number of research and commercial applications database. The majority of related projects addressed are in the form of web sites, web applications and mobile guide applications all addressing ubiquitous mobile tourism solutions. The projects investigated were classified into four groups (see Figure 3-): mobile guides, navigational assistants, web-to-mobile applications and mobile web-based.

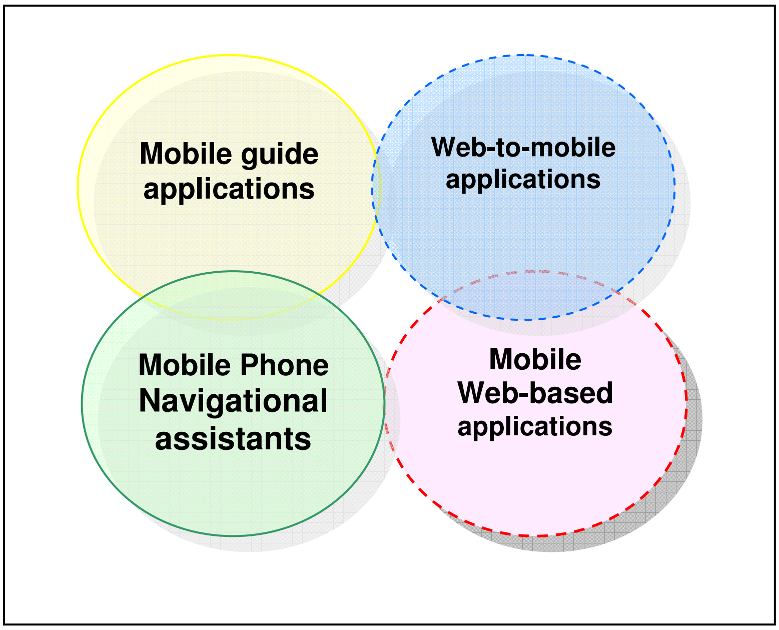
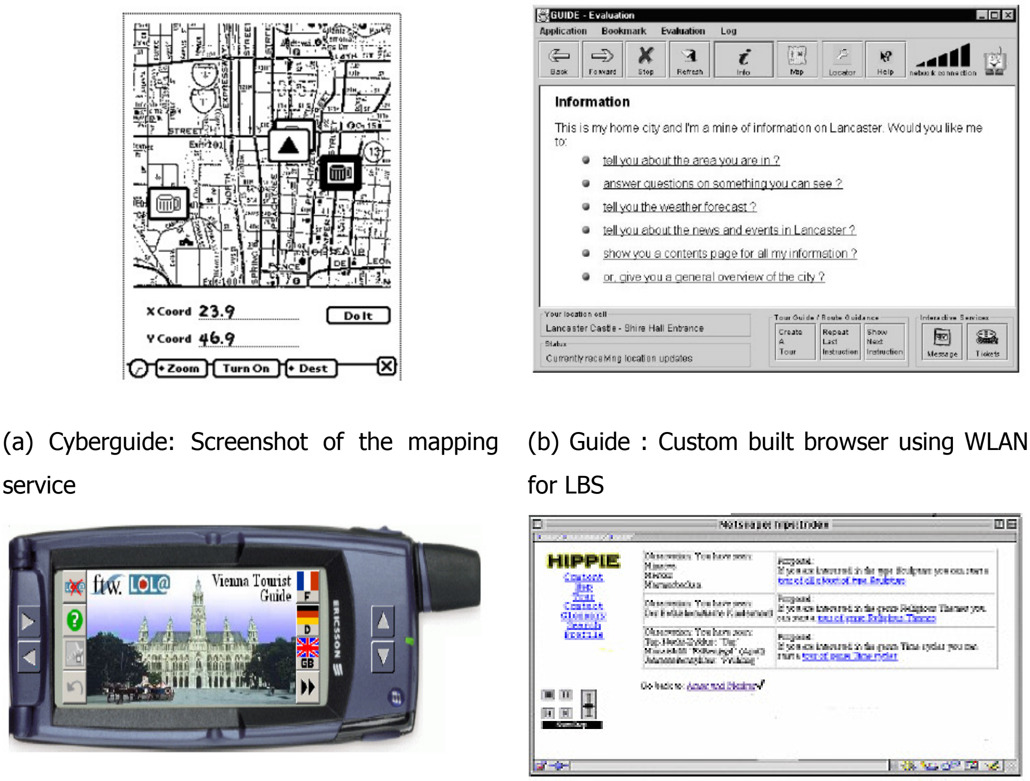

Mobile guide applications
Web-to-mobile applications
Mobile Phone Navigational assistants
Mobile Web-based applications

Figure 3-1. Projects classified into 4 groups

Mobile guides applications are projects that use mobile devices as the key user platform offering tourist information and the use of services in various forms.

The majority of the overall projects that were evaluated were thick applications running as stand-alone applications or in a networked-centralized application mode. One of the original milestone mobile guide projects was Cyberguide [1]. Due to the incompatibility of mobile devices; many different application development platforms were used. The main goal of the Cyberguide project (see Figure 3-) was to support rapid prototyping [105] resulting into separate systems which were prototyped for outdoor and indoor use. The Guide system [42] was a mobile tourist guide project implemented for the city of Lancaster. The project was designed to be flexible, to enable visitors to explore and learn about the city without the need to follow guided tours (see Figure 3-b). The Local Location Assistant (LoL@) [168] was a research project which investigated location-based multimedia Universal Mobile Telecommunications System (UMTS) applications (See Figure 3-c). The main idea behind the usage scenario of LoL@ was being able to access tourist information via tourist mobile phones without renting a project-specific device. HIPS (Hyper Interaction within Physical spaces) [134] [133] was a hypermedia-based guide application (see Figure 3-d) designed to offer support in the multiple stages of a museum visit: Preparation at home (pre-visit), execution on site (on-site) and evaluation process (post-visit).



Information

This is my home city and I'm a mine of information on Lancaster. Would you like me to:

- tell you about the area you are in ?
- answer questions on something you can see ?
- tell you the weather forecast ?
- tell you about the news and events in Lancaster ?
- show you a contents page for all my information ?
- or, give you a general overview of the city ?

Your location cell: Lancaster Castle - Shire Hall Entrance

Status: Currently receiving location updates

Tour Guide / Route Guidance: Create A Tour | Repeat Last Instruction | Show Next Instruction

Interactive Services: Message | Tickets

(a) Cyberguide: Screenshot of the mapping service

(b) Guide : Custom built browser using WLAN for LBS

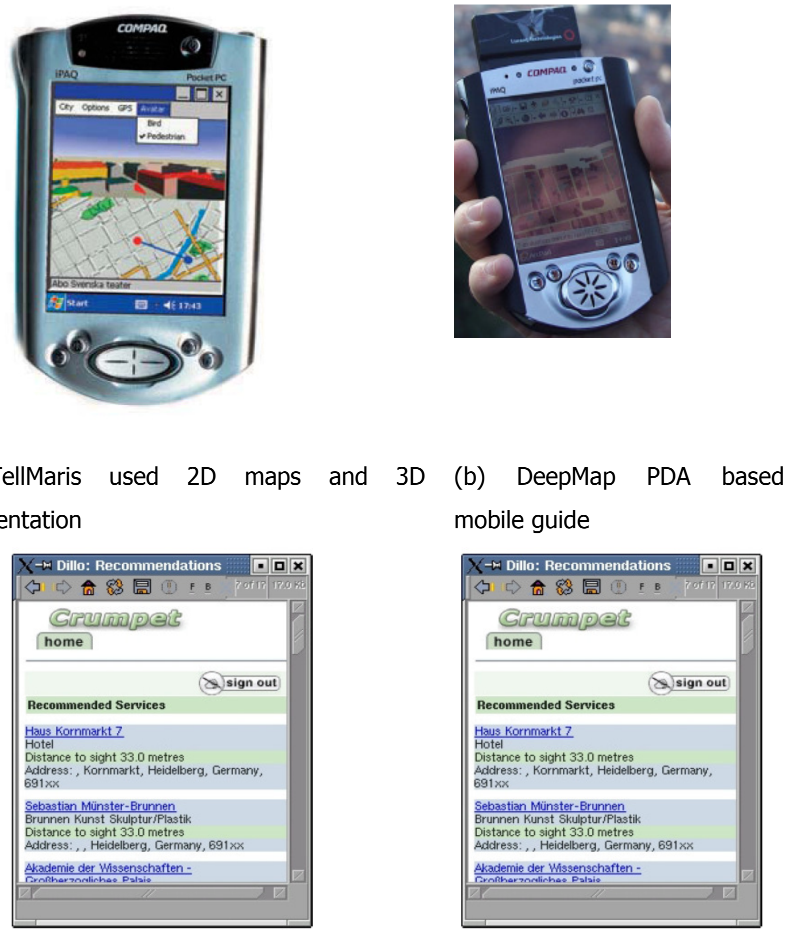
(c) LoL@ guide running on a UMTS mobile phone

(d) Hippie browser based user interface

Figure 3-2. Various screenshots of the mobile guide projects.

TellMaris, a Nokia Research centre prototype [98], [151], was one of the first mobile systems to use OpenGL-based 3D maps prototype in combination with 2D maps (see Figure 3-a) for the city of Tonsberg in Norway targeting boating tourists in the Baltic Sea area. The project presented 2D and 3D maps on mobile devices in a way in which to provide easier orientation for tourist [143]. The DeepMap project [107] was a research framework conducted by the European Media Lab and several cooperating institutions that envisioned the future of tourist guidance systems that worked as mobile guides and as a web-based planning tool (see Figure 3-b). The CRUMPET project [141], [153] implemented, validated, and tested tourism related value-added services for nomadic users across mobile and fixed networks (see Figure 3-c). SmartKom

[182] was a multimodal dialogue project which combined speech, gesture and facial expressions for both input and output (see Figure 3-d). REAL [15] was a hybrid (combination of client-server with application-based architecture) pedestrian navigation system, which helped the user to find information by generating a graphical route description. The REAL project developed a pedestrian navigation system that combined active and passive location sensitivity in such a way that the change-over between both adaptation paradigms was barely noticeable for the user.



(a) TellMaris used 2D maps and 3D representation

(b) DeepMap PDA based mobile guide

Recommended Services

Haus Kornmarkt 7
Hotel
Distance to sight 33.0 metres
Address: , Kornmarkt, Heidelberg, Germany, 691xx

Sebastian Münster-Brunnen
Brunnen Kunst Skulptur/Plastik
Distance to sight 33.0 metres
Address: , , Heidelberg, Germany, 691xx

Akademie der Wissenschaften - Großherzogliches Palais

Recommended Services

Haus Kornmarkt 7
Hotel
Distance to sight 33.0 metres
Address: , Kornmarkt, Heidelberg, Germany, 691xx

Sebastian Münster-Brunnen
Brunnen Kunst Skulptur/Plastik
Distance to sight 33.0 metres
Address: , , Heidelberg, Germany, 691xx

Akademie der Wissenschaften - Großherzogliches Palais

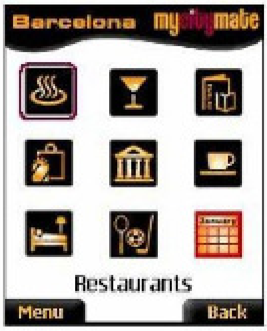
(c) Crumpet PDA browser based mobile application

(d) SmartKom used mapping technologies

Figure 3-3. Screenshots of PDA-based mobile guides.

The web-to-mobile projects are projects that use the web to offer tourist information and services to tourists by deploying a mobile application to the user's device. There are quite a few projects that use the web to deploy mobile applications to its users. A popular web-to-mobile application is the

Google maps [61] application, due to the incompatibility of mobile web browsers, when a user opens the Google maps website via a mobile phone, instead of opening the map in the users' browser; it redirects them to a webpage where the user can download the Google map Java ME [62] application. This is the case for G-mail and for YouTube.com, when a user accesses these pages instead of prompting them to download a thick mobile application the mobile sites state the benefits of their mobile applications in respect to the mobile websites, allowing for users to decide to download the mobile application or to use the web application as a limited mobile web application. In the scope of tourism web applications, there are not that many web-to-mobile projects that allow mobile applications to be customised online then built and downloaded to the mobile phone. The Mycitymate [120] is a web-to-mobile project (see Figure 3-a) providing information namely of city locations like venues, café, pubs, bars, accommodation etc, but also offering personalised social features like where are my friends, make new friends etc. The system has a Web interface for the pre-visit stage where users select content and then can build their customised mobile phone guide application to download and install on a mobile phone.



MyCityMate main user interface

Figure 3-4. Screenshots of mobile phone guides.

As for navigational assistants, these are classified as mobile applications using a map as the basic user interface, offering routing and guidance services to tourists through the use of Points of Interest (POI) displaying specific tourist information. A large number of off-the-shelf commercial navigational applications were investigated but only the Nokia maps application [53] was incorporated. This is because the other applications did not entirely target tourist end users nor did they have an in built tourist guide application as an option.

Nokia Maps (to date now called Nokia OVI maps [135]) is a navigational mobile application found on most new Nokia phones or is readily available to download from the official Nokia website. This application compromises of mapping and navigation services which require either built-in GPS circuit or an external Bluetooth GPS receiver, although the application works in simulation mode for those devices which do not have a GPS option. It was stated as having maps for more than 150 countries, and with 15 million points of interest (POI) pre-loaded as such offering navigational and route planning features [149]. This application gives the tourist the ability to enhance the Navigational assistant using -the Nokia map application- electronic guides (see Figure 3-) from traditional tourist guide companies like Berlitz™, Insight Guides™, Lonely Planet™ etc, offering photos, video, audio commentary, and informed coverage on places of interest to tourists [53].

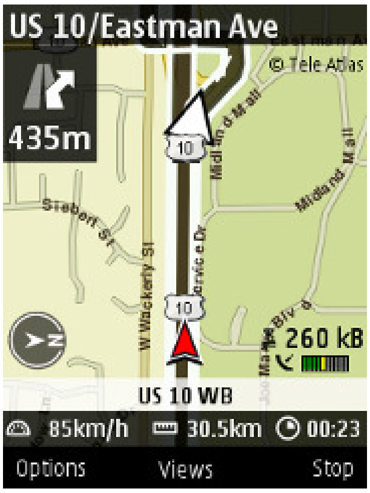


Figure 3-5. Screen shot of a mobile navigational mapping application (Nokia Maps mobile phone based navigational tourist guide)

The mobile web-based applications refer to mobile tourist (XHTML Mobile) portals which offer tourist information to mobile device browsers through a client-server HTTP interaction (e.g. [119], [49], [51] etc). Practically, these applications do not differentiate from traditional 'desktop' web applications, as they treat mobile devices like thin (web) clients; hence, they are not thoroughly reviewed in this article.

The three out of the four groups of projects were included in this research resulting to an overall evaluation of mobile tourist guides whereby, design principles extracted are summarized. The Mobile Web-based applications have not been extensively evaluated due to their resemblance with their web counterparts. Following is the summarized evaluation ordered by design criteria.

3.3.1. Information Models

In the included evaluated projects, the information models which were used varied from project to project. Table 3-1 summarizes all the information models which are reviewed in this section. Some projects used a centralized hypermedia model whereby a browser-based application is used to browse through hypermedia content. Others used a distributed dynamic information model where the network would push information to the users' mobile device upon entering predefined network vicinity. In addition, other projects used a decentralized on-device storage facility system allowing for users to have unlimited connectivity to the information. This however made it difficult to update the content on each individual device. Notably, the majority of projects used a centralized approach, i.e. a connection of some sort to feed information to networked mobile devices, which meant costly wireless metropolitan installations or cost incurred for tourists using such connections directly from a mobile phone device. Apart from the information models, this section also reviews projects that make use of personal profiles, context-aware systems and discusses the support of multiple languages.

The Cyberguide project used a centralized hypermedia information model basing information on maps. The system was conceptually divided into four independent components: the Cartographer; which offered its users knowledge of their physical surroundings through the use of maps, the

Librarian; which provided access to sight information, the Navigator; which provided navigational information and the Messenger; which offered communication services for tourists to communicate with sight staff and for the system to communicate with visitors or groups of visitors. The GUIDE project provided multi-lingual information and was based upon a distributed and dynamic information model [42]. It extended traditional hypermedia models [34] offering information based on Personal Context. The project named it ‘dynamic information serving’, informing users of sudden change of sight operating times, if sights are closed, the status of ticket queues etc. The personal context involved the use of a personal profile, in which the system explicitly prompted the visitors to complete an entry level survey of their personal information, getting information based upon age, technical background and the preferred language and also involved the use of environment context, e.g. the time of day, the opening times of the attractions, etc. The content was adaptable with respect to, what the visitors had already seen, e.g. welcomed back visitors if re-visited a site.

The LOL@ project used a hierarchical approach to model information enabling a centralized browser metaphor of hypertext links, linking to text and multimedia information [142]. The content was based on XML/XSL using flexible templates and the multimedia data was accessed using the browser’s functionality. The Hippie project also used a centralized information model and was based on context sensitive models which apart from location, positioning and direction also used an adaptable personal profile, determined by explicitly and implicitly logging the users’ preferences which resulted to a context-aware system. The HIPS project was based on an Adaptive Hypermedia information model which provided information about the exhibits through a hierarchical method using dynamically created Web pages having additional knowledge of the user model and interests, e.g. each time an object was visited, it was marked by the system so as to adapt information provision. The user model was one of the key strengths of the HIPS project: by evaluating user interactions and physical navigation, recommended tours of exhibits were offered to its users. The TellMaris project was based on a centralized client/server application model, which meant all data was downloaded upon request via a wireless communications network connection. In the mobile version, the system displayed both 2D and 3D maps.

The core of DeepMap was the centralized geographical information system (GIS) and other databases. The GIS database stored spatial data while there was a database to store temporal data (i.e. historical information of sights) and a separate database to store topological information such as user information and general information about places (i.e. restaurants, cafes, shops). The information model was quite complex as it was connected to the spatial and the other databases. The CRUMPET project also integrated GIS as a means to integrate large volumes of geographical data. This meant that the project offered information on topics such as personal tours, navigational assistance and route finding. Map adaptation examples include culture specific map colouring, map generalization, user-orientation dependent maps, focus maps and personalization. As of personalized services, this project argued that the solution to the problems associated to mobile devices (such as restricted screen size, input methods, network capacity) could lie within the adaptation of personalized information and services for nomadic users i.e. view the same information on any device. In CRUMPET, the adaptation of such services resided on the notion of filtering based on a user profile which was gathered by getting information of interests, abilities and characteristics of the user.

The MyCityMate mobile system used a decentralised hierarchical information model based on XML documents and user menus. The MyCityMate system was one of the first systems using the web-to-mobile dynamic application generation technology to mass deploy applications to users mobile phones. The MyCityMate system used the web platform to attract users to choose tourist content of personal interest or automatically generate selection using an explicit personal profile system. The system would then deploy an application to be downloaded to the users' mobile phone to run on an adaptable standalone mode or connected mode. The

Mytilene guide used a similar propriety XML information model but also used maps to show location of a specific sight and incorporated multimedia tours in the mobile application.

	On device storage	Centralised hypermedia model	Propriety application model	Personal Profile	Context-awareness
Cyberguide	✓				✓
GUIDE		✓		✓	
LoL@		✓			
Hippie/HIPS		✓		✓	✓
TellMaris			✓		
Deep Map			✓		✓
CRUMPET		✓		✓	✓
SmartKom		✓			
REAL			✓	✓	✓
MyCityMate	✓		✓		
TIP					
Nokia Maps	✓		✓		

Table 3-1. Summary of information models

3.3.2. Position and Map technologies

The use of maps and positioning technologies prevailed in most of the projects reviewed. Table 3-2 summarizes all position and map technologies stated in this section. Some systems where somewhat simplistic using raster image maps where as a number of systems used GIS running a client/server functionality model. This meant that all GIS data was stored on a central server and when needed was rendered to be viewed on the mobile device. This method surely needs a network connection to run, but allows for heavy calculations to be carried out on the server side and not on mobile devices. In the same way a mobile network connection is needed to extract routing information. Some projects used sophisticated routing systems to map navigational paths for users whereas other used maps to show an area of interest. The use of positioning technologies varied by indoor outdoor use; most projects used GPS systems outdoors and IrDA /RFID /Bluetooth technologies indoors. Some projects offered user position adaptation which allowed for users to manually calibrate their position directly from the map.

The Cyberguide project used image maps for indoor situations, using IrDA positioning technologies and vector maps stored and run on-device using GPS positioning technology outdoors. The Cyberguide project also used a logging system to track user sight visits. The GUIDE system incorporated maps to allow visitors to use in specific situations, but did not use them as a means of route finding or did not use the installed WLAN technology for showing positioning on the map, yet showed position by using the identity of the wireless hotspot. The LOL@ used maps as guidance and proposed an adaptable user positioning system using GPS but also using a mobile networks cell-id positioning system. This system also offered route finding functionality i.e. shortest path, closest route and used a manual logging system as a means for the user to capture the sights visited in order to create a user diary log later. The Hippie/HIPS project offered guidance not using maps but using IrDA positioning sensors at room level and at object level and also incorporated an electronic compass to proactively notify users of upcoming exhibits. When a user visited a sight this was automatically logged and when revisited different content information was provided to its user.

The TellMaris project used both 2D and 3D maps as a navigational aid for the city of Tonsberg, Norway. The use of such technology allowed for routing functionality using a client/server infrastructure and GPS positioning in order to generate a map on the server-side using geospatial representation as input and to portray this information on the end user client-side device. The DeepMap and SmartKom projects also used GIS server side technology to generate the 2D maps including user adaptable GPS positioning as well as route finding features. The CRUMPET project took the Deepmap project one step further incorporating GIS maps as guidance and route finding features using personal profiles and an agent based recommendation system also. The Mycitymate used a Google maps mash up application to show POI's and used GPS for location tracking. The Mytilene city guide used raster map to show the specific POI on the map. No route finding or positioning technologies were stated as being used. The Nokia map mobile application used maps stored on the mobile device. The user could track POI on the device using GPS positioning. No means of adaptation was available in the Nokia maps application and the route finding capabilities was calculated on the server and later returned and showed on the end use mobile device.

	Map	Outdoor Positioning	Indoor Positioning	User Adaptable Positioning
Cyberguide	✓	GPS	IRDA	
GUIDE	✓	WLAN		
LoL@	✓	GPS / Cell ID		✓
Hippie/HIPS			Electronic Compass / IRDA	
TellMaris	✓	GPS		
Deep Map	✓			✓
CRUMPET	✓			
SmartKom	✓	GPS		✓
REAL	✓	GPS	IRDA	
MyCityMate	✓	✓		
TIP				
Nokia Maps	✓	✓		✓

Table 3-2. Summary of positioning and mapping technologies

3.3.3. Architecture/ Network Infrastructure

The architecture and network infrastructure of the reviewed projects also varied. As such, a trend to use generic mobile applications and customized user interfaces was seen as enhancing user experience. These mobile applications needed a specific hardware platform to run, incurring fragmentation problems which arose to newly available devices with new operating systems. The projects basically could be split up to four main platform groups: the Personal computer platform, the Personal Digital Assistant (PDA) platform, the Java mobile platform and the browser technology based platform. Only applications which used native browser technology could claim platform independence. The network technology used mirrored available infrastructure of the projects and to the availability of such network infrastructure. Emphasis was given to projects which were cognitive resource adaptable i.e. the software adapts seamlessly to a change in the network environment. Table 3-3 shows a summary of all architecture and network infrastructure stated in this section.

Cyberguide used the Microsoft Visual Basic runtime system running on portable device technology. The portable devices used a WLAN infrastructure installed on the campus university test site. No cognitive resource adaptation was mentioned for this project. The technological infrastructure varied because of the different prototype projects implemented. The system used a commercial PDA yet at some stage the project stated the use the Apple MessagePad 100 with Newton 1.3. The indoor version of the system was tested using IrDA sensors for location tracking. The prototype was also tested on the Dauphin DTR-1 palmtop. The GUIDE system used specific mobile devices, namely Fujitsu TeamPad 7600 portable PC empowered by custom Java applications. The Java application included a Hot Java browser [162] and also included gauges to show signal strength, an indication to which WLAN hotspot the user was connected and another indication when the mobile guide was downloading information. This project did not have a system of cognitive resource adaptation and it could only run using the WLAN as it was also its means for user position finding. The project was adaptable to failing connections where a caching system would download and store content on the device at earlier stages allowing for the device to gracefully degrade.

The LoL@ system was based on conventional Internet software technology and user interface paradigms, extended by concepts to improve usability for the mobile domain. This application targeted high end mobile phones and smart phones with Java Applet enabled fully fledged Web browsers and used touch screens as input. The mobile terminal used applet technologies, while Java Servlets technology was employed on the server side. A network-centric 3-tier application architecture was chosen for this implementation enhanced with telecommunication specific constant network connection using UMTS (3G) or GPRS network via a mobile phone. The fact that the LoL@ project required constant network connectivity where all content data was stored on a server database and prepared on demand resulted to data intensive costs for the user which was observed as a disadvantage during user tests, visitors were reluctant to use the system considering the high roaming fees [168]. No adaptation capabilities were designed for failing network shortages. The clients of the HIPS system were PDAs which used a thin client (Web browser) application with client-server architecture requiring a stable network connection to operate. The PDA devices were called HIPPIES and were connected via a wireless LAN (WLAN) infrastructure. Apart from the use of PDA devices, the project stated the use of notebooks or PCs to access web content. The TellMaris project was based on a client/server model, which meant all data was downloaded upon request via a wireless communications network connection. The OpenGL system was developed on Linux, Windows and Windows CE platforms; however the system also ran on the Nokia communicator 92XX. In the mobile version, the system displayed both 2D and 3D maps simultaneously; however, the user could choose the desired type of map. No reference was found for support of cognitive network resource adaptation. The system was implemented on 4 main databases using agents to access the required information. These databases were accessed using four main agents: The database agent to retrieve non-spatial information; the geo-spatial database to retrieve spatial information and to calculate geo-spatial information (i.e. place with regards to user location); the route agent used to compute and manage routes; the map agent which generated at first raster maps as a picture case and later as vector maps used to display features on the maps. The Deep map system architecture was based on the agent-oriented software paradigm which allowed reusability of various system components. The prototype was implemented using two technologies; one of a belt worn Xybernaut mobile assistant IV having a visual output on a flat touch screen mounted on the arm and the other a laptop PC placed in the user's backpack.

The CRUMPET implementation was based on a standards-compliant open source agent framework, extended to support nomadic applications, devices and networks. The system was built using a 3-tier structure; with mobile clients and user services on the two ends and the use of multi-agent systems in between both. At the stage of usability testing a PDA was used as the client device using GPS as positioning technology. Even so, it was argued that any mobile device able to display rendered maps and simple HTML pages could be used. The system could use all types of networks that a tourist might be exposed to, i.e. WLAN, GSM, GPRS, UMTS. The REAL project developed a pedestrian navigation system that combined active and passive location sensitivity in such a way that the change-over between both adaptation paradigms was barely noticeable for the user. The REAL project used both PalmOS and Pocket PC platform for indoor and used a SONY VAIO notebook outdoor for computational power but for graphical and textual presentations a special clip-on for glasses from MicroOptical was used in conjunction with a customized Garmin GPS unit as a pointing device. For both indoor and outdoor systems the 2D and 3D graphics were generated via the embedded Cortona VRML1-browser. The SmartKom used distributed component architecture using an agent-based multi-blackboard system. The integration platform was called MULTIPLATFORM (Multiple Language Target Integration Platform for Modules), built on top of open-source software, making it open, flexible and scalable able to integrate heterogeneous software modules implemented in diverse programming languages and running on different operating systems. SmartKom modules were coded in C, C++, Java, and Prolog. SmartKom supported dynamic multi-lingual interaction by introducing a semantic layer that encoded interactions in a language-independent way.

The MyCitymate and the Mytilene guide systems included a mobile application developed on the top of the Java ME Platform [163], essentially comprising a certified collection of Java APIs for the development of software for small, resource-constrained devices such as cell phones and PDAs etc. These two projects are purposely custom built 'stand alone' applications not needing a network connection to operate. 'Tourists' incorporate people visiting international locations where roaming charges apply and not needing a network connection to operate could be critical in choosing a mobile tourist application. However, just like the Nokia navigational assistant if needed, many features are available if users want to use a network connection. Features such as 'where are my friends' in the MyCitymate project or the 'download more information' in the Mytilene guide project.

The Nokia Navigational project has been developed using Nokia Symbian platform supporting all s60 3rd edition phones and other Nokia platforms. The Nokia maps also has a free pc based application where users upon installation to their pc can download extra maps and voice navigation files instead of downloading them straight to the user's phone.

	Application programming environment	Offline use	WLAN	GPRS, UMTS, GSM HSDPA	Resource Adaptive
Cyberguide	Visual basic runtime		✓		
GUIDE	Java portable PC	✓	✓		
LoL@	Java Applet			✓	
Hippie/HIPS	Microsoft .net		✓		
TellMaris	Windows CE PDA Nokia s60		✓	✓	
Deep Map	Standards compliant open source agent		✓	✓	
CRUMPET	Standards compliant open source agent		✓	✓	✓
SmartKom	Agent architecture			✓	
REAL	PalmOS/Pocket PC		✓	✓	✓
MyCityMate	Java ME application based system	✓	✓	✓	
TIP			✓	✓	
Nokia Maps	Symbian S60 3rd edition platform	✓	✓	✓	

Table 3-3. Summary of Architecture/ Network Infrastructure used in stated projects

3.3.4. Input/Output Modalities

At first glance, the use of input/output technologies in mobile devices seemed to be trivial and should be incorporated with the system architecture review criteria. However, in the context of this chapter which is an in depth review of related work, the input/output modalities was considered as separate criteria for all projects reviewed. In this section the projects are reviewed with respect to input technologies used by the users of the mobile devices also as to output technologies. In addition the projects are evaluated in regards to accessibility and their ability to seamlessly internationalize the specifically targeted tourist applications.

The mobile device used in Cyberguide was a standard PDA using a pen input and a standard PDA screen as an output. In its documentation there was no reference to multi lingual support or use of any other modality technology to support other forms of output. The GUIDE project employed devices with larger input area for use with an input pen and a transreflective screen for use in direct sunlight. The GUIDE project stated multilingual support in its documentation but that was not referenced extensively. The LOL@ project used speech as a form of input. Tourist used speech to input control to the hierarchical interface gaining access to menu items. This project allowed speech access in three different languages English, German, French using not only one-word commands but also natural language input phrases to control the menu. This system did not use natural language processing other than menu control for input purposes. The system used a VoIP architecture solution based on the Session Initiation Protocol (SIP) [181] and the GSM voice codec [71].

The Hippie/HIPS system was a pen input based system not offering multilingual support to its users. Standard pen input and PDA screen output was also used in the TellMaris project, along with 3D representation on 2D maps in an augmented reality environment to depict buildings of historic nature (including buildings that did not exist anymore). The DeepMap project integrated natural language processing to mobile devices. The Deepmap user could gain access to information by queering the system using natural language. The system upon queering the database used text-to-speech to output information findings. The project could also be used as a translator to interact with locals in their local language. Again, VoIP technology was used to transfer the voice to a server for further processing and speech recognition system to translate the spoken text to a query towards the database. Also, GIS maps in conjunction to 3D representation were used to create an augmented environment. The CRUMPET project supported multilingual content and used a PDA browser application, having a pen/screen modality. The SmartKom project provided for full symmetric multimodality in which all input methods were also available as output. This meant that the SmartKon system captured speech, gesture and facial expressions using sensor technology in order to try and capture a natural experience for the user in the form of daily human-to-human communication. This was achieved by allowing both the user and the system to interact. The Mycitymate, the Mytilene guide project and the Nokia maps project made use of the current mobile telephony devices available using keypad and joystick as input. No reference is given for pen input support. The Mytilene guide project offers video or audio guide as output information of sights while Nokia maps offer speech output as navigation support. All the three above stated projects offer multilingual support. The REAL system used speech input to accept a request for a route description. This request was then transformed into a user specific request, taking into account limited cognitive and network resources. The request was then passed to the route finding module which determined and forwarded the optimal route to the presentation planning module. This module optimized the presentation of the route not only according to the resolution, screen size, and color capabilities of the output device, but also to the quality of the given sensor information (i. e. precision of location, orientation and speed of the user).

	keypad	Pen/Screen input	Multilingual support	Voice /speech input/ouput	3D Modelling output
Cyberguide	✓	✓			
GUIDE	✓	✓	✓		
LoL@	✓	✓	✓(3)		
Hippie/HIPS	✓	✓			
TellMaris	✓	✓		✓	✓
Deep Map	✓	✓	✓	✓	✓
CRUMPET		✓	✓		
SmartKom	✓	✓	✓	✓	
REAL				✓(input)	✓
MyCityMate	✓	✓	✓		
TIP					
Nokia Maps	✓	✓	✓	✓(output)	

Table 3-4. Summary of Input/Output Modalities

3.3.5. Unique Services

Many unique services where highlighted in the aforementioned reviewed projects. The services basically fell into three main categories: communication amongst users, tour generator and a log system. These projects felt that communication amongst tourist users was important. It was interesting to see that most projects tied the user's personal profile with the tour generator offering tours that reflect the users' personal interests. The log system was found in various forms, but as a fundamental system was used to allow users to keep track of visited sights and in some cases included a commenting system which was used to input comments about sights visited. Table 3-5 summarizes unique services stated in this section.

Cyberguide using rapid prototyping design methods implemented a number of applications. One such implementation had a messaging service whereby users could contact each other and also contact members of university staff. For the messaging service a wireless access system was designed to cater for communication between tourist users and the system. The system could document users' visits which, at a later stage, could be sent via email to the visitor in the form of diary log book. The GUIDE project developed an intelligent tour guide builder which calculated customized tour guides based on time constraints and dynamic changes to the user's environment (stopped, for coffee, slow walker). The users could override the guided tour to change the next location giving them freedom of choice. The Guide system also offered support for interactive services; a communication tool for visitors to contact the local Tourist Information Centre, messaging tool amongst visitors. It also had a built-in ticketing service where visitors could book accommodation and buy tickets avoiding queues.

In the LOL@ project, a tourist diary service was offered to users upon manual confirmation of arrival to a sight and or by accessing the *My Data* menu item to enter information. The visitor could enter comments in the form of text via a predefined screen including title, text and a link to a photo or a video taken from the tourist mobile device. This was later uploaded to the server and offered to the user for viewing as a log of visited sights. The log file was not integrated in existing Web technologies but was implemented on a propriety based Web platform in the form of Web pages. Similarly to the Cyberguide project the Hippie guide also allowed for interpersonal communication and general public communication of ideas through message sending. The mobile system also allowed for personal annotation added to the user's personal space to be accessed at the post-visit stage. Other prototypes of the HIPS project implemented dynamic generation of presentations depending on the distance the user has to an object of interest and how long they stood in front of that object. The TellMaris project quoted a number of services which could be implemented to be offered for both portable PC systems and for mobile systems. For portable PC services like weather forecasts, hotel reservation and navigational guidance services were thought of and for mobile devices users would be able to request information about various sights or restaurants, find closest facilities or to buy specific products. The CRUMPET system implemented guided tours and a group messaging service.

	Added agent-based services	Exploratory	Messaging/ Group Comm/tion services	Pre-visit Website	Post-visit web	Friend position finding	Ticket e-services	Guided tours
Cyberguide		✓	✓		✓			✓
GUIDE		✓	✓				✓	✓
LoL@		✓		✓	✓			
Hippie/HIPS		✓	✓	✓	✓			
TellMaris	✓	✓						
Deep Map	✓	✓						
CRUMPET		✓	✓					✓
SmartKom		✓	✓					
REAL		✓						
MyCityMate		✓				✓	✓	
TIP								
Nokia Maps		✓						✓

Table 3-5. Summary of unique services

3.4. Evaluation Results

In this section we discuss the results stated from the five (5) evaluation tables listed above. Clearly, with respect to the information model most systems used a decentralised web-based approach; others implemented proprietary-based applications using some sort of an adaptable information model which offered personalised information to its users, via a hierarchical menu system.

Most systems incorporated the use of a map as its central feature, which in turn offered navigational and routing services to its users. With only a few exceptions, GPS had been the standard choice for outdoor positioning technology. Certainly, this can also be confirmed in today's increasing tendency to incorporate GPS units in mobile phone devices, in comparison to telecommunication cell-id positioning technology which was used in the LoL@ system. Apart from navigational and route finding capabilities, only the GUIDE and the CRUMPET system offered basic itinerary planning for its users based on a personal profile. This feature was noted as being popular amongst tourist users visiting the city for the first time.

As of network capabilities, all systems used a HTTP IP connection which could be implemented in all cases of a network connection i.e. WLAN, GPRS, and UMTS. Nevertheless, only a few systems had the ability to adapt to network fluctuations offering on device cached content to its users; systems such as the CRUMPET and the REAL project were stated as being cognitive resource adaptable[7].

As of situational factors, only some systems enabled alternative input and output modalities. The systems implementing speech modalities for input or output were resource constrained acquiring constant large bandwidth to

[7] Resource adaptable is the ability to sense changes in network connectivity and being able to change the network connection without interrupting content flow.

function properly and efficiently. This was due to the fact that all speech processing was completed on the server side.

As of architecture, only systems using agent-based architecture (CRUMPET, SmartKom) were said to be standards-based and could be easily extended offering other services and integrating other sources of repository. Notably, only a few systems used thin clients of some sort to visualize a hypermodel; most systems employed the use of thick client applications to offer a richer customizable interface.

Finally, as of services offered to tourists, it was noted that these were split into three different stages of a tourism lifecycle: the pre-visit stage, the visit stage and the post-visit stage. The pre-visit stage was implemented by some systems offering the use of information for the places to visit and the ability of explicitly collecting personal information to build a personal profile. During the visit stage, services offered were mostly in the form of communication tool / friend finder tools and diary-commenting systems. The post-visit stage stressed the need of documenting the visit offering it to tourists as a means of logging their visit to be retrieved via the internet at a later stage.

3.5. Design Principles vs. Current Trends

This chapter focused on the evaluation of a significant number of research and commercial applications used by tourists to retrieve information, navigation and guidance using some form of mobile device in the scope of obtaining design principles to aid application designers and technology application developers. In order to evaluate these projects a number of design criteria were extracted targeting both designers and developers. These projects were grouped into four main groups: mobile guide applications, web-to-mobile guides, mobile phone navigational assistants (with built in tourist guides) and mobile web-based applications (the latter have not been extensively evaluated due to their resemblance with their web counterparts).

A number of milestone projects were incorporated which according to the considered bibliography offered unique experimental features and so was a number of commercial projects offering unique services.

After analysing the above evaluated findings (Section 3.3), we found that an application designer (when designing a mobile tourist application system) should have in mind at least the following three design issues:

1. which Information model to be used,
2. what Unique Services to be provided to its users,
3. which Input/Output modalities should be incorporated in the initial project design.

As of Information models, there are basically three main choices:

- Decentralised information model
- Centralised information model
- Distributed dynamic information model

The decentralised approach is an on-device solution where the information content is stored on the device. A number of solutions used this approach where some projects used memory cards to store and encrypt the content. This approach though, brought about content update issues and was mostly used in solutions that did not have the need for continuous content updates, i.e. static tourist guides. The distributed dynamic information model was used to push specific content to a user entering the vicinity of the network node. This approach works in environments wherein a network installation exists or in scenarios where the system can use the mobile network cell ID to push information to its users. The Centralised approach in most cases was used in conjunction to hypermedia content models. This was the case for the majority of projects evaluated and certainly there are a number of advantages of using the centralised hypermedia model (easy updating of content, access on a number of different mobile devices, etc); the problems that arise in this case are normally due to compatibility problems in microbrowsers (mobile browsers), which normally lead to project developers implementing a custom mobile browser based application i.e. the Guide project and also in some cases where there is a cost of connection which is incurred by the user, the roaming charges are very high making.

The use of a personal profiling system could increase the usability of the system in issue. Using a personal profiling system means the information model must be dynamically generated in regards to the explicit profile of the user and the implicit usage history. In conjunction with a personal profile and knowing the users whereabouts (using context awareness methods) as stated in the relevant projects (see Section 3.3) increase the usability of such context-aware systems.

Furthermore, an application designer should design a tourist mobile guide system keeping in mind Internationalization (multilingualism); the case of easily bringing more languages to the tourist guide should be integrated in the initial system design. Moreover, the designer should cater for users that do not have easy access to keyboard input and screen output implementing alternative input/output modalities, like speech input and output. This solution is network-dependant and might increase the cost of fast network connection.

Finally, the case of unique services and which services should be incorporated into a mobile tourist guide; comes down to the scope of each application designer's project. Below is a potential list of 'common denominator' services:

- guided tours,
- communication amongst users and the system,
- e-services (e.g. ticketing service, diary service, currency conversion, etc)
- group meeting scheduler,
- registering position to friends,

- pre-visit and post-visit services support,
- rating/commenting service.

Indeed, a technology developer should take into account the three above mentioned design choices but should also decide upon the system architecture, selection of application development platforms, network infrastructure, use of positioning technologies and mapping technology uses.

The architecture chosen by the evaluated projects' developers reflected the devices which they had to choose from. There was no real drift to particular devices or development platforms due to the large fragmentation problem which mobile developers face. Most of the systems implemented used the client-server architecture and some used propriety-based architecture, while a few used agent-based systems. The agent-based systems were mentioned as being standards compliant, which meant that the existing infrastructure without major changes was extendable in regards to new services using the same standards. As of development platforms, Visual basic .Net was popular for PDA-based projects while Java-based systems where popular for every other mobile device stated. All available network platforms which where incorporated to the system infrastructure included WLAN, GPRS, UMTS and GSM, yet notably WLAN was noted as being an expensive solution where as GPRS, GSM were mentioned as being a slow solution in terms of network bandwidth, as was the UMTS mentioned as being a costly solution which the systems users incurred.

Most systems used some sort of mapping technology, some of which were raster-based maps and others which were GIS-based vector maps. Surely in situations where routing and guidance was necessary a GIS map server is useful; yet, raster maps better suited systems where maps were solely used as a means of displaying the location of POIs, when comparing the network usages and technology requirements. As of positioning in outdoor situations GPS represents a reliable technology, while for indoor positioning IrDA and RFID tags are better suited.

As such, our evaluation revealed open research issues and some specific areas of mobile tourist guides research that need to be systematically investigated tying them to current trends of the Web. Specifically, more work could be carried out on social networking for tourist users especially in the eye of recent advent of mobile social networks [98]. The Mycitymate attempts to add such functionalities in the form of 'locate your friends' services and via commenting of POIs service but no attempt to connect content to current social networks (i.e. Facebook, twitter, etc) have been carried out. Even though much work has been already carried out on personal profiling systems no research has been noted on server-side clustering of users or collaborative filtering techniques of tourist content used in many popular sites (e.g. Amazon) [70]. This server side attempt would decrease device system resources (heavy algorithm calculations would be done on the server), while clustering of user profiles would enable proper assignment of users to group of tourists with similar interests, thereby providing space for the development of innovative personalized features.

Lastly, this evaluation identified a few projects stated the use of dynamic tour generation which the system apart from proposing specific POIs' to visit would also consider the user's profile in dynamically choosing the tour before generating it. Yet, there was no claim of dynamic itinerary generation, wherein users would state the time and days they have available for visiting tourist sights and the system depending on external parameters (i.e. opening times, weather conditions, peers choices and ratings) could generate a n-day itinerary for the specific user keeping in mind the 'must see' sites and the user's profile. Recent works proposing algorithmic solutions for optimizing personalized tourist itineraries [68] [157] have revealed the potential of such services but remain to be implemented and evaluated through field trials.

Early parts of this chapter were published in a WSKS'2008 conference paper with an extended version accepted for publication from the from Personal and Ubiquitous Computing journal (see Appendix A).

CHAPTER 4

AN INNOVATIVE MOBILE ELECTRONIC TOURIST GUIDE APPLICATION

4.1. Introduction

Tourists mostly use the web to seek out relevant information from the tourist industry regarding prospective places to visit. In addition, with the advent of Web 2.0 technologies such as social networks, blogs and wiki's, tourists can now also seek out information on the web from other tourists who have visited the destination and have published information for others to be informed. Similarly, the mobile phone goes beyond traditional voice communication devices to being an instrument facilitating interaction with web services [176]. Wireless access through mobile devices adds the element of 'portability' to the Internet connection, i.e. connection with no time or geographical constraints, by using devices with high dissemination in the public sector; tourists are amongst this technology-oriented public (e.g. the official online tourist office for Dublin [45] offers multiple platform support of HTML based website - PC, PDA, XHTML).

Specifically for the mobile tourism field, mobile guides enhance the tourism experience and offer a more appreciative experience; even more so by incorporating features like maps and location-based services [34]. There are still issues, however, which hinder the penetration and market success of such technology. Our in-depth study in Chapter 3 showed that tourists who have access to a mobile tourist guide containing personalized content, with no need for a constant connection, are more likely to use the mobile device in some way while at the destination. Furthermore, we argue that tourists could be motivated to use the mobile web to attain updated content and have access to extra services, if the benefits are sufficient enough. Building upon the extracted design principles discussed in Chapter 3, a framework for a mobile tourist application is presented in this chapter. This framework shows encouraging results towards personalizing tourist content using mobile technologies and represents a rather promising mobile business paradigm making use of the 'web-to-mobile' model. This model enables the web platform to create and deploy a custom mobile application that executes with no strict requirement for a constant connection. Little work has been carried out on this model, leaving enough space for research contribution.

Herein is presented the design and implementation aspects of a multiplatform tourist framework which, amongst other services, allows for tourists to use the web in order to 'build' unique mobile standalone guides that run on any mobile device offering tourist information and at the same time also offering extra personalized tourist services. The optional use of a network connection enhances the mobile tourist guide application by offering personalized content and social application services, For instance, use of personal profiles to classify users into groups (by applying clustering algorithmic solutions) with the intention of sharing multimedia content among users with similar interests.

This framework incorporates a complete multimedia tourist guide solution for a city with rural districts by utilizing multimedia platforms covering web, mobile phone and PDA applications along with an info-kiosk solution. This is the first attempt to study the synergy of these media platforms to date. Furthermore, this chapter conveys the experiences gained from implementing and evaluating a complete tourist framework for the Municipal council of Mytilene[8] for both web and mobile device users and reports the results of a thorough usability study.

The remainder of this chapter is organized as follows: Section 4.2 describes the main design decisions made with respect to this project; Section 4.3 presents the development issues related to the Multi-Platform Mobile Tourist Guide system for the Municipal Council of Mytilene, Greece; Section 4.4 presents a usability study of the system and evaluation results that evolved both from experimental and field studies and Section 4.5 draws conclusive remarks and presents directions for future research.

4.2. Designing the Multiplatform Tourist Guide Framework

One of the main design objectives of this project has been to provide tourist information via a web platform which would be readily available to users as a case of a pre-visit stage to the Municipal council. The main issue that was examined in the web application was how to make it obvious that users are able to adapt the web content into a personalized selection of content which would be dynamically 'loaded' into a customized ubiquitous mobile multimedia application.

Originally, the design of the platform was broken down into two separate systems:

- the web application and
- the mobile thick client application

A dynamic web application was designed and implemented, where, among other capabilities, users' could preview POIs illustrated on a scalable

[8] The multiplatform tourist framework project was jointly funded by the EU LEADER+ program, the Community Initiative Operational Program via ETAL S.A. and Municipality of Mytilene through the action 1.4.2.5 Signposting of sights, monuments, paths – mapping. In this project, Internet users may use the Mytilene e-guide to virtually navigate the Municipality of Mytilene, its monuments, museums and sights and get to know the history and the culture of the region as well as create a personalized e-guide for their mobile devices tailored to their individual interests.

Ajax-based map. The web application allows for users to dynamically select content of interest which the system then records and uses these selections to build a personalized mobile application. In the course of designing this dynamic web application, e-commerce concepts were used. The user tags content of choice, just as they might do with an "add to shopping cart" function, before moving onto the "download now" sub-section just like the "checkout" area of an e-commerce site. In this sub-section, the user is given the choice of downloading a customized mobile application to a PC first and then to the user's mobile phone.

Whilst in the process of the "Download now" sub-section the user has the ability, although not compulsory, to create an account in order to save the list of selected content with the intent of reusing it at a later stage. If the user creates an account, upon returning and logging in, they are given access to their own personal profile pages. These are separate web pages where the users have access to a number of services i.e. viewing their entered personal information, viewing or editing the POI selection "cart", using the comment and rating system and viewing and uploading photos via the web application or on the spot directly from their mobile device. The user can also use the recommendation system that is interconnected to their personal profile with the aim of adding new content to the selection 'cart'. Clearly, users must create an account in order to have a personal profile. This is compulsory if users want to benefit from having a customized personal guide and access to extra services available via the mobile web (such services will be presented in Chapter 5).

During the early analysis stages of this tourist framework, two extra scenarios were anticipated. Firstly, the scenario of a visitor who has arrived on the island without previously having visited the web application in order to download the personalised mobile guide. Secondly, the scenario where a tourist does not possess the minimum required mobile device capabilities to use on their trip. The first scenario was anticipated in the form of an info kiosk installation which gave users access to the web application and the ability to directly push the generated personal mobile application directly to user’s Bluetooth-capable phone. As for the second scenario, in cases where tourists did not possess the required mobile device or in cases where users were reluctant to use their own mobile device, a PDA installation was anticipated at the design stage. However, the Municipal lacked a WLAN installation, this led to the PDA installation having on-device storage, caching all POI data and it has a system for the centralized synchronization and recharging of all PDAs simultaneously.

Overall, the framework has been designed and implemented taking into account several usability guidelines from Section 2.3, as well as requirement results extracted from past experiments [37], [57], [66], which include:

- Both the web and mobile interface must be appealing to a wide range of users with various skills and expertise.
- The mobile application must include short and concise textual descriptions accompanied by pictures and multimedia content used to provide tourist information (e.g. museums, monuments, buildings, walking paths, etc.).
- Both web and mobile application presentations must follow a hierarchical multi-level structure that helps users to easily search for, browse and understand specific information tailored to their interest. This design approach has been dictated by usability studies which have shown that a combination of summary and keywords of each document at first is more efficient in helping users locate relevant information from a list of documents than displaying entire documents directly [50].
- Menus should be consistently designed to secure easy access to the desired information [57]. Menus and buttons should be clearly labelled and consistent to help the user navigation, learnability and memorability. To minimize cognitive overload, long lists of choices have

been avoided and support has been added for backtracking and easy accessing of earlier pages/home page. Finally, the menus' structure assist users to finish tasks with minimum interaction with the mobile device (e.g., scrolling and button clicks).

4.3. The Development of a Multi-Platform Tourist Guide Framework

Our main project objective has been to develop a web application which would collect user preferences to content and trigger a sub-system which dynamically builds a personalized mobile guide that could be used in both an online and offline environment. Our secondary objective was to incorporate a recommendation system, a mobile commenting system and have mobile-accessed location-based services. In the process of choosing a mobile environment which could satisfy the above objectives, current mobile web microbrowsers were reviewed. As a result, relevant research [89] revealed that current mobile microbrowsers are not fully compatible with the current PC based browsers. This means that mobile web applications are not yet able to provide rich user interfaces using available mobile scripting technology [77]. In addition, choosing to use microbrowser thin clients as the main mobile user interface meant that applications could not run in an offline situation or even less so, have the ability to detect a network connection in order to run 'online' whenever there is data to retrieve [141]. Additionally, in the case where the mobile application needs to access device resources like the GPS unit or the camera, this would not be directly possible via the microbroswer. It was decided that the functionality needed could not be achieved solely by using a mobile web application through a readily available microbrowser. Specifically, the web application which would be designed would need to be reengineered to have the ability to be accessed via a mobile device.

The following sections in this chapter describe implementation issues related to the development of the multi-platform framework which includes:

The web application with the embedded mobile application builder, the dynamically generated mobile application, the PDA installation and the supportive subsystem, the infokiosk installation.

4.3.1. The Tourist Web Application

The tourist web application is implemented using dynamic JSP (Java Server Pages) web pages which have access to tourist content accessed via a MySQL database.

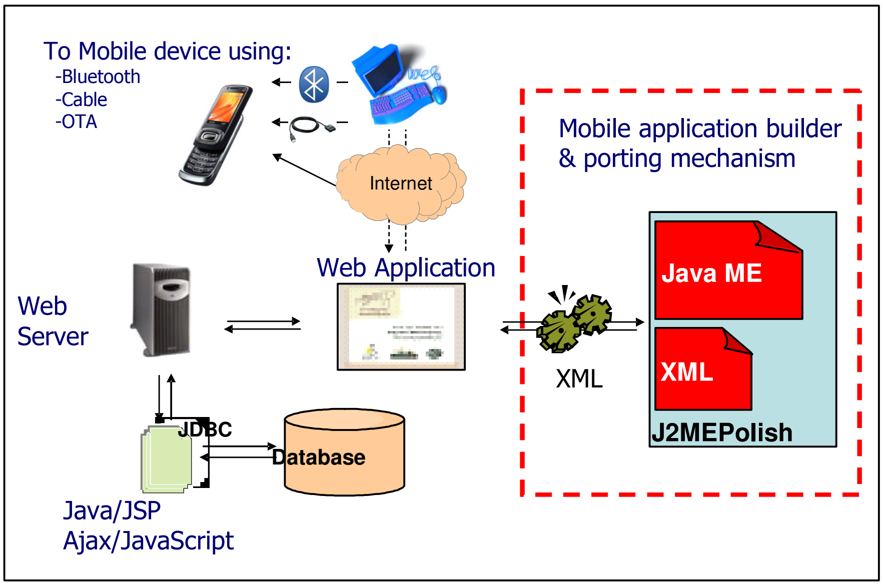


Figure 4-1. The architecture of the e-guide web subsystem

This application follows a 3-tier model, i.e. presentation - logic - database. An extra server is placed in parallel to the web server to serve as a dynamic mobile application builder and porting mechanism (see Figure 4-1) which is triggered from the web application. This server is responsible for accepting content which is passed from the web application in order to dynamically build the mobile application which is customized according to user parameters. The main objective behind the web front end is to allow the user to select content and generate a personalised mobile application which can later be downloaded either to PC and then passed to the user's mobile device (via Bluetooth or cable) or be downloaded OTA directly to the mobile phone.

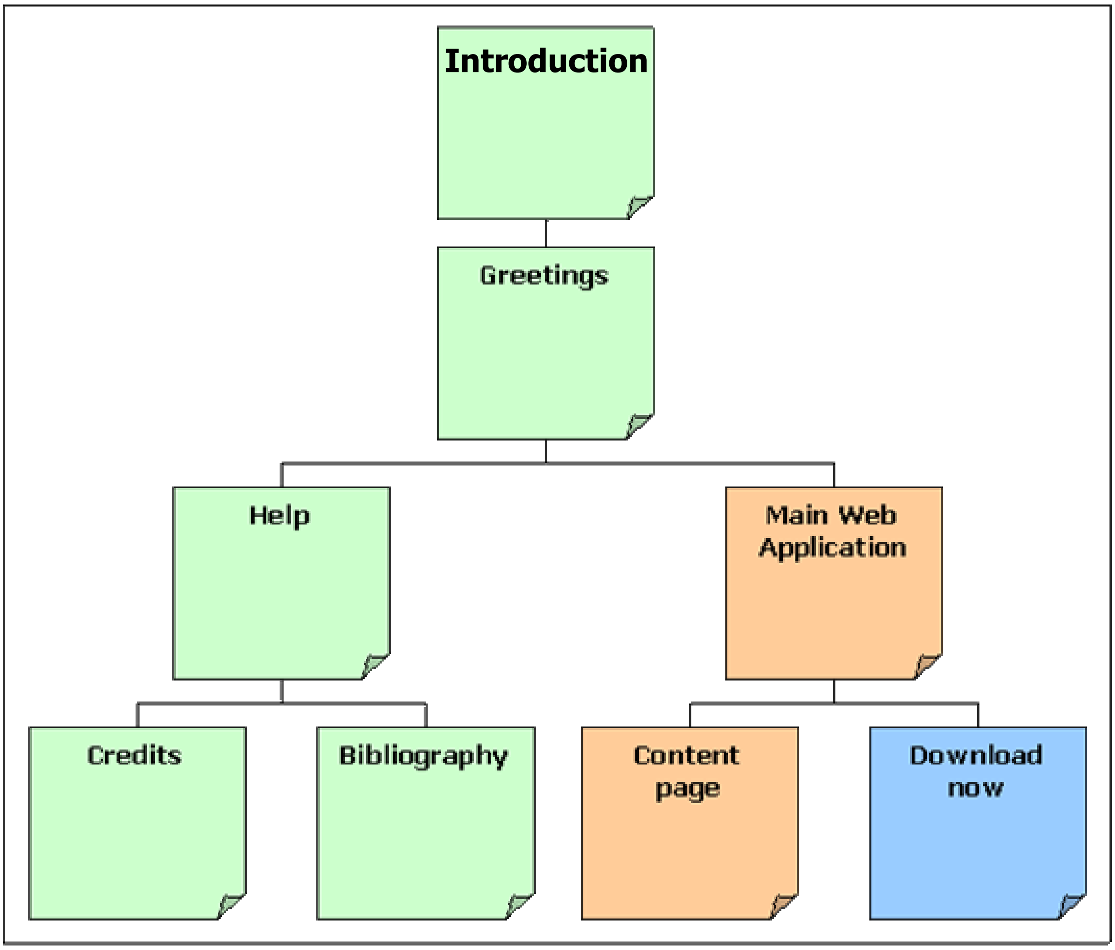


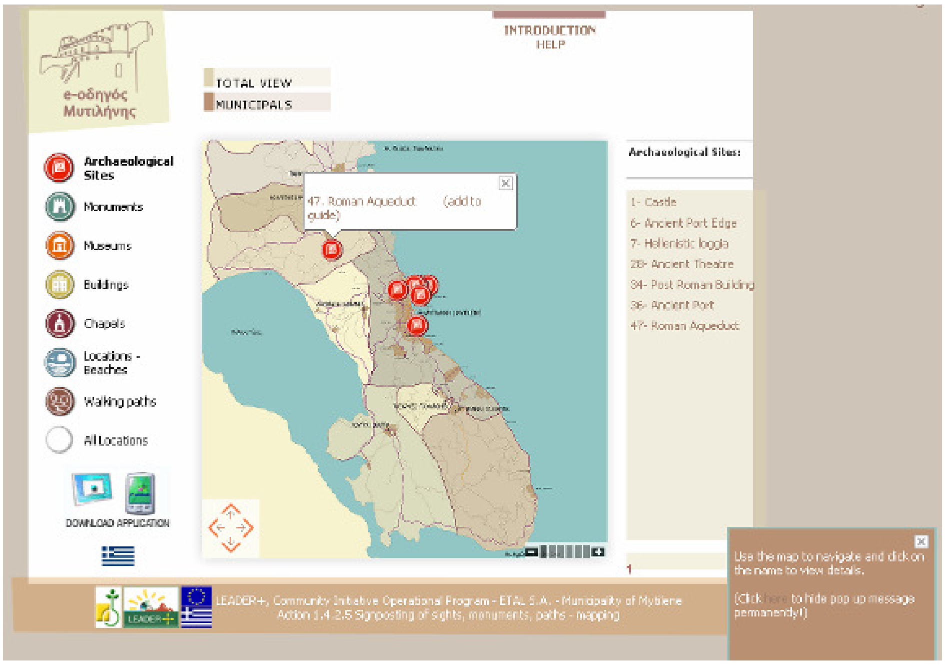
Figure 4-2. Web application content organization diagram

The web application[9] flow diagram can be seen above in Figure 4-2. The first page is the introduction web page where the user can select the language in which they wish to continue. The web application has been designed to support multilingual use throughout. All the web interface objects are localised for the chosen language and all content is saved to the database. In this implementation bilingual support has been tested for Greek and English. Following the choice of language settings, the next page is the "Greetings from the Mayor" Webpage in which the user can enter the main application page or go on to the help pages. The help pages have access to the bibliography pages and the credits page.

The main user interface screen is depicted in Figure 4-4 (below). The centre of the web page is occupied by a fully functional zoomable map which

[9] The Mytilene e-guide web application can be accessed in http://meguide.gr.

shows clickable markers of the designated tourist content. The map was designed from scratch using AJAX[10]-based technology [75] because at the time of implementation, Google maps^{tm} did not have any GIS data for roads of Mytilene to work with. The map was taken from GIS data and rasterised using vector graphics editing. In the map the POIs can be depicted using a color coding icon system found in the menu on the left hand side. When the user moves their mouse over the POI, the system shows a graphic pop up item, prompting the user to add the POI to their selection or to click on the text title of the POI at which point the user will be taken to the main content page (see Figure 4-3).



Use the map to navigate and click on the name to view details.

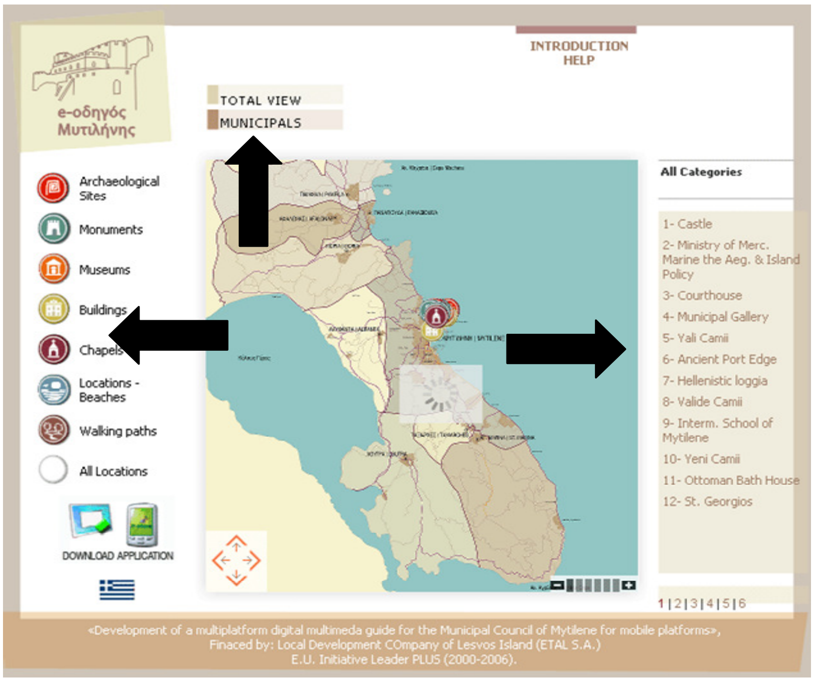
(Click here to hide pop up message permanently!)

LEADER+, Community Initiative Operational Program - ETAL S.A. - Municipality of Mytilene
Action 1.4.2.5 Signposting of sights, monuments, paths - mapping

Figure 4-3. Main web application screen showing the pop up POI item menu.

The menu located on the left shows the categories of all provided content, i.e. archaeological sites, monuments, museums, churches, etc., while the menu on the top shows all the regions of the municipality of Mytilene where various sites can be found. Navigation to the content web page can be achieved with the use of the left and top menus or the mouse or the keyboard to drag the map and click on the desired marker of choice.

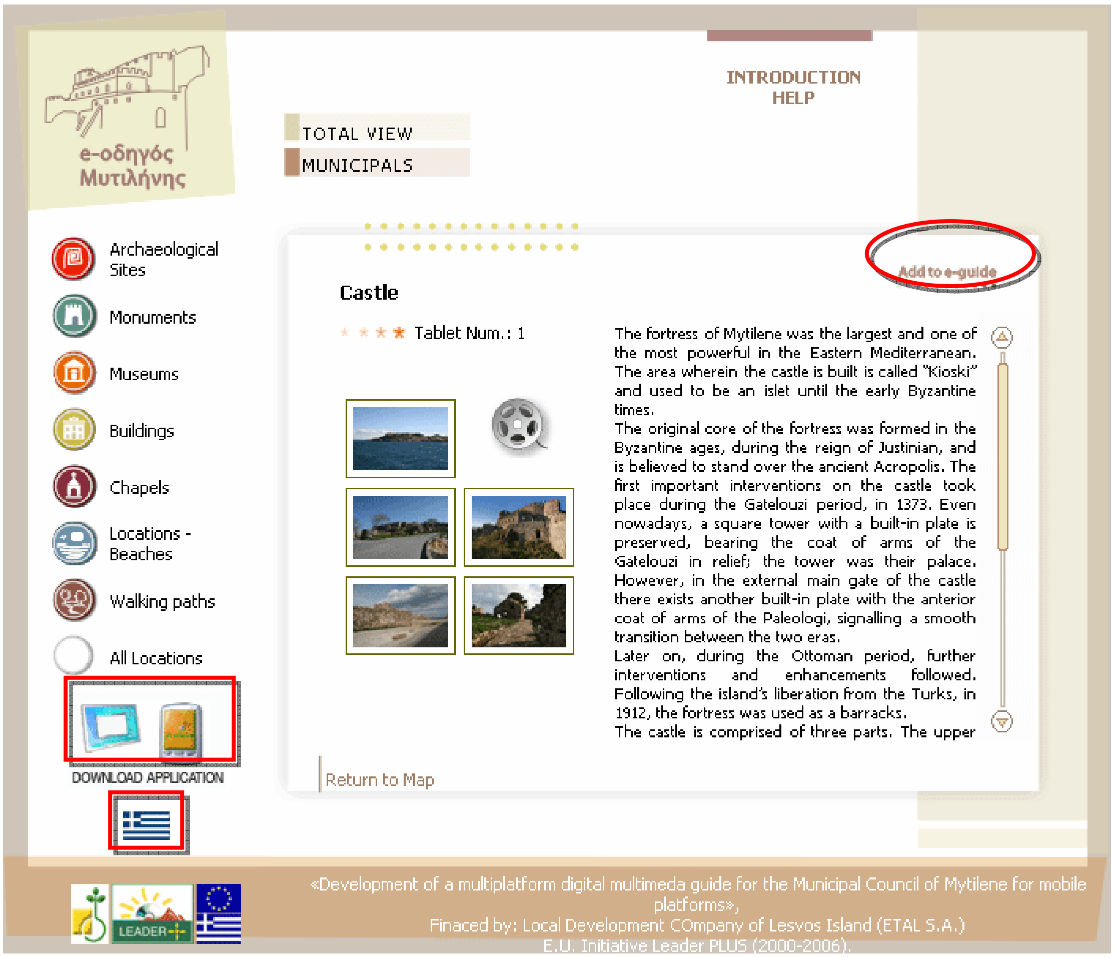
[10] Ajax (shorthand for asynchronous JavaScript and XML) is a group of interrelated web development techniques used on the client-side to create interactive web applications.



«Development of a multiplatform digital multimeda guide for the Municipal Council of Mytilene for mobile platforms»,
Finaced by: Local Development COmpany of Lesvos Island (ETAL S.A.)
E.U. Initiative Leader PLUS (2000-2006).

Figure 4-4. The main web application user interface

In Figure 4-5 we can see the main content page. This content page includes textual descriptions, thumbnail images, slide shows and a link to a multimedia video narration. On the top right corner of Figure 4-5 the "add to e-guide" button is emphasized. When pressing this button, the specific content item is added to the user's personalized content 'suitcase'. This suitcase is where the users' selections are held until the user is ready to move through the web application. On the bottom left corner of Figure 4-5 the "download application" button (check out) initiates the process of adapting the content to the user's device and generating the personalized mobile application to be downloaded and installed. In Figure 4-5 underneath the 'Download now' button there is a Greek flag. This represents the language button i.e. the ability to change over to Greek by pressing the button etc.



Castle

Tablet Num.: 1

The fortress of Mytilene was the largest and one of the most powerful in the Eastern Mediterranean. The area wherein the castle is built is called "Kioski" and used to be an islet until the early Byzantine times.

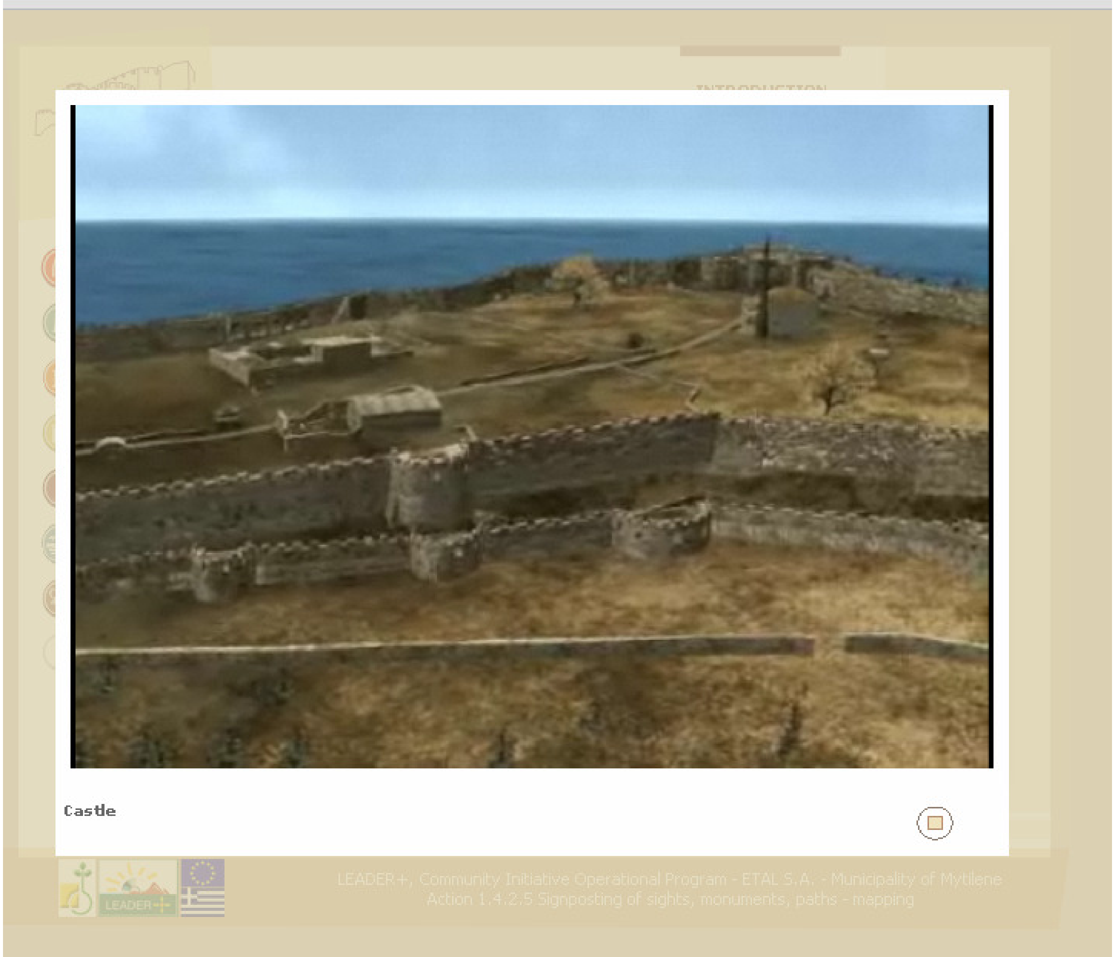
The original core of the fortress was formed in the Byzantine ages, during the reign of Justinian, and is believed to stand over the ancient Acropolis. The first important interventions on the castle took place during the Gatelouzi period, in 1373. Even nowadays, a square tower with a built-in plate is preserved, bearing the coat of arms of the Gatelouzi in relief; the tower was their palace. However, in the external main gate of the castle there exists another built-in plate with the anterior coat of arms of the Paleologi, signalling a smooth transition between the two eras.

Later on, during the Ottoman period, further interventions and enhancements followed. Following the island's liberation from the Turks, in 1912, the fortress was used as a barracks.

The castle is comprised of three parts. The upper

«Development of a multiplatform digital multimeda guide for the Municipal Council of Mytilene for mobile platforms»,
Finaced by: Local Development COmpany of Lesvos Island (ETAL S.A.)
E.U. Initiative Leader PLUS (2000-2006).

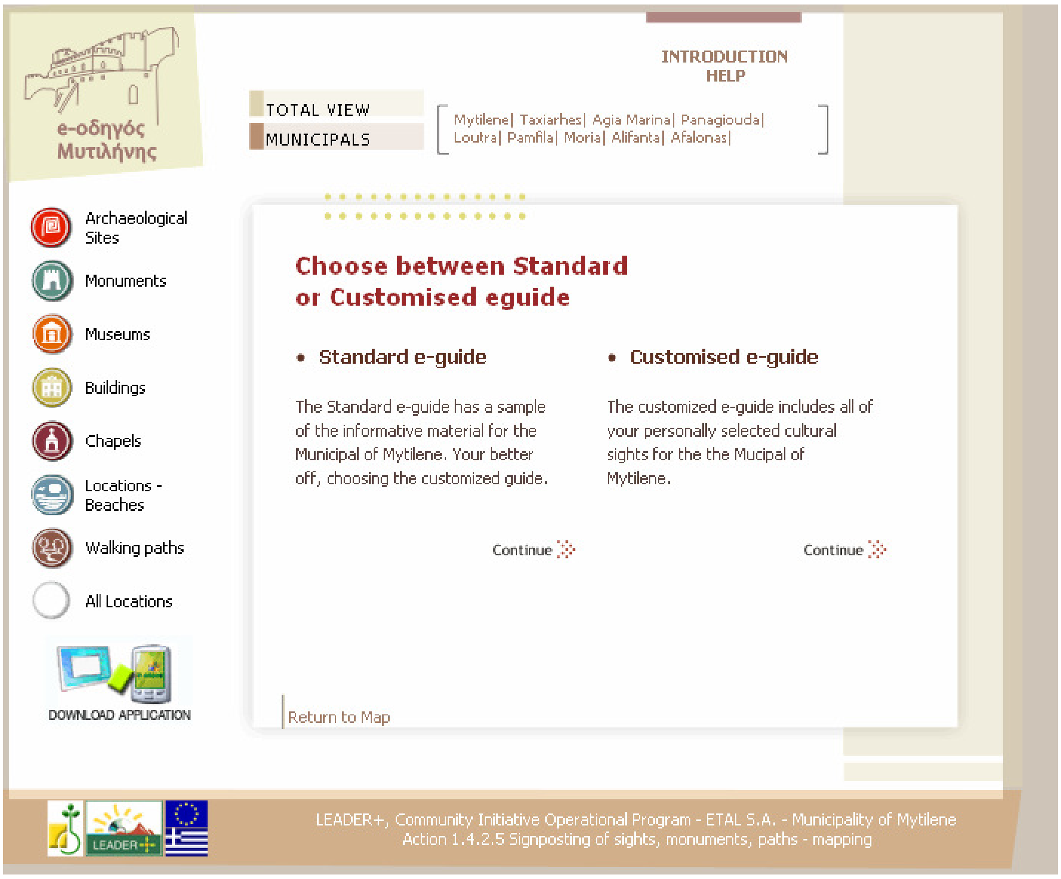
(a)


Castle
LEADER+
LEADER+, Community Initiative Operational Program - ETAL S.A. - Municipality of Mytilene
Action 1.4.2.5 Signposting of sights, monuments, paths - mapping

(b)

Figure 4-5. Screenshots taken from the Mytilene e-guide web application (a) typical content page (b) 3d video representation showing the ancient castle

In the 'Download now" process the user is prompted to choose between a standard or customised guide (See Figure 4-6). The standard guide contains a sample of ten content items chosen by significance from the municipality of Mytilene representing places of the most importance. The standard guide is readily available to users who do not wish to enter a profile or who do not wish to choose any sights by themselves.



Choose between Standard
or Customised eguide

- Standard e-guide

The Standard e-guide has a sample of the informative material for the Municipal of Mytilene. Your better off, choosing the customized guide.

- Customised e-guide

The customized e-guide includes all of your personally selected cultural sights for the the Mucipal of Mytilene.

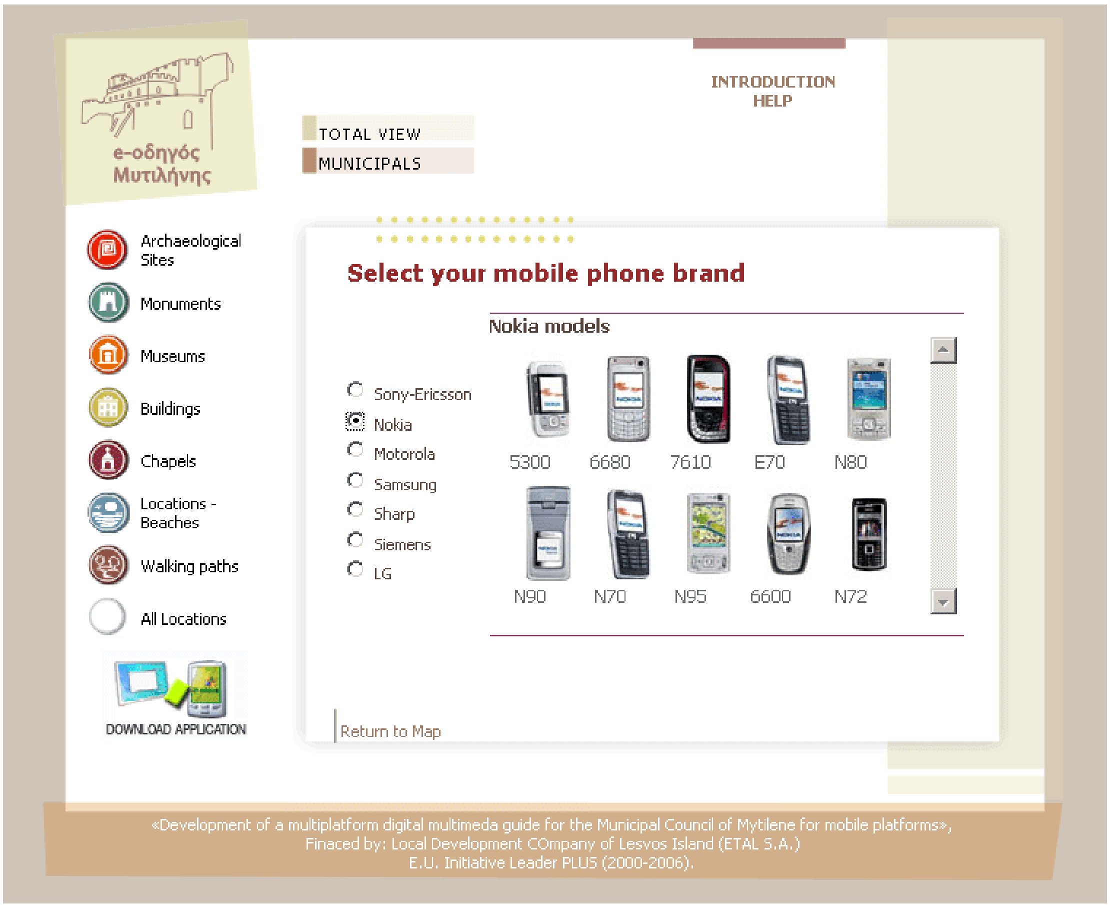
LEADER+, Community Initiative Operational Program - ETAL S.A. - Municipality of Mytilene
Action 1.4.2.5 Signposting of sights, monuments, paths - mapping

Figure 4-6. Screenshot where the user is prompted to build a Standard guide or a Customized guide

If the user wishes to generate a personalized dynamic mobile application a 4-step process is triggered, where the user is prompted to:

1. Login/register an explicit personal profile (indicating content preferences). If the user does not wish to register a personal profile the option of downloading a system generated mobile application is activated thereby bypassing this stage.
2. Specify the targeted mobile device for the application to be ported on (see Figure 4-7).
3. Choose memory consuming image files, video narration files, and scalable map files and confirm the desired language of the personalized mobile application. When this stage is completed, the personalized application is dynamically generated.

4. Download the mobile application to the user's PC and give instructions on how to install the application on her mobile device. The user has a number of choices to download the application to her mobile phone: (a) download the application to a PC and then by USB, IR or Bluetooth Connection upload to a mobile phone; (b) input a URL to automatically download the file; (c) receive an SMS notification which has the download URL already in place.



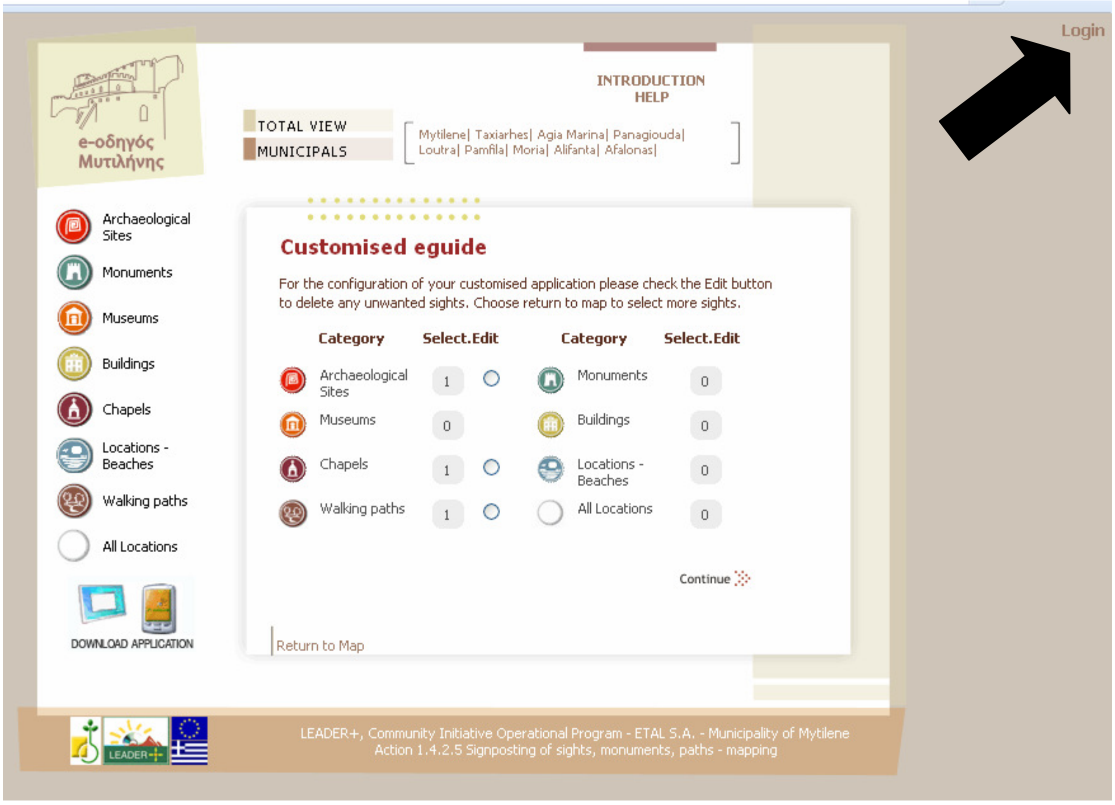
Select your mobile phone brand

○ Sony-Ericsson
● Nokia
○ Motorola
○ Samsung
○ Sharp
○ Siemens
○ LG

«Development of a multiplatform digital multimeda guide for the Municipal Council of Mytilene for mobile platforms»,
Finaced by: Local Development COmpany of Lesvos Island (ETAL S.A.)
E.U. Initiative Leader PLUS (2000-2006).

Figure 4-7. selection of the targeted mobile device.

The user is not obligated to create a personal account if she selects the Standard guide; however she is encouraged to select the Customised guide in order to create an account to save POI selection and to enable personalized services by the system. Figure 4-8 depicts the POI selection list where the user can see how many POIs of each category has been selected and edit i.e. delete POIs from the selection. In the top right hand corner when the user has not logged in he/she can do so at any time.



Login

e-οδηγός Μυτιλήνης

INTRODUCTION
HELP

TOTAL VIEW
MUNICIPALS

[Mytilene| Taxiarhes| Agia Marina| Panagiouda| Loutra| Pamfila| Moria| Alifanta| Afalonas|]

- Archaeological Sites
- Monuments
- Museums
- Buildings
- Chapels
- Locations - Beaches
- Walking paths
- All Locations

DOWNLOAD APPLICATION

Customised eguide

For the configuration of your customised application please check the Edit button to delete any unwanted sights. Choose return to map to select more sights.

Category	Select.Edit		Category	Select.Edit
Archaeological Sites	1	○	Monuments	0
Museums	0		Buildings	0
Chapels	1	○	Locations - Beaches	0
Walking paths	1	○	All Locations	0

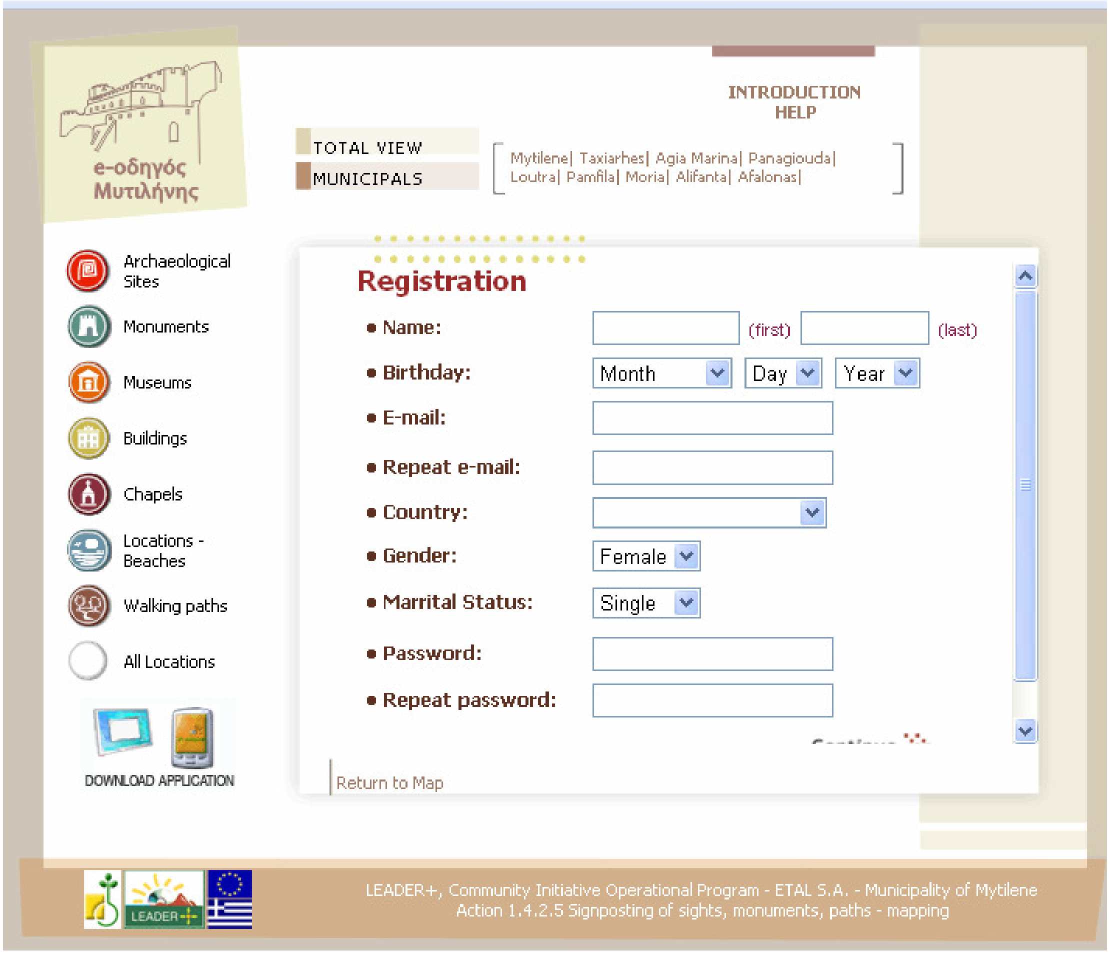
Continue

Return to Map

LEADER+, Community Initiative Operational Program - ETAL S.A. - Municipality of Mytilene
Action 1.4.2.5 Signposting of sights, monuments, paths - mapping

Figure 4-8. The user POI selection 'cart' and the Login button emphasized.

When the user 'checks out', the suitcase content is transformed into XML format whereby the web system automatically triggers the Mobile application builder and porting mechanism which generates the mobile application (to be executed on the user's mobile phone) incorporating the selected XML-based tourist content. The registration page is designed in such a way to allow the creation of an explicit personal profile used by the recommendation system to suggest content that matches user preferences. The registration form, other than the usual username/password/country, includes fields such as gender, marital status and age, etc (see Figure 4-9).



Registration

● Name:	(first) (last)
● Birthday:	Month Day Year
● E-mail:	
● Repeat e-mail:	
● Country:	
● Gender:	Female
● Marrital Status:	Single
● Password:	
● Repeat password:	

LEADER+, Community Initiative Operational Program - ETAL S.A. - Municipality of Mytilene
Action 1.4.2.5 Signposting of sights, monuments, paths - mapping

Figure 4-9. User registration form - Creation of user explicit profile.

While the user browses and selects content for her personalized guide, the system monitors and logs this activity (implicit profile data).

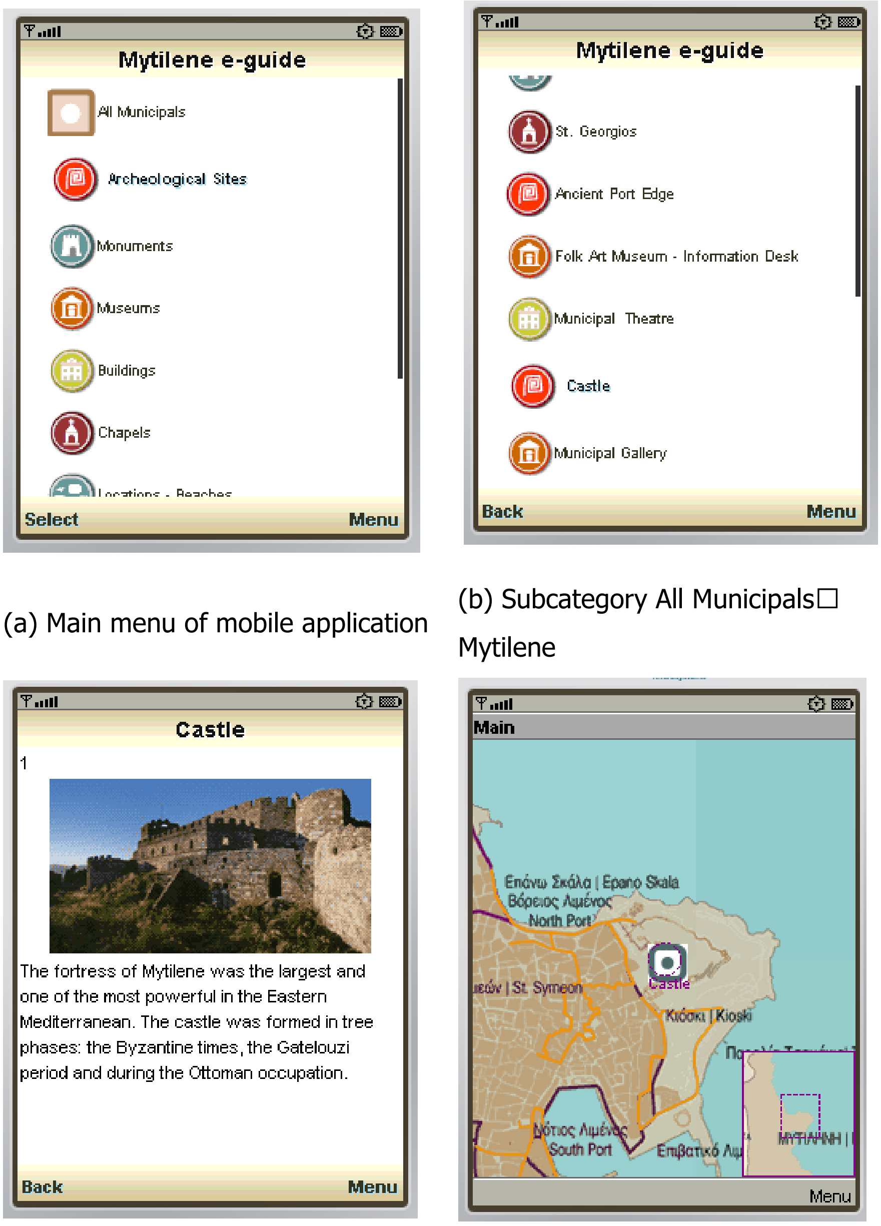
4.3.2. The Mobile Application

As already stated in 4.2, the best fit scenario for the development of the mobile application was the use of a thick mobile client. The actual mobile application has been developed on top of the Java ME Platform [163].

In order to port the mobile application to target mass scale mobile devices we have utilized the Java ME Polish porting framework[11] [86] in the

[11] Porting frameworks automatically adapt an application to different devices and platforms. Such frameworks include the following features: client library that simplifies development; build tools that convert code and resources to application bundles; device database that provides information about devices; cross compilers to port applications to different platforms.

form of a GPL license [87]. In particular, Java ME Polish enabled customization and porting of the mobile application to many devices, without having to modify the application's source code (the design of the mobile application along with animations and effects are specified in external CSS files).



The fortress of Mytilene was the largest and one of the most powerful in the Eastern Mediterranean. The castle was formed in tree phases: the Byzantine times, the Gatelouzi period and during the Ottoman occupation.

(a) Main menu of mobile application

(b) Subcategory All Municipals□ Mytilene

(c) Content page

(d) Map and marker of content page

Figure 4-10. Screenshots of the *Mytilene e-guide* Java ME-based mobile tourist application taken from a mobile phone emulator.

Although, (X)HTML-based web applications are nowadays a reality, mobile application developers demand more powerful abstraction mechanisms that hide all the complexity and enable the creation of rich user interfaces in (X)HTML. Market response has been signified by the appearance of AJAX Toolkits (Dojo, Yahoo, etc. and higher-level tag-based abstractions, such as JSF[12], XAML[13] or XUL[14] [140]. However, none of these technologies represent an open standard widely accepted by the industry, nor are they suitable for designing ubiquitous applications.

The mobile tourist application allows for dual navigation, from the text menu or directly from the map where the content is shown via markers (displayed either by category or by municipal region). Shown in the Figure 4-10 are screens of the Java ME application. In Figure 4-10a, the main menu is shown; this menu is dynamically built by the mobile application via the XML file which is generated by the mobile application server.

```
<?xml version="1.0" encoding="UTF-8" ?>
<city name="Mytilene" usr="131" cluster="1" total="89">
    <site cname="Ecclesiastical Byzantine Museum" municipalid="1" cat="3"
                                    code="24">
   <ctext label="Number">24</ctext>
   <cimage>24C.png</cimage>
       <ctext> The Ecclesiastical Byzantine Museum of Mytilene is located
               opposite the temple of St. Therapon, in the city centre. The icons
               collection of the museum is of significant value.</ctext>
    <cmedia>24_en.3gp</cmedia>
   <location type="point">
         <point x="8268" y="6796" />
   </location>
</site>
```

Figure 4-11. XML excerpt generated from the mobile application server.

[12] JavaServer Faces (JSF) is a Java-based Web application framework intended to simplify the development of user interfaces for J2EE applications. JSF uses a component-based approach; it also uses JavaServer Pages (JSP) for its display technology, but can also accommodate other technologies (such as XUL).

[13] Extensible Application Markup Language (XAML) is a declarative XML-based language created by Microsoft and extensively used by .NET Framework 3.0 technologies; typically, it is used as a user interface markup language to define UI elements, data binding, eventing, and other features.

[14] XUL (XML User interface Language) is an XML-based user interface markup language (developed by the Mozilla project) which operates in Mozilla cross-platform applications such as Firefox and Flock.

Figure 4-11 shows an excerpt taken from the XML file included in the mobile application. This XML file uses an English locale and includes user information, user identification and cluster group identification (the group which the user belongs to which is used for personalized recommendation services). This specific file includes POI information using a screen size of 240*320 pixels (the C in 24C.png indicates the largest screen size) and indicates that the user has selected 3gp video files (this user possesses an old mobile device which does not support the mp4 file format). A total of 89 POIs have been selected by this specific user.

If no 'Monuments' were chosen by the user on the web site, this menu item would not appear. In Figure 4-10b the content items of All Municipality of Mytilene subcategories are shown; this menu item is color coded in accordance with the municipal region (white for Municipality of Mytilene). Figure 4-10c illustrates a typical content page. The example content page contains only an image along with a short description; other pages may include more images and supplementary descriptive fields. Via the menu button the user can navigate to a video narration or audio narration (depending on what the user chooses when customizing the application via the web application).

The content is dynamically generated via the mobile application server after the user chooses his mobile phone model and is prompted to select the size factor which suites the user. The constraints related to supported multimedia file types and content are many. Not only are there constraints which arise from the selection of a mobile device with limited file storage but also constraints related to the storage space left free by the user. However, as device storage and external memory storage space offered to consumers continuously increases, this will become less of a problem in the future. Initial download time is not a major factor because, when the user creates the mobile application online, she is prompted to download it to a PC and upload it to the mobile device via a sync cable or Bluetooth connection. Users are also given the option to download the customized application directly from the web server (on average, such application files do not exceed 1.5 MB, as shown in Table 4-1).

For users with limited mobile device storage there is an option in the application customization pages (i.e. the 4-step process described in Section 4.3) for including map files and/or media files. As for the device limitations, after careful consideration of overall device screen characteristics, it was decided to use three main image file sizes adaptable to most devices available in the mobile market (see Table 4-2.). All media files, including image and video files, are optimized for these three screen sizes taken from initial media files of the web platform and converted to an appropriate size by the mobile application server during the build process. Following optimization, cumulative image file sizes are negligible for a mobile application (see Table 4-2). The original web application jpg files are converted to png format according to the device screen size. Media files sizes vary in regards to screen size and required quality. The system automatically selects audio files instead of larger video files for users having small storage capabilities or limited video playback options on their device. Average file size for mp3 audio files is 280Kb for both web and mobile device whereas video files downloaded to the device are converted from flv to mp4 format (or 3gp format for devices not supporting mp4). Average video file size is 320Kb. A limit of 10 audio or video files has been determined to limit initial application file sizes.

Parameter	Value
Application size (bytecodes, graphics for application, XML help files & Application language file)	620 KB
XML file size for 10 POIs	4.64 KB
Graphics file size with 10 POIs	120.8 KB
Total application size	746 KB
Total application size with audio files selected	+2800 KB
Total application size with video files selected	+3200 KB

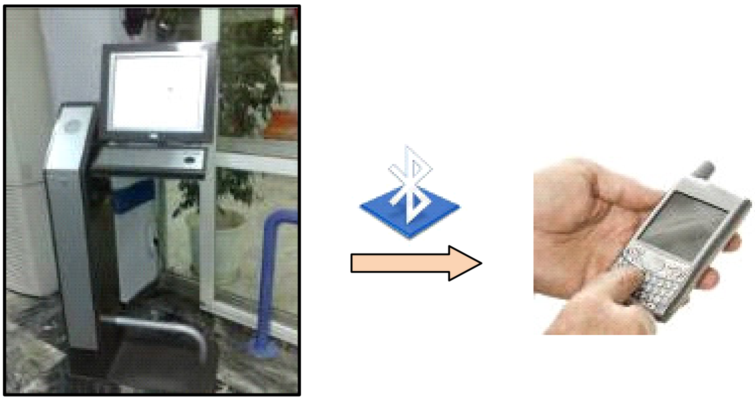
Table 4-1. Standard application file sizes.

As for available mobile device model characteristics, they are stored in a database synchronized with the device database of the Java ME Polish platform.

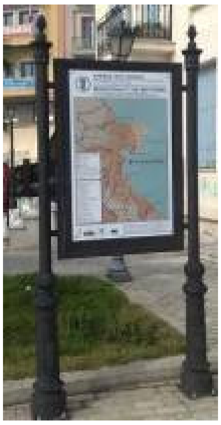
Device Full Canvas Size	Image size in pixels	Image size (kb)
128x160	100x55	6.08 KB
176x220	160x88	9.89 KB
240x320	200x110	12.8 KB

Table 4-2. Mobile device screen sizes which media files are adapted to.

The Java ME application has been designed in such a way as to dynamically build the content pages accordingly. Figure 4-10d depicts a screenshot of the map application and the position of a selected site. This application also handles multiple video and audio files used as narrations for tourists whilst on location.



(a)



(b)

Figure 4-12.(a) An infokiosk application is designed to push mobile applications upon being customized from tourists arriving at the airport. (b) Appropriate signposting in various areas highlighting the sights that tourists can visit in a particular region.

We have anticipated situations where visitors have already arrived on the island and have had no access to the web application beforehand. Thus, an info-kiosk subsystem was designed to be installed at the airport of Mytilene and at the local tourist office. The kiosk needed to have Bluetooth capabilities which would allow the user, upon completion of the building process, to push the mobile guide application to her mobile phone (see Figure 4-12a).

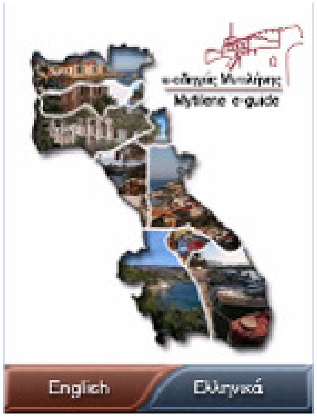
In addition, appropriate signposting has been placed in various areas of the Municipality of Mytilene which indicate the current position of the user and highlight the sights to be visited in the particular region as well as in the rest of the Municipality (See Figure 4-12b).

4.3.3. The PDA Application

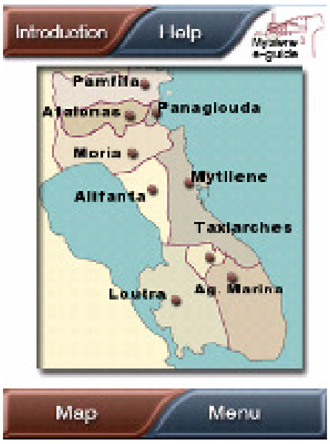
The web application is used as a tool to build the mobile application. As the early prototypes progressed, our first usability study (see Section 4.4) showed that users were reluctant to download and install the mobile application using their own mobile device or did not have a device with the minimum specification requirements (such as a color screen or Java-enabled mobile device). This confirmed the need for the municipal council to have a

PDA installation readily available to be offered to tourists. Nevertheless this changed the design implications of the architecture resulting in a separate PDA system design and implementation.

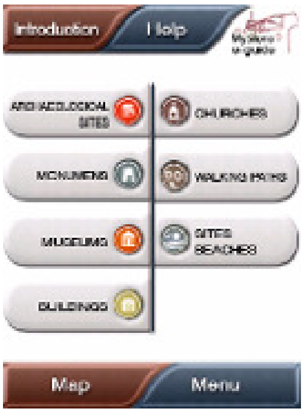
The PDA installation has been implemented using the NaviPocket v. 2.4 by OPHRYS SYSTEMS [132]. NaviPocket is an authoring tool which allows the creation of multimedia applications on PDAs or Tablet PCs The current version works with Microsoft Pocket PC 2002 and Windows Mobile 2003. The product is a complete set of an "Editor" (where the user creates a set of pages in text format, built according to an object-oriented model), a "Simulator" and a "Runtime". 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.



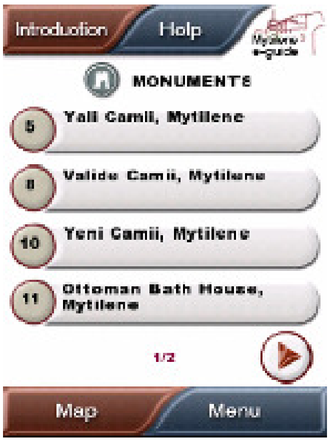
(a) Language selection screen of the PDA



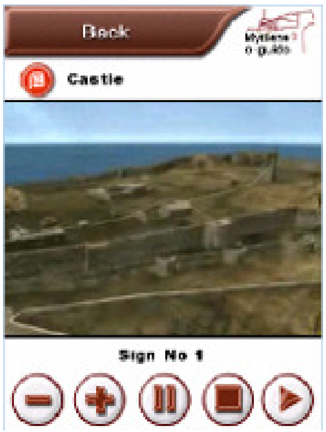
(b) Map screen of the PDA



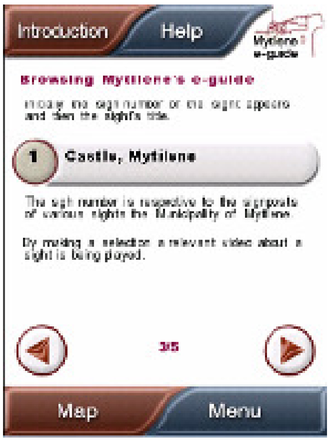
(c) Category selection screen



(d) Hierarchical menu screen



(e) Content screen showing video



Initially the sign number of the sight appears and then the sight's title.

The sign number is respective to the signposts of various sights the Municipality of Mytilene.

By making a selection a relevant video about a sight is being played.

(f) Help page

Figure 4-13. Screenshots of the PDA application which tourist can freely loan from the local tourist office.

The application has been designed for mobility by providing an interface with simple menus (see Figure 4-13) and concise information so that interaction with the application requires minimal effort and does not distract the user's attention from other activities (walking, talking etc.). Figure 4-14 shows the synchronization unit which can handle multiple application updates using one connection while also serving as a centralized recharging unit.

The user can navigate through the content by choosing the sections of interest manually with a PDA stylus. Alternatively, the content items may be automatically selected by the software, since it makes use of GPS technology (i.e. the users may be automatically notified when approaching to specific landmarks). The user can choose to switch between manual and auto guidance at any time while using the system. In addition, in order to avoid user distraction and address the stakeholders' request for incorporating multimedia techniques to effectively provide information about a sight, videos made of narration and animated pictures were included in the application.

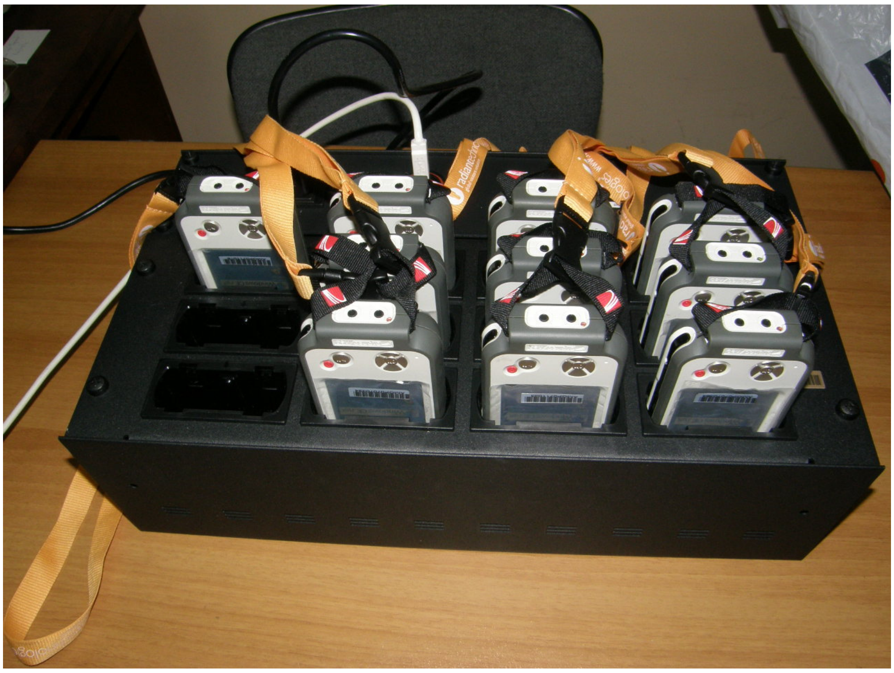


Figure 4-14. Suitcase synchronization and recharge unit.

4.4. The Evaluation

Three forms of evaluation have been performed to validate the effectiveness, usability and usefulness of our multi-platform guide. Expert/heuristic evaluation [128] and experimental studies were used to study basic usability and user interface standards. While field studies were used to extract various qualitative aspects of usability and user experience of the multi-platform solutions. The heuristic evaluation has been performed at early stage of the design (when application prototypes were developed) and within short span of time these helped to extract conceptual and formal weaknesses for the formulation of usability goals. Experimental studies are more suitable for standalone mobile applications, i.e. those without network connectivity requirement [186]. Field studies allows the transcription of use of mobile applications in a realistic environment and takes dynamic mobile context into consideration since laboratory experiments cannot record situations which are difficult to simulate [94].

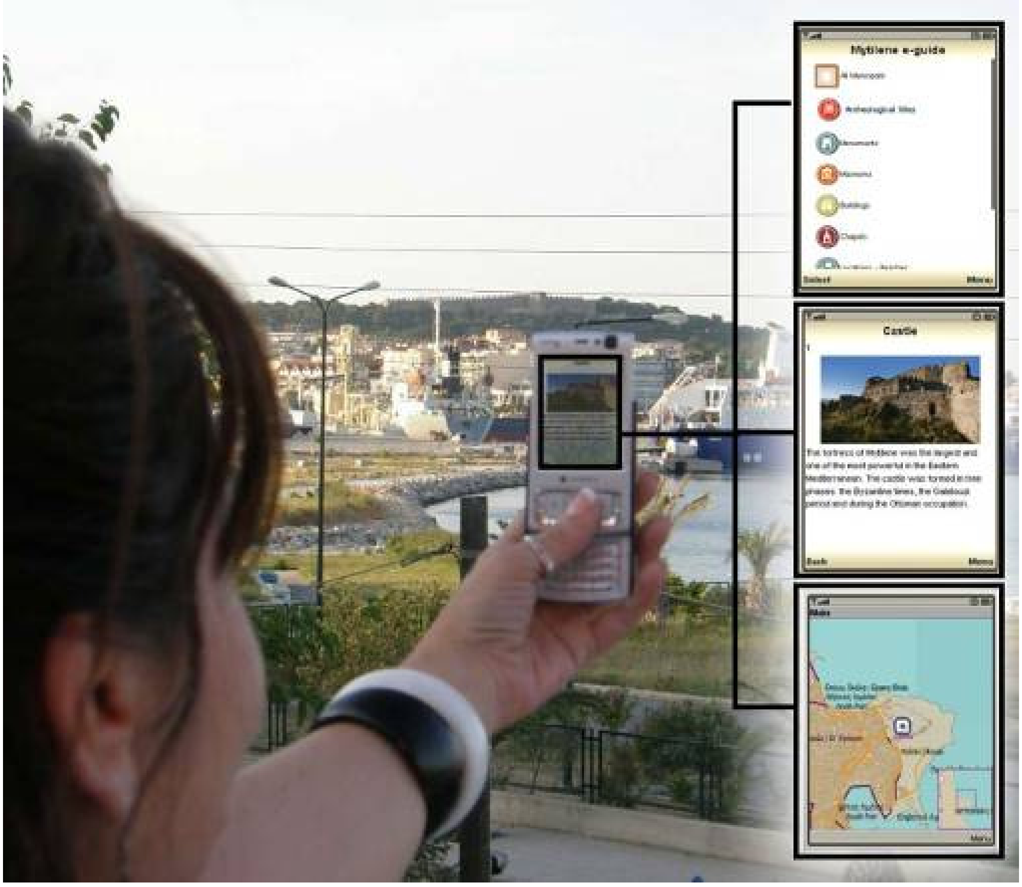
For the heuristic evaluation two trained specialists performed a usability inspection to judge whether each element of the user interface followed a list of established usability heuristics. To evaluate the tourist web site the ten heuristics by Nielsen were used [127]. To evaluate the mobile and the PDA application the following eight heuristics by Bertini et al. [22] were used:

- consistency and mapping,
- ease of input, screen readability and glancability,
- match between system and the real world,
- realistic error management,
- visibility of system status and device losability/findability,
- flexibility, efficiency of use and personalization,
- good ergonomics and minimalist design,

- aesthetic, privacy and social conventions.

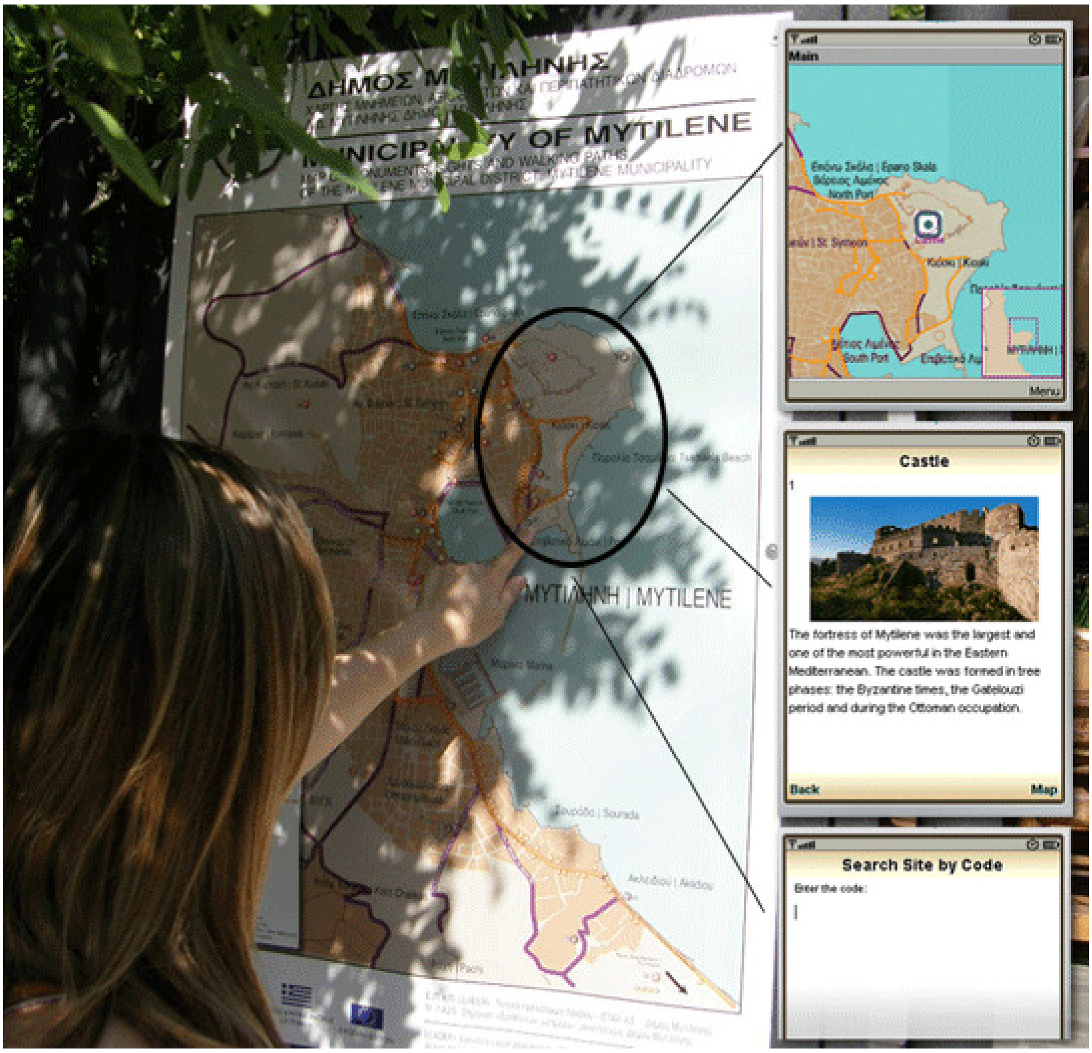
For the experimental and field studies twenty participants; twelve males and eight female, varied from 20 to 53 years, participated in the tests. All had visited the city of Mytilene before and were quite experienced with printed guides and maps in general, but only a few had seen or used a printed tourist guide or a map of Mytilene The participants were all experienced mobile phone users; nine of them had used standalone mobile phone applications in the past (e.g. calendar, address books, games, etc), yet, none had previous experience with electronic tourist guides usage.

For the experimental studies a phone emulator has been used, while for the field studies the participants' personal mobile phones have been utilized as test tools (after downloading and installing the mobile guide application (see Figure 4-15).



The fortress of Mytilene was the largest and one of the most powerful in the Eastern Mediterranean. The castle was formed in tree phases: the Byzantine times, the Gatelousi period and during the Ottoman occupation.

Figure 4-15. A participant navigating with mobile guide in the city of Mytilene.



The fortress of Mytilene was the largest and one of the most powerful in the Eastern Mediterranean. The castle was formed in tree phases: the Byzantine times, the Gatelouzi period and during the Ottoman occupation.

Figure 4-16. A participant navigating with mobile guide application by entering the number of POI that finds on the 2D map.

Each usability test session comprised four parts, an introduction to the scope of the session and the application, the main testing task, an interview and a questionnaire completion.

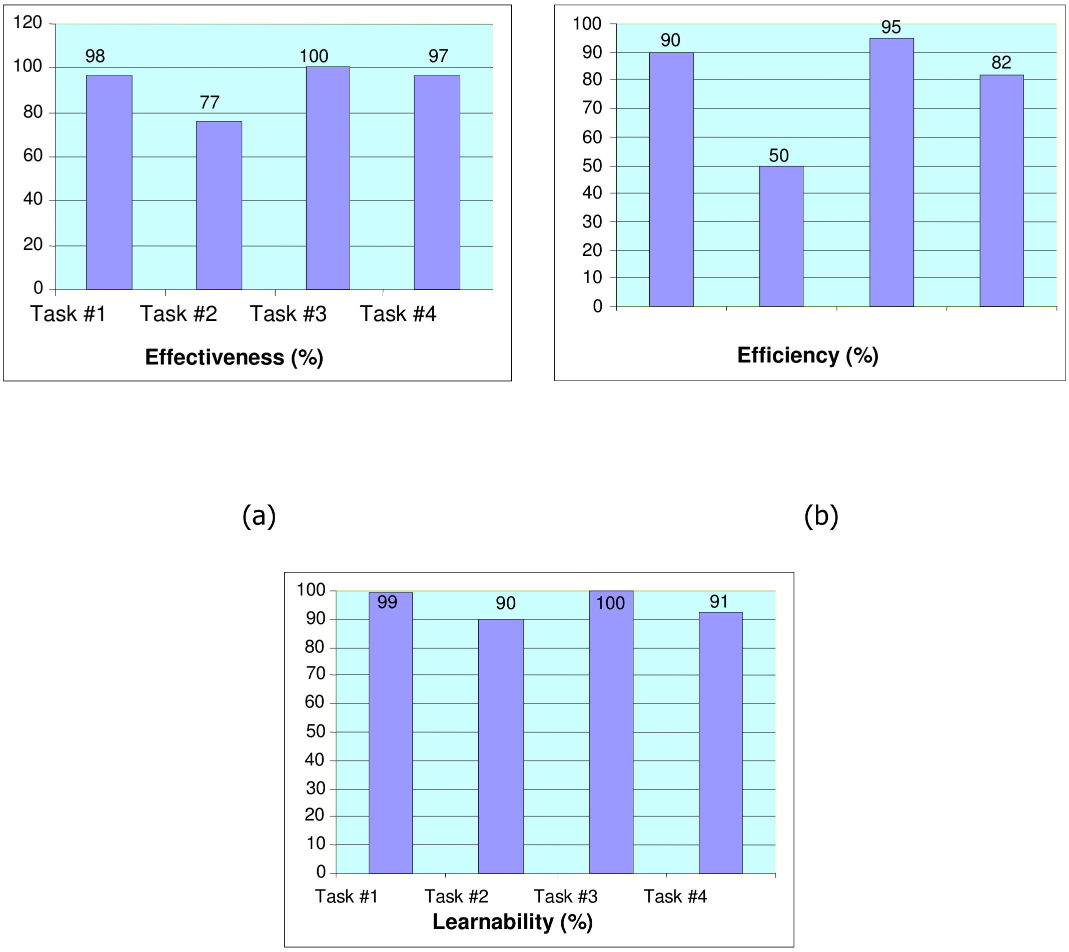
During the introduction, the overall project and the application were introduced to the participant, demographic information about them was collected and then the test procedure was briefly introduced. The testing stage included four different tasks which were assigned to each participant and were repeated twice. The four tasks are as follows:

1. Select content items from the web application or the info kiosk and then download the mobile application directly to the mobile device through a mobile network and the Internet;
2. Repeat the first task, but trying to first download the application to a PC and then to the mobile device (through Bluetooth);
3. Access specific content pages via the menu system of the mobile application;
4. Use the application’s 2D map to find the way from a place to another (see Figure 4-15) or use the map from the sign posts to enter the number of POI to visit (see Figure 4-16). It is noted that no GPS-enabled device was used to highlight the participant’s current position on the map.

During experimental studies, participants have been asked to accomplish specific tasks using the mobile application (i.e. it is easy to measure usability attributes and interpret results, while it makes it possible to use video or audio recording to capture participants reaction, including emotions) [29]. Out of the generic usability attributes identified in [63] and [186], usability tests recorded quantitative usability attributes measurements that fit in the nature of our electronic guide application such as:

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

Figure 4-17 summarizes the results of the quantitative usability attributes measurements recorded throughout the usability tests.



(a)

(b)

(c)

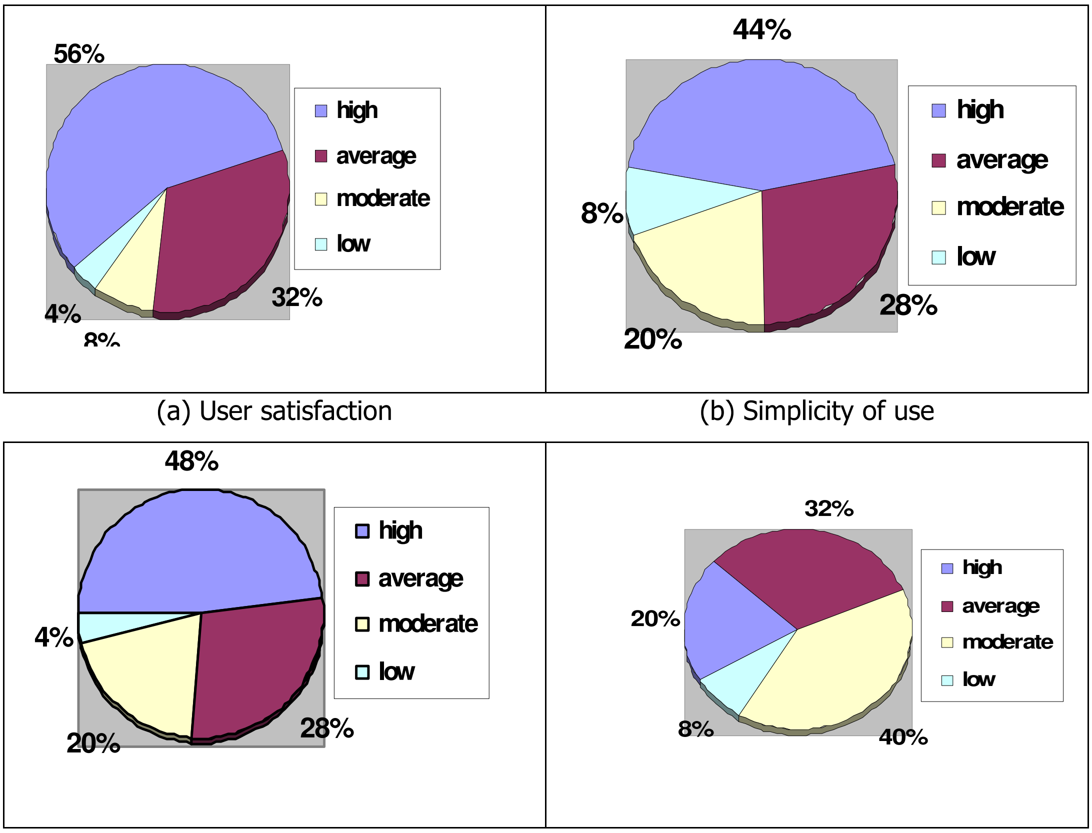
Figure 4-17. Measurement of quantitative usability attributes: (a) effectiveness; (b) efficiency; (c) learnability.

It was found that all participants were able to use the web application to view and select POIs with ease. Also, the 4-step process to build the mobile application was carried out by most of our experimental users without any problems. Table 4-3 shows quantative data which was recorded from the 20 users in regards to average number of POIs selected, average file size of mobile application, average download time via Internet (ADSL) to a PC, average installation times via Bluetooth and average installation times via OTA. The platform was found to be effective in regards to the OTA installation as also the task involving the use scenarios of the mobile guide. Only one participant could not complete the build process because their mobile phone was not Java-compatible, hence not in the list of mobile devices offered for selection via the web application. This user was guided to use the PDA installation. OTA download was found easier and straight-forward to deal with compared to downloading the application on their desktop first and then to their mobile phone through a Bluetooth connection. In fact, almost half of the participants had never used the Bluetooth connectivity service of their mobile phone. The platform was found to be effective in the use of the 4-step process. Highly effective was the use of the OTA installation (see Figure 4-17). However, most of the participants pointed out that the OTA application download time (~ 1.5 min) was too long and expressed concern of cost incurred due to use of the mobile network. As expected the learnability showed encouraging results due to the fact that users handled the system faster the second time around.

Percentage of users	**65%**	**15%**	**15%**	**5%**
Average number of POIs selected	24-89	10-23	Standard guide (10 POIs)	Used a PDA
Average file size of mobile application	1600-3200 KB	980 KB-2900 KB	980 KB -1200 KB	
Average download time via Internet (ADSL) to a PC	55 sec	45 sec	45 sec	
Average installation times via Bluetooth	127 sec	110 sec	110 sec	
Average installation times via OTA	250 sec	230 sec	230 sec	

Table 4-3. Usability results for mobile application selection, download and installation process.

Each test session ended with an interview (following a verbal protocol) and the completion of a short questionnaire, which revealed a number of qualitative usability attributes (see Figure 4-18) such as: user satisfaction, simplicity, comprehensibility and perceived usefulness. In particular, the participants rated these attributes in a 0-100 scale (0-24: low, 25-49: moderate, 50-74: average, 75-100: high).



(a) User satisfaction

(b) Simplicity of use

(d) Comprehensibility

(d) Perceived usefulness

Figure 4-18. Measurement of qualitative usability attributes: (a) user satisfaction; (b) simplicity of use; (c) comprehensibility; (d) perceived usefulness**.**

The experimental and field studies revealed a high degree of satisfaction among participants in terms of the amount and quality of the available tourist information. Most users found the web application simple to use in order locate and select content, to go to the "Download now" section, to generate the personalised mobile application and to install it to their mobile phone. The overall web and mobile platform was found to be highly comprehensible, yet many participants perceived the usage of the platform moderately useful, including the mobile application. On average it was found that users were reluctant in choosing many POIs and relied on the profiling system to decide. Viewing what the users peers chose gave them an idea of what would be interesting to choose.

Overall, participants appreciated the use of the personal recommendation system as it adapted the application's content to fit participants' personal preferences; also, the fact that the application could function on their personal mobile devices brought added a perception of familiarity to them. They also realized the business perspectives of including advertisements in the web and mobile applications. A number of issues/recommendations were raised from participants in the interview and evaluated to suggest future enhancements of the mobile application:

- the ability to view own and other tourists' comments/ratings (in addition to being able to 'publish' tourist comments and ratings for a given sight);
- enhancement of digitized maps so as to provide three zoom levels (instead of two), target highlighting, street names and clickable objects, like city attractions, to provide quick access to important tourist information;
- inclusion of emergency contacts in the city map (pharmacies, hospitals, port & tourist police authorities, etc) and search facilities to locate the shortest contacts, depending on the user's location;
- provision of daily/weakly weather forecast reports based on the user's location;
- suggestion of daily, personalized tourist itineraries, dynamically adapted to predicted weather conditions.

Further, this feedback taken from participants will be used to improve the future versions of the platform including application content, enhanced features and services of the mobile application. In fact, some of the evaluators recommendations have been taken into account in our following research steps (see Chapters 5 and 6.

4.5. Conclusions

The web has a great deal of potential in which to promote local cultural information to tourist user groups. Due to the mobile nature of tourist users a multiplatform system was found to best fit this type of user group.

In the context of the Municipal Council of Mytilene project, a three-part system was implemented. A web tool was implemented to allow tourists creating customized mobile guides for their visit. A thick client application model was chosen that does not require constant connection to the Internet. The users "tag" the content which interest them through a familiar PC web interface and select their device model to download the application to a PC. Also, a standalone PDA application was implemented either used to manually view tourist content or used in automatic mode to present multimedia information depending on the user's position. Emphasis was given to personalized services in wherein the user's explicit and implicit profiles are used as an input to the content recommendation system; as such the use of personal web profile pages allow users to upload comments and photos for other peers to view.

A thorough evaluation aided at extracting conceptual and formal weaknesses and the formulation of usability goals at early stage of the development cycle. While usability studies showed that even though tourists were keen to use the application via a static web device, they were reluctant to use the mobile system via the web interface. For this case a mobile application was implemented which used the web as an interface to build the application and allowed users to use their personalized mobile application without the need to connect to the web. However, our studies showed that users who wanted extra content or wanted to receive more content which the system would generate by using a personal profiling system, would use a web connection to do so. Even more so, our studies showed that using a web interface to connect users to each other by allowing the uploading of photos and the writing of comments, complimented the use of the web platform allowing for tourists to connect with peers and a maintain a visit log to share with their family and friends at the post-visit stage.

Parts of this chapter were published in the Personal and Ubiquitous Computing and Multimedia Tools and Applications journals. Also, early versions of this work have been presented in numerous conferences (see Appendix A).

CHAPTER 5

A PERVASIVE MOBILE TOURISM RECOMMENDATION SYSTEM

5.1. Introduction

Tourists systematically publish tourist information making use of Web technologies such as social networks, blogs and wikis, providing dynamic content about visited destinations or relevant information of their visit to be sought out by other tourists. Extending the notion of e-tourism [159] to meet the vision of tourist service provision to nomadic users with no spatial-temporal restrictions is expected to become a reality within the next few years. Mobile tourist guides increase tourist engagement and offer a more complete tourism experience with a number of benefits such as audiovisual content, interactive maps, location-based services, etc [35].

Personalization has been recognized by researchers as a critical factor of effectiveness, added value and commercial success in tourism [147], [153]. Personalization originally found success in e-commerce sites providing recommendations for products and offering information to consumers to aid decisions on product or service purchases. These systems, known as 'Recommendation Systems', are based on information filtering in order to recommend content to users (e.g. films, books, news, web pages etc). One of the most popular approaches, referred to as *collaborative filtering*, utilizes knowledge collected by monitoring the behaviour and personal choices of the system's users, collectively known as the user's personal profile [165]. This approach nowadays represents the most popular and effective technology used in web recommendation systems [147].

In mobile tourism, personalization has mainly been addressed in the context of guides providing content recommendations that match user preferences which are then typically consolidated in 'user profiles'. The most commonly used methods to build a user profile enable extraction of user data:

- Explicitly, e.g. rating content within a given scale, ordering content from the most to the least interesting item, statement of preference among various content items, statement of favourite content items list, etc.

- Implicitly, e.g. recording pages visited by the user (also taking into account the visit duration and visit recurrences), monitoring the content selection/buying behaviour of users, and analyzing the interests of their social network (e.g. the list of user contacts in a social network), etc.

In Chapter 4 we proposed a multiplatform tourism framework which incorporated a dynamically generated mobile tourist guide application for the Municipality of Mytilene, Greece. The e-guide framework only addresses personalization in the context of allowing users to explicitly select tourist content to be included in a customized mobile application which is generated on the fly, adapting the application so as to meet the screen size and hardware constraints of the user's mobile phone.

Chapter 5 discusses numerous extensions upon this theme, mainly addressing various aspects of mobile tourism personalization. Indeed, some of the ideas presented herein have been shaped throughout the evaluation of our original prototype presented in Chapter 4 and the recommendations of user evaluators (see Section 4.4).

Hence, the mobile guide application extension presented in this Chapter allows the user to collaboratively contribute to the uploading and sharing of tourist-related information with peers, such as ratings, comments and multimedia content relevant to specific POIs. Additionally, through employing collaborative filtering techniques [165], we enable the delivery of personalized tourist content recommendations based on the ratings and evaluations of peers with similar preferences.

We also introduce the concept of 'context-aware rating' which implies the importance of users who upload reviews, ratings and comments while onsite (via their mobile devices) in comparison to other users that perform similar actions through a standard web interface while not on site. In this context, our Mobile Tourism Recommendation System (MTRS) assigns different weights to content provided by tourists in respect to the technology infrastructure used by them. Hence, MTRS captures context-aware user evaluations and ratings and uses such data to provide recommendations to other users with similar interests. Furthermore, MTRS provides several innovative personalized recommendation services to mobile users, taking into account contextual information such as the user's location, the current time, weather conditions and user's mobility history (e.g. POIs already visited by the user).

Additionally, recognizing that tourists often have difficulty interacting with remote tourist portals while on vacation due to the lack of networking infrastructures or even avoid mobile data communications altogether due to high roaming charges, we have implemented a cost-effective Wireless Sensor Network (WSN) [6] prototype (incorporated within MTRS) which may be deployed around tourist sites in order to provide mobile users with effortless and inexpensive uploading of tourist information and ratings about POIs through their own mobile device. WSN installations also enable accurate user localization (hence, location-aware recommendations) without requiring any specialized hardware (e.g. GPS modules) on the end user devices.

The remaining of the chapter is organized as follows: Section 5.2 discusses the concept of service personalization with respect to electronic and mobile tourism and highlights challenges and opportunities for mobile recommendation systems. Section 5.3 describes the architectural, functional and implementation aspects of MTRS and Section 5.4 presents the infrastructure and update mechanism for tourist information portal (based on WSN installations). Section 5.5 summarizes this chapter.

5.2. Personalization and Recommendation Techniques in Electronic and Mobile Tourism

Existing recommendation systems in e-tourism typically emulate services offered by tourist agents where prospective tourists refer to seeking advice for tourist destinations under certain time and budget constraints [21],[147]. Typically these systems compare elements of a user's profile with certain characteristics that function as threshold elements in order to predict how the user would potentially 'rate' content items (e.g. product, content, service) which a user has not yet considered [147]. These characteristics can be associated with informative content (content-based approaches) or with the user's social environment. From a technical point of view recommendation systems use content-based approaches whereby a user states her needs, interests and constraints based upon selected parameters. This system then correlates user choices with catalogued destinations described using the same list of parameters.

As with e-tourism, personalized services represent a crucial factor for the further adoption, infiltration and success of mobile tourism [150]. The unique characteristics of mobile tourism bring forward new challenges and opportunities for the evolution of innovative personalized services which have no place in the field of e-tourism. For instance, the knowledge of the exact user location offers appropriate ground for the provision of location-based services. Furthermore, user mobility could allow exploiting the knowledge of user's mobility history and taking advantage of a user's social environment lying in geographical proximity.

Existing research and commercial mobile tourism prototypes incorporate personalized services that fall into one or more of the following project categories:

- personalization based on preferences explicitly stated by users [118], e.g. recommendation for a visit to the Archaeological Museum of Athens to a prospective visitor that explicitly expressed interest in museums and archaeological sites;
- personalization dependent on the users geographical location (location-based) [20],[152], e.g. use of audio notification when a user is in the vicinity of a sight, display of nearby museums, etc;
- context-based personalization [72], [154],[169], e.g. proposing sights to visit depending on location, means of transport or time of day; semantic parameters could also be considered such as the sight's category, its similarity to other sights, etc.

Some systems also allow for the dynamic creation and adaptation of personalized mobile applications not only based upon selection of personal content but on the characteristics of the user's personal device (screen size, ability to handle multimedia content, free memory resources, etc) [120].

Common ground for the above discussed projects are services based on personal interests, location, context of use and device capabilities; they all exclusively take into account the individual user of the mobile application. Although this approach addresses important aspects of personalization reducing the information burden for the user, it fails to take advantage of information, behaviours, ideas, evaluations, assessments and ratings of other tourists with similar interests, giving rise to the cooperative production of tourist recommendations. Namely, there exists a lack of tourist applications which use collaborative filtering techniques15 [70] or data mining [117] at content level. Such techniques have proved particularly useful in electronic tourism environments (e.g. [59], [109]). However, collaborative filtering approaches have not yet been investigated with respect to service personalization in mobile tourism [55].

We argue that the element of mobility raises challenges and important opportunities for the service of personalization. For instance, a recommender system may evaluate user input and uploaded content with respect to user context, e.g. the user's device type and location at the time she uploaded content. Similarly, content recommendations may depend not only on the user interaction history but also the user's location and context, e.g. current local weather conditions or places already visited by the user. Furthermore, mobile tourist applications may utilize innovative services which support direct communication and social interaction between tourists sharing similar interests and situated in nearby locations.

5.3. Design and Implementation of the Recommender System

Our proposed Mobile Tourism Recommendation System (MTRS) builds upon the ground work laid by the Mytilene eguide, a web-to-mobile tourist framework which allows tourists to use the web in order to dynamically 'build' customized mobile standalone guides that run on any mobile device offering tourist information. In the original prototype (see chapter 4), users manually chose content items (information about POIs) after browsing all available tourist content; the chosen content items were later included in the mobile

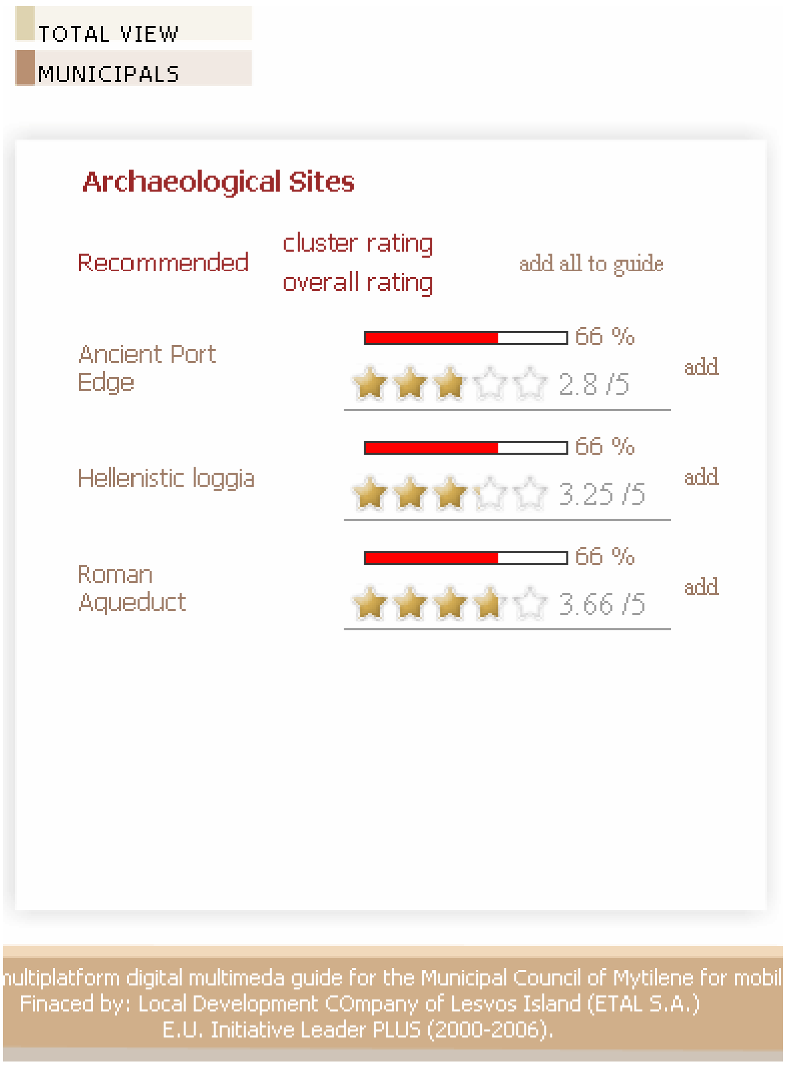
[15] Collaborative filtering, also known as social filtering focuses on the behaviour of users towards items/services, such as purchasing habits or preferences, rather than on the nature of items or services the system offers. In systems that use collaborative filtering approaches, recommendations are made by matching a user to other users that have similar interests and preferences. In this way, each user is suggested items/services that other users with similar interests have chosen in the past. Collaborative filtering techniques work best when there exists a broad user community and each user has already rated a significant number of items [55]

guide application and adapted to the user's mobile device (depending on the device's screen size and resource constraints).

The user evaluation (see Section 4.4) revealed that users experienced difficulty in locating items of potential interest to them within a pool of available POIs (it involves a time consuming process and considerable cognitive load). They also expressed interest in knowing the choices and opinions of other tourists with similar interests. The original system did not offer access to such information. Consequently, the user evaluation participants highlighted the need to implement a recommendation system which would automate the recommendation of POIs corresponding to the specialized user interests.

Herein, we discuss the design, architectural and technical aspects of MTRS. MTRS is a web-based system that records user interactions and feeds the user profile with information provided either explicitly or inferred by user actions. The user is not obligated to explicitly create a web account to register personal information (e.g. age, gender, tourist habits and interests, etc). However she is encouraged to do so in order to match the user's model to one of the available generalized user classes (stereotypes)[16] [148] thereby enabling suitable personalized recommendations from start-up. However, information about the user is also inferred implicitly by recording her usage and interaction history. Besides, implicit user modelling is considered more accurate, reliable and non-intrusive than explicit user modelling [92]. Figure 5-1 shows a screen where a particular user is recommended archaeological sites-related content based on the profiles of her peers and another displaying comments uploaded by peers.

[16] Stereotypes represent an important subject of recommendation systems research which refers to cases where a user that has not yet interacted with the system requires content recommendations. Even more so, in the case that only few users have interacted with the content system to implicitly provide usage data. As such, the system does not hold the required critical data mass to trigger the recommender mechanism and ensure reliable and suitable recommendations. User stereotypes address this problem associating user attributes (age, educational level, gender, profession, etc) with relative preferences; users characterized by certain attribute values are assumed to have specific preferences, hence, they are recommended relevant content.



TOTAL VIEW

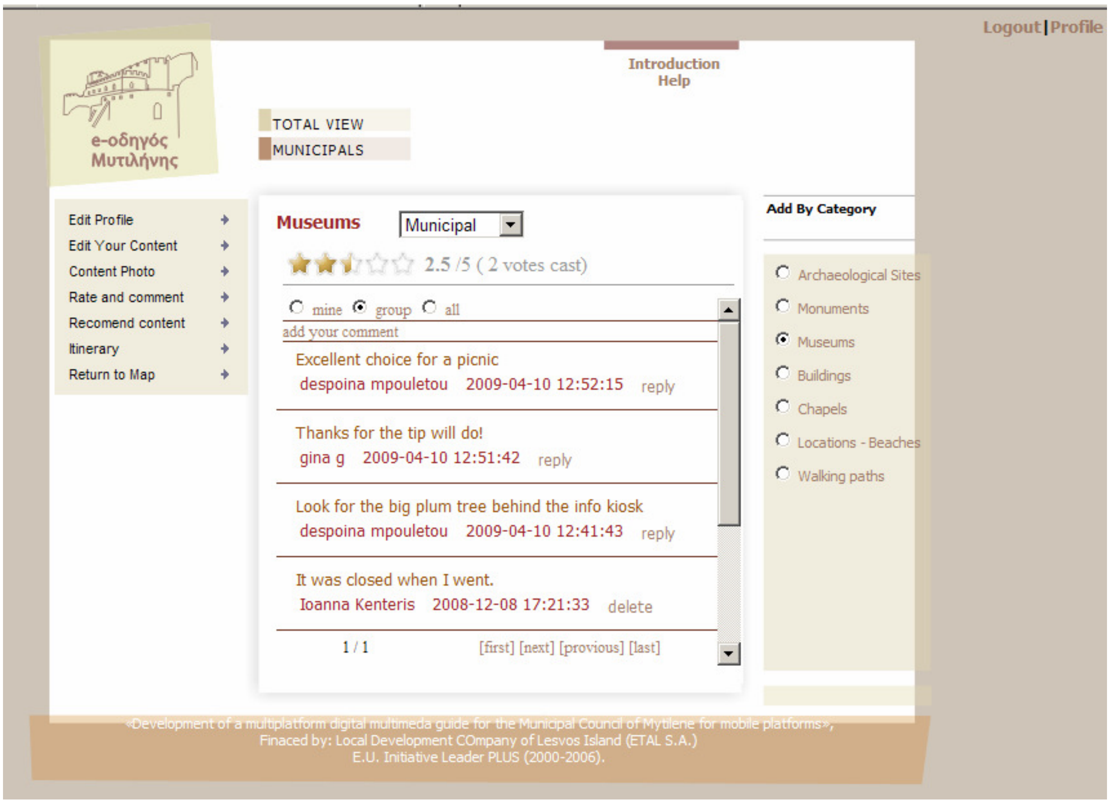
MUNICIPALS

Archaeological Sites

Recommended	cluster rating	overall rating	add all to guide
Ancient Port Edge	66 %	2.8 /5	add
Hellenistic loggia	66 %	3.25 /5	add
Roman Aqueduct	66 %	3.66 /5	add

ultiplatform digital multimeda guide for the Municipal Council of Mytilene for mobil
Finaced by: Local Development COmpany of Lesvos Island (ETAL S.A.)
E.U. Initiative Leader PLUS (2000-2006).

(a)



Museums Municipal

2.5 /5 (2 votes cast)

Excellent choice for a picnic
despoina mpouletou 2009-04-10 12:52:15

Thanks for the tip will do!
gina g 2009-04-10 12:51:42

Look for the big plum tree behind the info kiosk
despoina mpouletou 2009-04-10 12:41:43

It was closed when I went.
Ioanna Kenteris 2008-12-08 17:21:33

«Development of a multiplatform digital multimeda guide for the Municipal Council of Mytilene for mobile platforms»,
Finaced by: Local Development COmpany of Lesvos Island (ETAL S.A.)
E.U. Initiative Leader PLUS (2000-2006).

(b)

Figure 5-1. (a) Personalised recommendations to MTRS web users for archaeological sites based on information collected by peers; (b) comments uploaded by MTRS user's peers.

The recommendation engine was designed and developed to exploit user interactions which are collected explicitly and implicitly via users with similar interests and used to offer personalized recommendations of tourist content. MTRS uses a clustering algorithm [40], [79] to classify users sharing similar interests (see Figure 5-2). Similar clustering methods have been applied to a wide range of fields, e.g. clustering of users with similar mobility patterns in ad hoc networks [93] and also document collections [18] or web pages [166] with similar content.

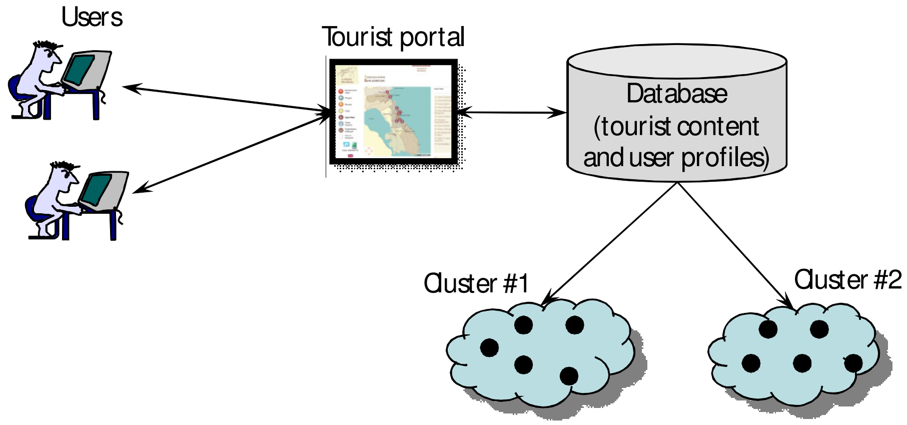


Figure 5-2. "Grouping" users by clustering the profiles of the registered web users.

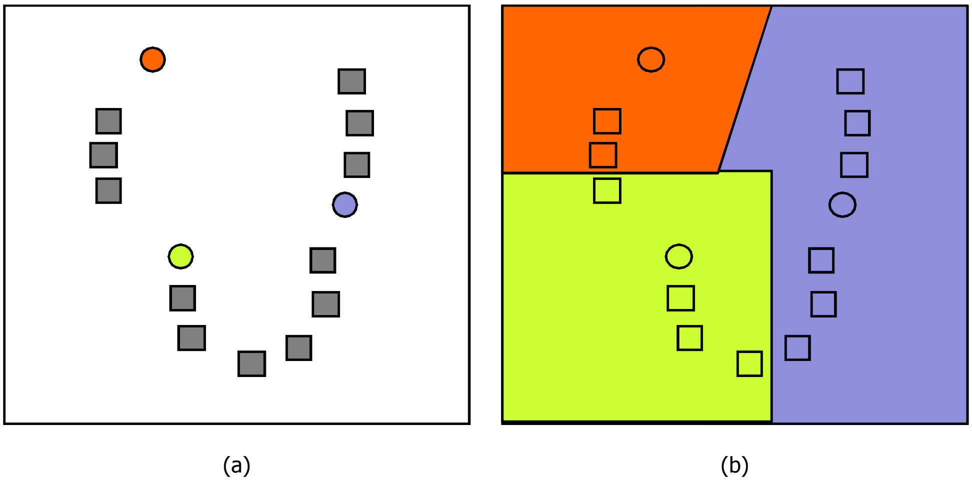
The registration form, other than the usual username/password/country, includes fields such as gender, marital status, age, educational level, etc. Optionally, the user may also state her preferences concerning favourite leisure activities (e.g. hiking, climbing, resting in parks) and POIs categories (e.g. museums, archaeological sites, monuments, zoos, etc). While the user browses and selects content for her personalized guide, the system monitors and logs this activity (implicit profile data). Using this personalized content list which is transformed into statistical data per category section (e.g. how many museum-related content items has the user selected, how many walking paths has she browsed, how much time was spent on archaeological site pages, how many photos were viewed, etc) and utilizing the users' explicit data, the system periodically executes a clustering algorithm based on the k-means [79] clustering algorithm.

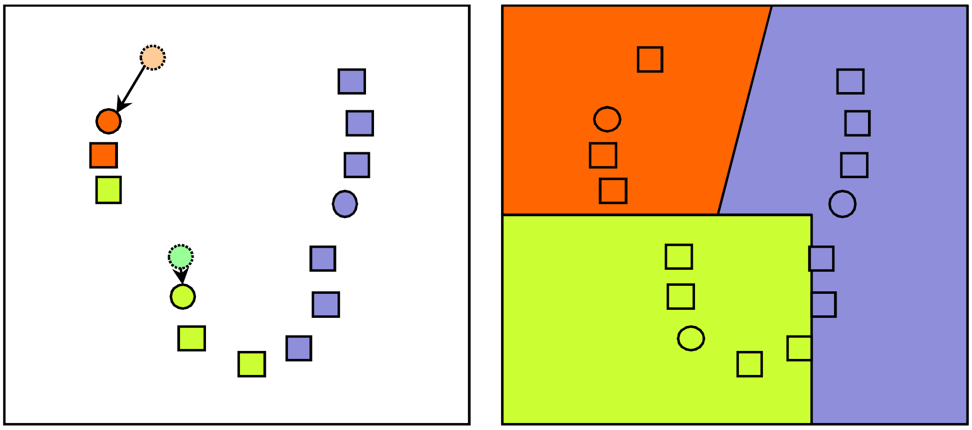
The k-means clustering is a method of cluster analysis which aims to partition N data points into K disjoint subsets S_j containing N_j data points so as to minimize the sum-of-squares criterion:

$$J = \sum_{j=1}^{K} \sum_{n \epsilon S_j} |x_n - \mu_j|^2$$

where x_n is a vector representing the nth data point and μ_j is the geometric centroid of the data points in S_j. Although, the algorithm does not achieve a global minimum of *J*, it is used fairly frequently due to its ease of implementation.

The algorithm clusters data points as follows: initially, *k* "means" are randomly selected from the data set; in the example of Figure 5-3a the selected means (*k*=3) are shown in color. On the next step, k clusters are created by associating every observation with the nearest mean (see Figure 5-3b). The centroid of each of the k clusters then becomes the new means, as shown in Figure 5-3c. The last two steps are repeated until convergence has been reached, i.e. no further changes are feasible (see Figure 5-3d). In our application scenario, data points include many attributes resulting in multi-dimensional vectors. A simplistic example is illustrated in Table 5-1, where attributes take values in a [0, 1] scale and represent the preference of each user to a specific POI category. The table entries are calculated through normalizing data derived either from explicit user statements (e.g. ratings) or implicitly (e.g. from the content items viewed/revisited/selected by the user, additional multimedia information requested/downloaded for specific POIs, time spent in viewing specific content pages, etc).





(c)

(d)

Figure 5-3. Shows a demonstration of the k-means algorithm classifying users into groups.

As the algorithm executes fairly fast even for a large number of users, we execute it multiple times with different starting conditions (i.e. selection of initial means) to ensure escape from local optima. It is noted that our implementation also considers additional attributes (such as the user's sex, profession and educational level) which typically take integer values.

user_id	architecture	monuments	museums	buildings	chapels	beaches	walks	cluster_id
1	0.6	0.3	0.5	0.6	0.3	0.2	0.7	**1**
2	0.7	0.4	0.4	0.2	0.7	0.3	0.2	**2**
3	0.8	0.2	0.2	0.3	0.3	0.6	0.2	**3**
4	0.5	0.3	0.4	0.7	0.3	0.2	0.8	**1**
5	0.7	0.6	0.4	0.2	0.7	0.3	0.2	**2**
6	0.9	0.3	0.2	0.3	0.4	0.6	0.1	**3**
7	0.8	0.5	0.5	0.5	0.2	0.4	0.7	**1**
8	0.5	0.5	0.5	0.6	0.1	0.0	0.7	**1**
9	1	0.4	0.3	0.0	0.6	0.2	0.5	**2**

Table 5-1. Sample clustering input data

(the final column refers to the cluster that the user is classified to).

The end result of the clustering algorithm is the classification of users into separate groups (users sharing similar interests); which is later used by the system to suggest content initially not chosen by the user but which is nevertheless, likely to be of interest to her (see Figure 5-4). In addition, the user can see the overall rating of the content by all users.

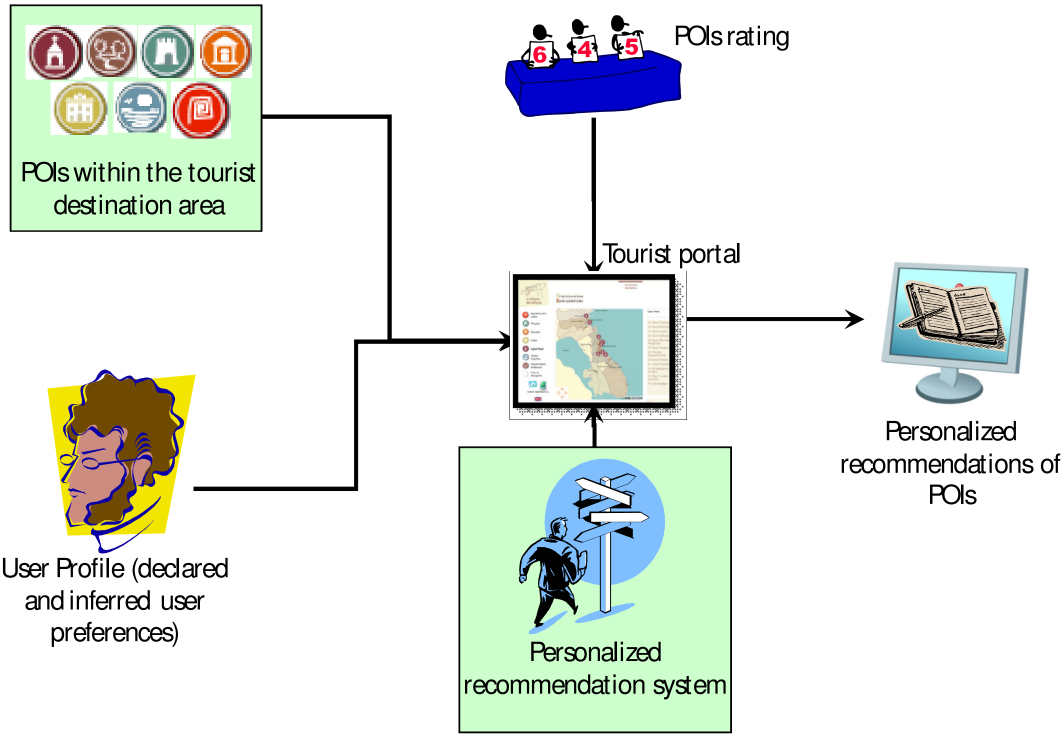

POIs within the tourist destination area
6
4
5
POIs rating
Tourist portal
User Profile (declared and inferred user preferences)
Personalized recommendation system
Personalized recommendations of POIs

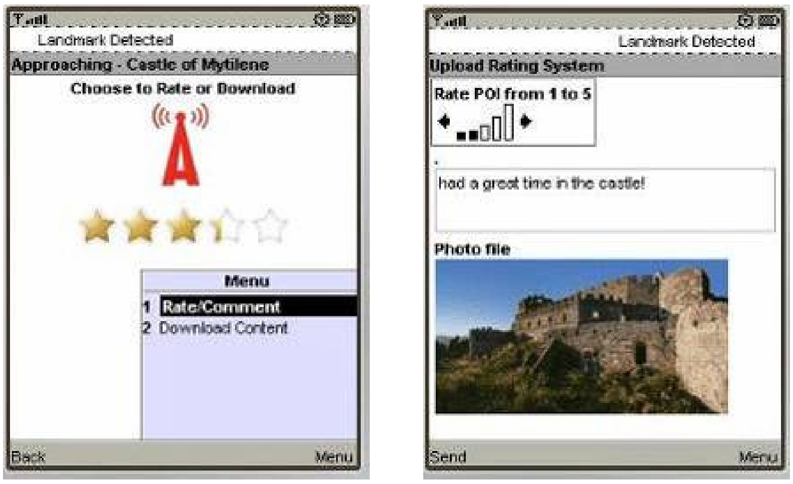
Figure 5-4. Personalized recommendations from the tourist portal to registered users.

In addition to web users, MTRS also supports personalized recommendations for mobile users. The server-side part of the system is essentially identical for both types of users; however, mobile users (typically smartphone users) are offered additional services as the MTRS takes advantage of contextual information transparently provided by the mobile applications.

The actual mobile application (i.e. the MTRS client) again has been developed using the Java ME. The content of the mobile application is personalized based on input given by the recommender system, while the application is adapted to the user's device screen size and capabilities. We assume that the Java ME-enabled devices support the Location API [90]; this is a valid assumption given that the Location API is part of both the full and the minimum JSR stacks defined by the Mobile Services Architecture (MSA)[17] [58], namely it is supported by the majority of Java-enabled phones.

[17] MSA is an emerging industry standard that aims to reduce fragmentation and provide a consistent Java ME platform for developers to target. In addition to specifying what component JSRs must be present on a compliant device, the MSA also clarifies behavioral requirements in order to improve the predictability and interoperability of the JSRs. The MSA defines two stacks: a full stack that comprises 16 JSRs (JSR 249), and a subset of eight JSRs (JSR 248). JSR 248 is being pushed ahead

Similarly to their web peers, mobile users may contribute to enriching web content by uploading their views, comments, ratings and photos of POIs they have already visited, using appropriate mobile application interfaces. The rating mechanism is either selected manually from the menu or is triggered when the application senses that the user moves within a short radius away from the POI (for a given time duration), whereby the user is prompted to rate the POI (see Figure 5-5a). The rating form has a five (5) star rating visual which is familiar to web users (see Figure 5-5b). The user is also prompted to write a comment in the form of a text message which will be posted to her profile and will receive a confirmation as soon as the upload is completed (see Figure 5-5c). The MTRS client transparently reports the current user coordinates (determined through Location API method invocations), which are stored in the MTRS database along with the uploaded multimedia content.

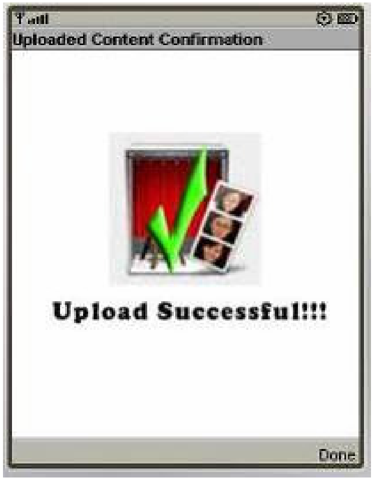


had a great time in the castle!

(a) Incoming proximity detection of a user

(b) User rating and commenting screen

of JSR 249 so developers can make the earliest possible start on MSA-compliant applications that will run on the highest-volume mobile devices. JSR 248 has recently been approved, yet, its adoption by OEMs remains to be proved.



(c) Successful upload confirmation

Figure 5-5. Screenshots of the mobile application.

The weighting of user ratings is increased if the user has already visited and rated POIs that belong to the same category (that implies that she can have a comparative viewpoint). Furthermore, the system allows the user to edit/update older ratings and comments so as to 'normalize' them based on the later experience of visits to POIs of the same category. Namely, we argue that a user may reconsider, for instance, an older strict rating for a museum of modern art having later visited another relevant museum hosting a poorer collection.

MTRS also allows for the rating of reviews and uploading multimedia content of fellow site members depending on the similarity of their interests as well as the clarity of their reviews, helpfulness and the credibility of their comments. Reasonably, the ratings of highly rated and credible users (those receiving high average ratings from other users) are treated with higher weight than others. MTRS presents ‘onsite’ POIs ratings and reviews (i.e. those uploaded from users via their mobile devices while onsite) with special markers, in order to differentiate them from reviews/ratings uploaded from web users.

Certainly, the MTRS mobile client primarily aims to deliver personalized recommendations to mobile users. While on the go, mobile users may request recommendations for the next POIs to visit. Apart from the user profile and the ratings/suggestions of users with similar interests, MTRS takes into account various context parameters for recommending appropriate POIs. Among others, these parameters include the user's location, current time, weather conditions and user mobility history context. Hence, MTRS prioritizes recommendations for POIs located in the vicinity of the user's current location. In particular, the recommendation list is built using a weighted function that assess the distance from the current location to the next candidate POI, the anticipated visit profit (depending on whether the POIs category matches user preferences) and the expected visiting time (this may depend on the actual POI 'size' and also on user preferences).

MTRS first excludes from the recommendation list POIs with opening times that do not match the current time, 'open-air' POIs if the weather is rainy and POIs already visited by the user. Local weather conditions are transparently retrieved from a web service (in our implementation we used the meteo.gr weather forecast service for Greece [111]). Similarly, POIs opening times may be retrieved from a cultural organization's web service (in our implementation, opening times were 'hardcoded' within the MTRS database). Having selected the next POI to visit, mobile users may use the MTRS 'How to Get There' service, which suggests and graphically illustrates the shortest path from the user's current location to the selected POI (using the Google Maps for mobile API [64]).

5.4. Infrastructure and Update Mechanism of Tourist Information Portals from Tourists

The increased demand for communication, the significant penetration of ICTs in the sector of tourism and the predominance of web 2.0 technologies have shaped a framework wherein users actively collaborate in authoring and updating online tourist content not only prior to but also during their visit onsite. Hence, tourists often need to update a visit diary and convey travelling experiences while at a tourist destination (e.g. upload photos, comments, evaluations, ratings etc.) and also to communicate with their friends and families or with peers sharing similar interests. This raises the need for tourists to interact with remote content servers directly via their mobile device in such a way as to be easy and inexpensive, without requiring the deployment of an expensive installation to support such a service.

In the design of such a system, our initial objectives included a rating system that could differentiate between a user rating a POI onsite or via a standard web interface. The problem that arose at the design stage was twofold; how to detect the proximity of users upon visiting a POI and also how to connect to the web server (tourist portal) not having access to a mobile connection. As for proximity, a number of technologies are readily available e.g. GPS, Assisted GPS (aGPS) and sensor technologies. In order to target a broad range of mobile devices, GPS could not be used since mobile devices that have GPS units represent a small range of devices today and the fact that GPS service may not be available indoors. On the other hand, a GPS seems to be increasingly available in all new devices that support data communications. Nonetheless, it was not chosen for proximity detection because it requires the use of a constant mobile connection to work and also due to localization accuracy concerns (assisted GPS techniques can have a range of accuracy from 10 meters to several hundred meters depending on the assistance technique used).

Apart from proximity detection, cases often occur where mobile device users do not have access to a mobile connection due to the rural positioning of POIs. In a city setting, a straightforward approach to the connectivity problem is to use a mobile device's HTTP connection (i.e. mobile browser) to navigate through mobile content. However, such solutions in mobile tourism do not always represent a viable business paradigm. This is mainly due to connectivity problems which arise at some remote sites and -most importantly- due to high roaming costs incurred to tourists. A reasonable alternative for this case is the use of a Wi-Fi installation. However, even if Wi-Fi installations exist, again, like GPS technology, not all mobile phones come with Wi-Fi capabilities.

We argue that a solid solution to this problem would be to install a low cost infrastructure to support proximity detection and a cost-effective means for remote content update. This infrastructure should fulfil the following two conditions: (a) support practically all available mobile devices, (b) allow relatively effortless and inexpensive deployment (to compress installation and maintenance cost), yet offer adequate coverage to its users.

The first requirement can be satisfied by the use of a Bluetooth connection which is commonly supported by the majority of mobile devices today. The second condition can be dealt with by the deployment of low-cost wireless devices at points with high concentration of tourists, placed either in open-air public places or in selected buildings' interior. A reasonable choice for such a solution is the use of small to medium-scale Wireless Sensor Networks (WSNs).

Sensor nodes are computational nodes of small dimensions and shipped with embedded low-cost sensors to record measurable parameters such as temperature, humidity, acceleration, etc. WSNs represent a modern wireless technology whereby nodes communicate with each other over a wireless connection and push collected data to a processing element (sink). A basic principle of WSNs is that they do not require a stable network infrastructure to operate. Namely, WSNs enable self-configuration and self-organization of an 'infrastructure – less' ad-hoc network topology.

The typical function of WSNs involves periodic multihop forwarding of recorded sensory data from distributed nodes to the centralized sink. Herein, we reverse this standard perspective and regard sensor nodes as inexpensive distributed network access points deployed around POI areas that provide the necessary infrastructure for tourists to upload multimedia content (text, photos, etc) to remote tourist portals. Within this approach, tourists located in the vicinity of a POI are revealed and prompted to connect to a remote content server (portal). The user then performs a Bluetooth handshake with the sensor node in order to establish a connection. Once in-sync with the node, the application receives the POI ID and the user is prompted to rate the

POI. The user may also upload photos taken from her mobile device to her personal profile web pages.

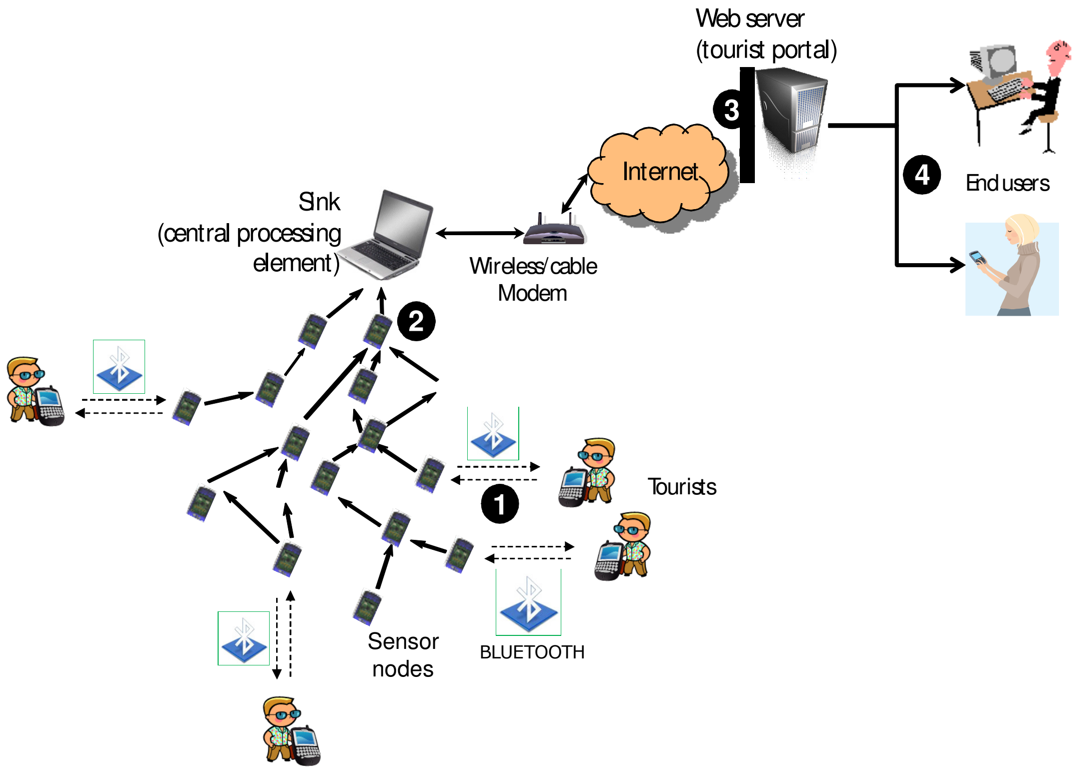


Figure 5-6. The use of sensor nodes as access points for tourists to rate and comment on visited POIs.

At a later stage, the sensor node forwards the data received from the user's mobile device to the sink (either directly or via multiple intermediate nodes) and the sink in turn transmits the data to the remote server (tourist portal). Direct communication of a sensor node with the remote server via end-to-end HTTP is also feasible. The sensor nodes may also transmit, along with the user rating and commenting data, environmental parameters values (e.g. temperature, humidity and light measurements) as well as their GPS locality. This allows the provision of up-to-date local environmental information to users interested in visiting nearby POIs. The architecture of our proposed approach is illustrated in Figure 5-6.

In our implementation, we used 20 SunSPOT sensor nodes of Sun Microsystems [161]. SunSPOTs are equipped with a processor (32 bit Risk clocked at 180 MHz), an IEEE 802.15.4-compatible radio transceiver, a rechargeable battery (3.7 V 720 maH) and three embedded sensors (accelerometer, temperature and light sensor). Each SunSPOT hosts a Squawk Java VM and may execute Java ME applications. Additional sensors and modules may be also attached to sensor boards. In our test bed, we attached 512 Kbps BlueSMiRF Bluetooth modems and GPS modules to SunSPOT devices. Furthermore, SunSPOTs enable ad-hoc connections with transfer rates of 250 Kbps which allows fairly fast uploading of low to middle resolution graphics. The transmission range of SunSPOTs in clear terrain is around 100m; hence, depending on the terrain's morphology, a 'grid'-like coverage with SunSPOT nodes placed within a distance of 30-50 m (that is less than 10 nodes per hectare) should be sufficient to guarantee connectivity even in the event of failure in a relatively large number of nodes [104].

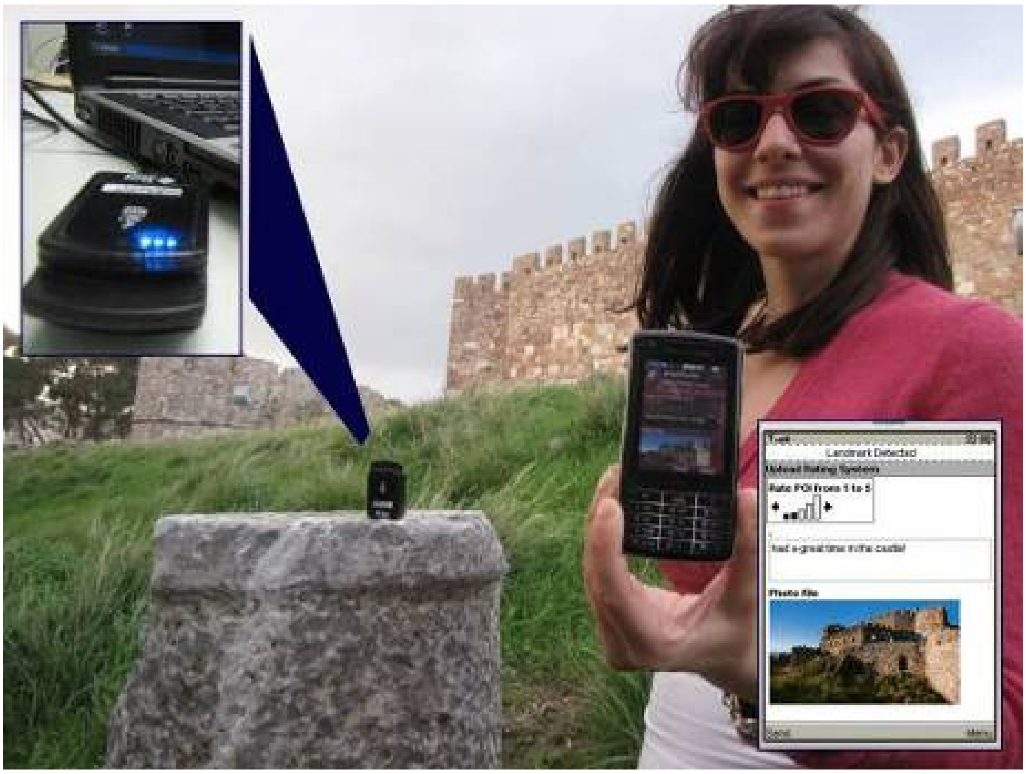


Figure 5-7. A user evaluator uploading content through a deployed WSN infratsucture (the SunSPOT node is shown on the top of the pillar, while the upper left part of the picture illustrates the sink which receives and forwards the tourist content to the remote web server).

5.5. Summary

We have described the design and technical aspects of MTRS, a web-based recommendation system tailored to mobile tourist guides. The primary design objective of MTRS is to deliver personalized services which assist tourists in choosing places to visit, reducing the information burden and cognitive load for the user. Unlike existing systems, MTRS employs collaborative filtering methods to make use of information, behaviours, evaluations or ratings of other tourists with similar interests. Furthermore, MTRS takes into account contextual information such as the user’s current location, time, weather conditions and the recorded history of user’s mobility in providing personalized recommendations.

We have also introduced the concept of ‘context-aware rating’ wherein ratings and evaluations uploaded by mobile users while onsite are considered moiré credible and weighted higher than those uploaded by users via standard web interfaces. We also proposed the use of cost-effective WSN installations around tourist sites for allowing mobile users to upload tourist information and ratings about POIs via their mobile devices. The use of sensor nodes as a means to enabling accurate users’ localization and also allowing users to connect and upload rating and commenting data was seen as promising when a mobile network connection was either unavailable or expensive to use.

Part of this chapter was published in the proceedings of ISCC’2010 with an extended version submitted to Personal and Ubiquitous Computing journal (see Appendix A).

CHAPTER 6

NEAR-OPTIMAL PERSONALIZED DAILY ITINERARIES FOR MOBILE TOURIST GUIDES

6.1. Introduction

Tourists that visit a destination for one or multiple days are unlikely to visit all tourist sights available. Rather, tourists have to manually choose which POIs would be more interesting for them to visit per day. These choices are normally based on information gathered by tourists via the web, magazines, printed tourist guides, friends' suggestions, etc. After deciding which sights to visit, tourists have to determine the route to take, i.e. the directions and the order in which to visit each POI, with respect to the visiting time required for each POI, the POI's opening and closing time and the time available for sightseeing on a daily basis.

Tourists encounter many problems when following this procedure. The information contained in printed guide books is often outdated (e.g. the opening times of some museums might have changed or a site might even be closed due to maintenance works, etc), the weather conditions might be prohibitive during one of the visiting days in which an important POI should have been visited, etc [48]. The selection of the most important and interesting POIs for visiting also requires a fusion of information typically provided from separate -often non credible- sources. Tourists are usually satisfied if a fairly attractive or feasible route is derived, yet, they are not aware of any alternative routes which would potentially be better to follow. Certainly, some tourist guides do acknowledge such problems and try to propose more generalized tourist routes to a city or an area. These routes are designed to satisfy the preferences of the majority of its readers but not those with specialized interests, needs or constraints [35].

Mobile tourist guides may be used as a tool to offer solutions to such problems [34],[107]. Based on a list of personal interests, up-to-date information and information about the users' visit (e.g. date of arrival and departure, accommodation address, etc), a mobile guide can suggest near-optimal and feasible routes that include visits to an ordered list of sights, as well as map directions [172]. Again, those tourist routes do not take into consideration the usage context, e.g. the start or end point of the user, the time span that the user can afford to visit sights, the current time and the predicted weather conditions for the journey, etc. Certainly, taking into account the parameters of context and location brings forward a new challenge for the design of more specific tourist routes [99]. In the past, Kramer et al. [95] had analyzed generic interest profiles and concluded that those profiles analyzed were particularly diverse. This conclusion supports the argument that there exists a need to derive personalized, instead of generalized, tourist routes.

Given a list of sights of a tourist destination where a tourist-user might potentially be interested in visiting, the problem involves deriving the order in which the tourist should visit the selected POIs, for each day the tourist stays at that destination. We define this problem as the 'tourist itinerary design problem' (TIDP). Interestingly, the TIDP presents similar problems as those which have arisen in the past in the field of operational research; such problems reside with the mathematical theory of graphs (graph theory) and comprise variations of the well-known travelling salesman problem (TSP).

For instance, the team orienteering problem (TOP) appoints an initial and final point as well as *N* points for visiting, where each point is associated with a 'score' or 'profit'. Given a particular time margin for each of the *M* team members, the TOP determines *M* routes (from the initial to the end point) via a subset of *N* points, aiming at maximizing the overall profit of visited points [36]. The TOP cannot be solved in polynomial time (NP-complete) [157], hence heuristics deriving near-optimal solutions are the only realistic way to tackle such problems, especially when considering time-critical web applications. TOP can be thought of as a starting point to model TIDP whereby the *M* team members are reduced to the number of days available for the tourist to stay and the profit of a sight signifies the potential interest (or degree of satisfaction) of a particular tourist visiting the POI within a given time span available for daily sightseeing (therefore, TOP considers the time spent while visiting each POI as well as the time needed to travel from one POI to another).

Nevertheless, TOP does not take into consideration the POIs' visiting days and hours. Therein, the resemblance of TIDP with another operational research problem (travelling salesman problem with time windows, TSPTW) [46] comes to bear. TSPTW concerns the minimum cost path for a vehicle which visits a set of nodes. Each node must be visited only once and the visit must occur within an allowed time interval (time window). The correlation of time windows with the POIs visiting days/hours is obvious. However, TSPTW involves planning of only one route (i.e. not *M*, as many as days available to the tourist to visit POIs), while it requires the vehicle to visit the whole set of nodes. A generalization of TOP and TSPTW is referred to as team orienting problem with time windows (TOPTW) [171] and considers multiple vehicles (i.e. itineraries) that should visit a subset of nodes, each within its allowed time window.

The main contribution of this chapter lies in the modelling and in the investigation of a generalization of TOPTW through introducing a novel heuristic that provides near-optimal solutions to TIDP: the Daily TouRist Itinerary Planning (DailyTRIP).

The remainder of this chapter is organized as follows: Some relevant research and commercial projects are discussed in Section 6.2. The modelling, design and implementation of DailyTRIP are presented in Sections 6.3, 6.4 and 6.6, respectively, while Section 6.9 summarizes the chapter.

6.2. Related Work

The issue of personalized tourist itineraries has not been looked at thoroughly in the electronic and mobile tourism literature, with the exceptions of the algorithms proposed in [157] and [158]. The algorithm presented in Souffriau et al deals with TOPTW, however it does not take into account neither the opening days of sites nor the time needed to visit a sight, i.e. it makes the unrealistic assumption of zero visiting duration.

Other relevant research projects with respect to tourist itineraries have been reported in [2], [56]. In [2] a method for deriving a *single* multi-modal tourist itinerary is proposed considering a variety of constraints and following a genetic algorithm-based approach. However, derived route recommendations are not personalized as user preferences about specific types of sites are not taken into account. P-Tour [108], Dynamic Tour Guide (DTG) [68] and City Trip Planner[18] [56] address these shortcomings; however, they derive routes day-by-day. The City Trip Planner currently offers the most advanced route planning functionalities considering several user constraints. Google city tours application[19] represents another interesting approach along the same line suggesting multiple daily itineraries through the familiar Google maps interface. Yet, the suggested itineraries are not personalized. Furthermore, both [2] and [56], city tours implementations are only provided through a web interface and have not been tested on mobiles; hence they lack location-based and context-aware features. On the other hand, P-Tour and DTG have been implemented on mobiles, but can only deal with a small number of POIs (i.e. their scalability is questionable).

[18] http://www.citytripplanner.com/

[19] http://citytours.googlelabs.com/

6.3. Dailytrip Modelling

DailyTRIP modelling involves the definition and the description of the user model, the visit model and the sight (POI) model taking into consideration parameters/ constraints like those listed below:

- **The User Model:**
  - device (e.g. screen resolution, available storage space, processing power, etc);
  - language of content, localization;
  - personal ‘demographic’ data (e.g. age, educational level);
  - interests (explicit declaration or implicitly collected);
  - disability (e.g. blind, deaf, motor disability);
  - budget threshold willing to spend on sightseeing.
- **The Visit Model:**
  - geographical location of accommodation;
  - period of stay (arrival and departure date);
  - time constraints (e.g. the available time each day to tour, the number and the duration of desirable breaks, etc);
  - means of travel (e.g. walking, car, bus, metro, etc).
- **The Sight (POI) Model:**
  - category (e.g. museum, archaeological site, monument, etc);
  - available multimedia resources (collection of texts, video, audio, etc, localized in different languages;
  - geographical position (coordinates);

- weight or 'objective' importance (e.g. the Acropolis of Athens is thought to be 'objectively' more important than the Coin Museum of Athens, hence the Acropolis is assigned a larger weight);
- average duration of visit (e.g. the Archaeological Museum of Athens typically takes longer to visit than the city's Coin Museum due to size differences and the nature of the exhibition);
- average time spent in a queue, before entering the site (this varies depending on day and time, for queues are longer in weekends);
- user rating and comments of other users;
- opening days/hours (time windows), which could be provided by the web service of an administrative body or the Ministry of Culture;
- whether it is an indoor or outdoor site;
- whether it is a accessible for people with disabilities;
- admission price (ticket prices).

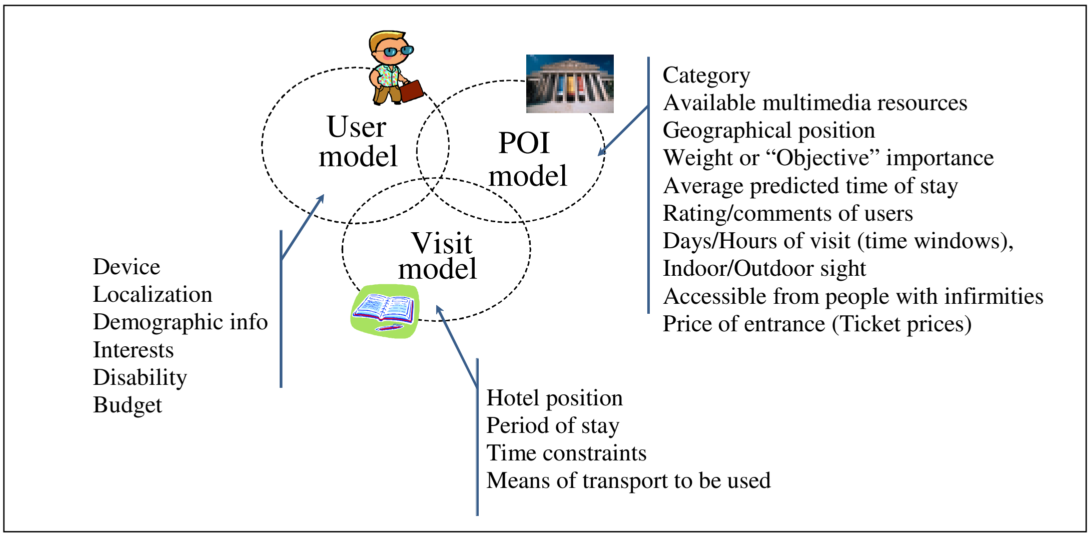

User model
POI model
Visit model
Category
Available multimedia resources
Geographical position
Weight or "Objective" importance
Average predicted time of stay
Rating/comments of users
Days/Hours of visit (time windows),
Indoor/Outdoor sight
Accessible from people with infirmities
Price of entrance (Ticket prices)
Device
Localization
Demographic info
Interests
Disability
Budget
Hotel position
Period of stay
Time constraints
Means of transport to be used

Figure 6-1. Description of user, visit and sight models in TIDP.

The above stated parameters/constraints can be seen in Figure 6-1. Notably they are not exhaustive. From these parameters a list of elements may be derived which include:

- The topological (or Manhattan) distance[20] between the POIs and also between the accommodation and the POIs is based on geographical coordinates and the local map.

- The number of routes that must be generated is based upon the period of stay of the user at the tourist destination.

- The anticipated duration of visit of a user at any given POI is derived from the average user visit duration and its specific user’s potential interest (concluded by examining the user’s profile).

- The possibility of visiting open air sites on a particular day during the user’s visit in regards to meteorological forecasts which can be retrieved from an Internet web service e.g. outdoor sites which are not recommended to visit during a rainy day.

The problem’s definition also includes the ‘profit’ of a POI, calculated as a weighted function of the objective and subjective importance of each POI (subjectivity refers to the users’ individual preferences). Our algorithmic solution maximizes the overall profit by enabling the construction of personalized routes which include the most important sights for each user under specific constraints e.g. opening hours, weather conditions, time available for sightseeing. The most crucial constraint in seeking solid algorithmic solutions is the daily time limit *T* which a tourist wishes to spend on visiting sights; the overall daily route duration (i.e. the sum of visiting times plus the overall travel time spent moving from one POI to another, which is a function of the topological distance) should be kept below *T*.

[20] The Topological distance refers to the shortest path (among the many alternative paths) between two topology nodes.

6.4. *Dailytrip: A Heuristic for Deriving Near-Optimal Personalized Daily Tourist Itineraries*

6.4.1. Problem Statement

The TIDP problem involves a complete graph $G=(V,E)$, $|V| = n$, where each node i,, $i=0,\ldots,n-1$, in V corresponds to a POI and each edge (i,j) in E corresponds to the shortest path (in terms of Topological distance $d_{i,j}$) linking POIs i and j.

Each POI $i \in V$ *is* associated with a weight w_i which denotes the 'objective' importance of the POI and a profit value p_i, which reflects the importance of that POI for a particular user and depends on her personal preferences. Each POI i is also associated with a set of days $D_c(i)$ where visiting the given POI is not feasible (e.g. Mondays, maintenance days and during some bank holidays) and the anticipated visit duration of the user at the POI $t_v(i)$; similarly, the profit, $t_v(i)$ also depends on the user's personal preferences (for instance, someone interested in archaeology is expected to take a longer time visiting an archaeological museum than others).

The cost of each edge (i,j) $c_{i,j}$, namely the cost of visiting j after visiting i, is a weighted function of travelling time from i to j , $t_{i,j}$ (the latter depends on the Topological distance $d_{i,j}$ between i and j and the means of travel), the profit of the arriving node p_j and the duration of visit at the arriving node $t_v(j)$: $c_{i,j} = a_1 \cdot t_{i,j} - a_2 \cdot p_j + a_3 \cdot t_v(j)$, where a_1, a_2 and a_3 are weight coefficients. This formula signifies that being on node i, the next itinerary stop j should be a node of relatively high profit that takes a short time to arrive and has little time to visit. Notably, $c_{i,j} \neq c_{j,i}$.

Travellers typically plan to visit the destination for a set of days, D. Users also define a starting and ending time per day for their daily itineraries, T_{start} and T_{end}, which denote what time the user prefers to depart from the starting point S and arrive at the end point (destination) E. Hence, a daily time budget devoted to visiting sights may be easily calculated: $T = T_{\text{end}} - T_{\text{start}}$. Without loss of generality, we assume that that the starting and end points of the $|D|$ daily itineraries coincide, i.e. $S \equiv E$ (typically these will coincide with the user's accommodation H).

In summary, the objective of DailyTRIP is to derive $|D|$ itineraries I_i that maximize the overall profit $\sum_{i=1}^{|D|}\sum_{j=1}^{|I_i|} p_j$, ensuring that the time needed to complete each itinerary does not exceed the user-defined daily time budget T, i.e. $T(I_i) \leq T$.

6.5. The Dailytrip Algorithm Flow

The DailyTRIP algorithm is comprised of the following execution phases:

Phase 1: Definition of the problem's model

The first phase first involves the definition of the problem space, i.e. the nodes of G, the nodes' weight w_i and the travelling time matrix $t_{i,j}$ that denotes the time needed to travel between node pairs; notably, $t_{i,j} \neq t_{j,i}$, since the route i=j differs from the route j=i due to the use of a geographical map where one-way roads apply (hence, the graph G is non-symmetric). Taking into consideration personalization issues (e.g. in a simplified scenario, user preferences about POIs' categories), the cost matrix (i.e. the cost values $c_{i,j}$ associated with the two-directional edges) as well as the nodes' profit p_i and visit duration $tv_{(i)}$ with respect to a specified user are also computed.

Phase 2: Reduction of the problem's space

The initial set of sights around the tourist destination are sorted in decreasing order of profit p, where the value of p mainly depends on its category (i.e. whether the POI is a museum or an architectural monument) and the user's preference about this category. To reduce the computational effort required to reach valid solutions (i.e. to reduce the problem's space) we discard:

- nodes (POIs) with profit p_i smaller than a threshold value $p_{\min}$
- POIs located too far from the origin point *H*, i.e. every node *v* for which $t_{H,v} > t_{\max\square}$, where $t_{\max\square}$ is an upper time limit (see Figure 6-2a).

An alternative approach would be to exclude the relatively low-profit POIs located far from *H*, i.e. exclude every POI *i* for which $a_1 * p_i - a_2 * d_{i,H} < t$, where a_1 and a_2 are weight coefficients and *t* a threshold value.

Phase 3: Selection of first daily itinerary nodes

DailyTRIP determines the |*D*| POIs that will be the first to include in the |*D*| daily itineraries I_i, where *i* =1..|*D*|. We select the set of |*D*| nodes {N_i}, where *i* =1..|*D*|, located furthest apart from one another, i.e. those for which the minimum distance from one another is the maximum among any other permutation of |*D*| nodes. For instance, in the example topology of Figure 6-2b, assuming that |*D*|=3, we select the nodes *i*, *j* and *k* that: $\max_{i,j,k} \min\{d_{i,j}, d_{i,k}, d_{j,k}\}$. Then, the |*D*| daily itineraries are initialized, each incorporating one of those nodes: $I_i = \{N_i\}, \forall i = 1..|D|$. The philosophy behind this approach is to achieve a geographical 'segmentation' of the tourist destination as it makes sense, for example, for a tourist that visits a city for two days, to focus on POIs located on the northern part of the city on her first day and on POIs located on the southern part on her second day.

Phase 4: Construction of itinerary trees

On each of the following algorithm's steps, itineraries I_i are considered interchangeably incorporating a new node *N* not yet included in any of the I_i. In particular, for each I_i, the candidate node *N* with the minimum connection cost $c_{j,N}$ to any of the nodes $j \in I_i$ joins I_i (through accepting the *j* □ *N* edge), given that the daily time budget *T* condition is not violated for this itinerary. Notably, as the candidate node *N* may be connected to any of the I_i nodes (i.e. not necessarily to the edge nodes of the itinerary), I_i grows as a tree structure rather than a multipoint line. The time T_i corresponding to the completion of the itinerary I_i is calculated first by temporarily connecting *H* with the I_i node nearest to *H*, then converting the I_i itinerary tree to a multipoint line (through a heuristic TSP algorithm) and finally calculating: $T(I_i) = t_{H,1} + \sum_{k=1}^{|I_i|} \llbracket (t \rrbracket_v(k) + d_{k,k+1})$. Namely, $I_i = I_i \bigcup N$, if $T(I_i) \leq T$.

Hence, on each step the itineraries I_i grow, typically approaching the start/end point *H*, until no further insertion is feasible (see Figure 6-2c). Upon completion, each itinerary is connected to the 'hotel' node *H*, i.e. the edge *j*□*N* is accepted, where *j* is the itinerary's node nearest to *H* (see Figure 6-2d).

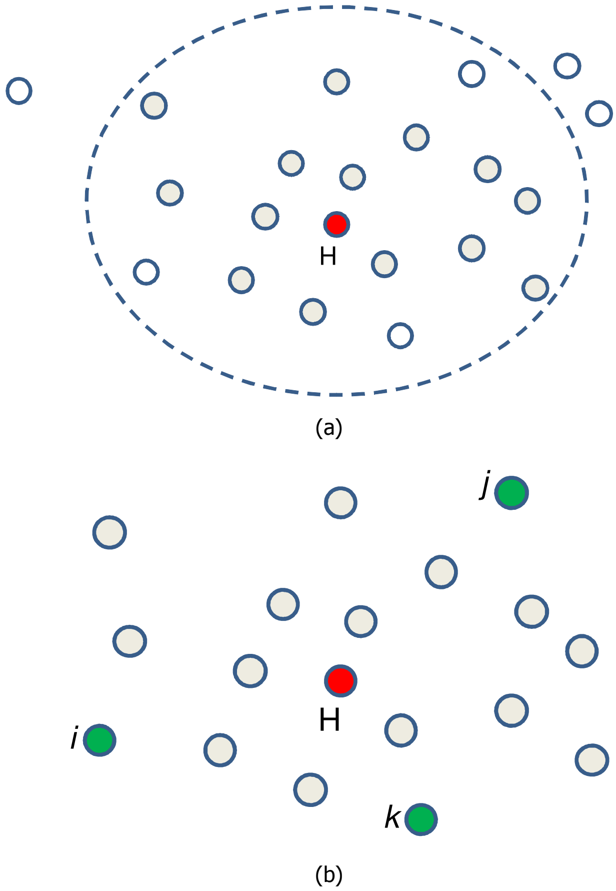
It is noted that the acceptance of candidate nodes also depends on the corresponding POIs' scheduled visiting days. In particular, for each joining node *i* that may not be visited during the days $D_c(i)$, the 'excluded' days of the itinerary *I* joined by *i* is adapted excluding those days: $D_c(I) = \bigcup_{i=1}^{|I|} D_c(i)$, signifying that during those days the itinerary is not feasible either. Apparently, a POI *i* may join an itinerary *I* if the intersection of their valid days (i.e. those when visiting is feasible) is not null and also this intersection includes at least one of the *D* days of visit, namely if $D_v(I) \bigcap D_v(i) \bigcap D \neq \emptyset$.

Phase 5: Rearrangement of itinerary trees

Phase 5 is optional and aims to improve the solutions derived in the previous phase, i.e. either increasing the overall profit or maintaining the same profit while reducing the itinerary completion time $T_{(I)}$ (see Figure 6-2d). Improved solutions are searched for every itinerary by: (a) substituting each itinerary tree node for any node not included in any itinerary at the end of the previous phase, (b) by swapping nodes included on different itineraries. In any case, the new itinerary solutions should satisfy the daily time budget constraint.

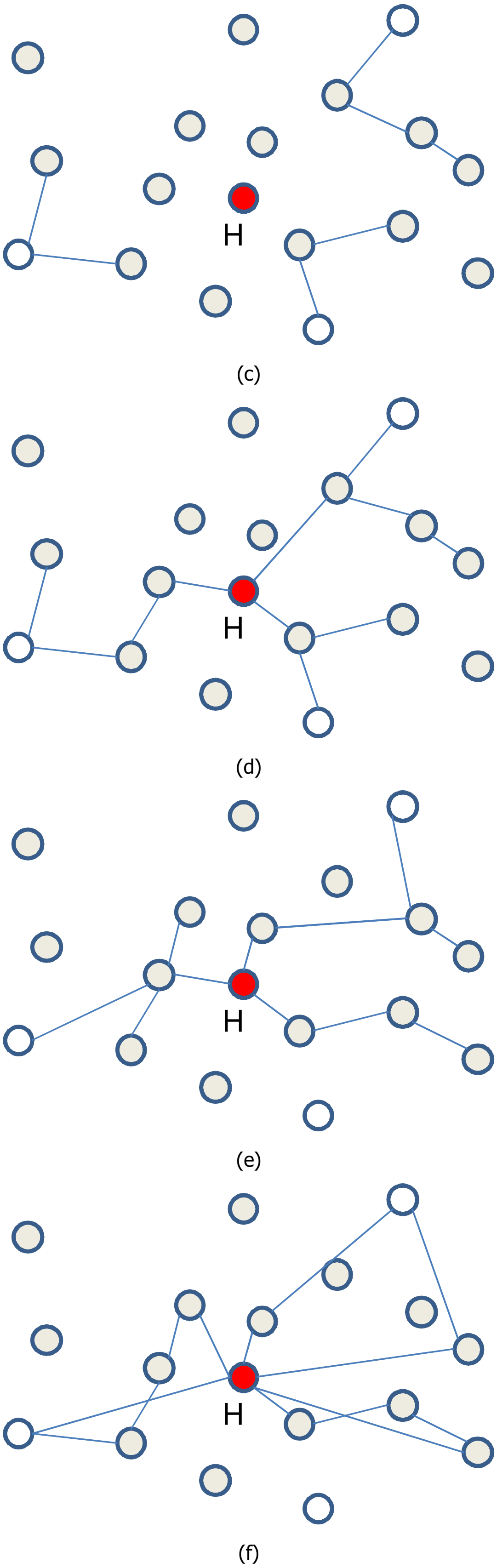
Phase 6: Traversal of itinerary trees

Notably, the outcome of the previous phases is not a set of itineraries, but rather a set of itinerary trees. Hence, the last phase of DailyTRIP involves the conversion of the $|D|$ trees to multipoint lines I_j through a heuristic TSP algorithm of the corresponding trees.



(a)

(b)



(c)

(d)

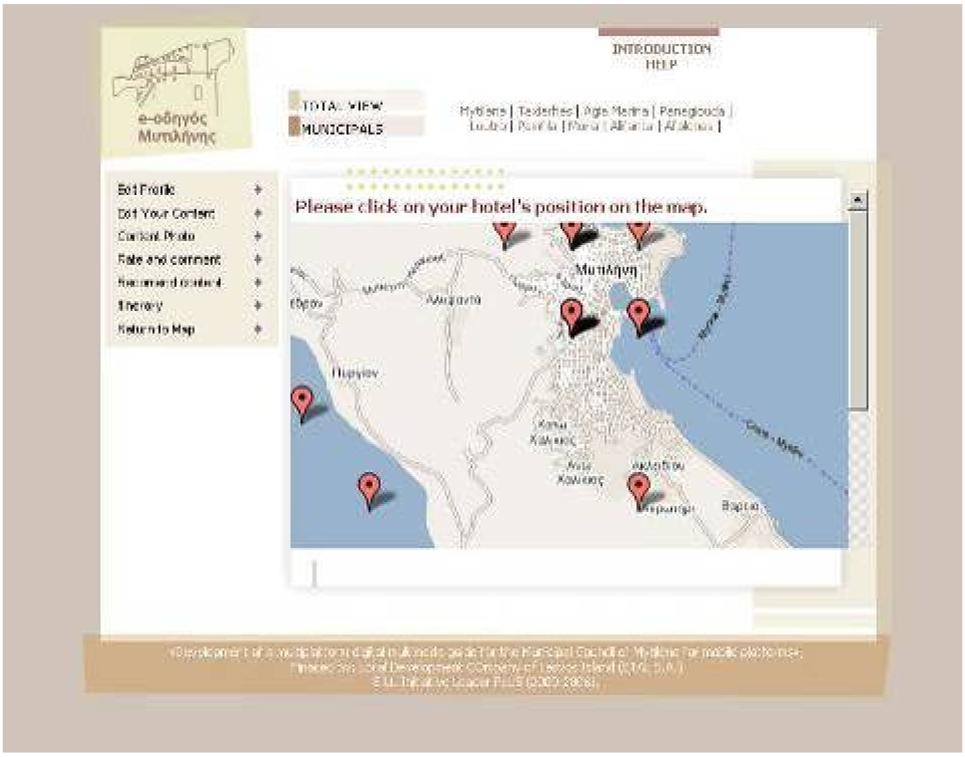
(e)

(f)

Figure 6-2. Execution phases of DailyTRIP.

6.6. Implementation Details and Evaluation of Dailytrip

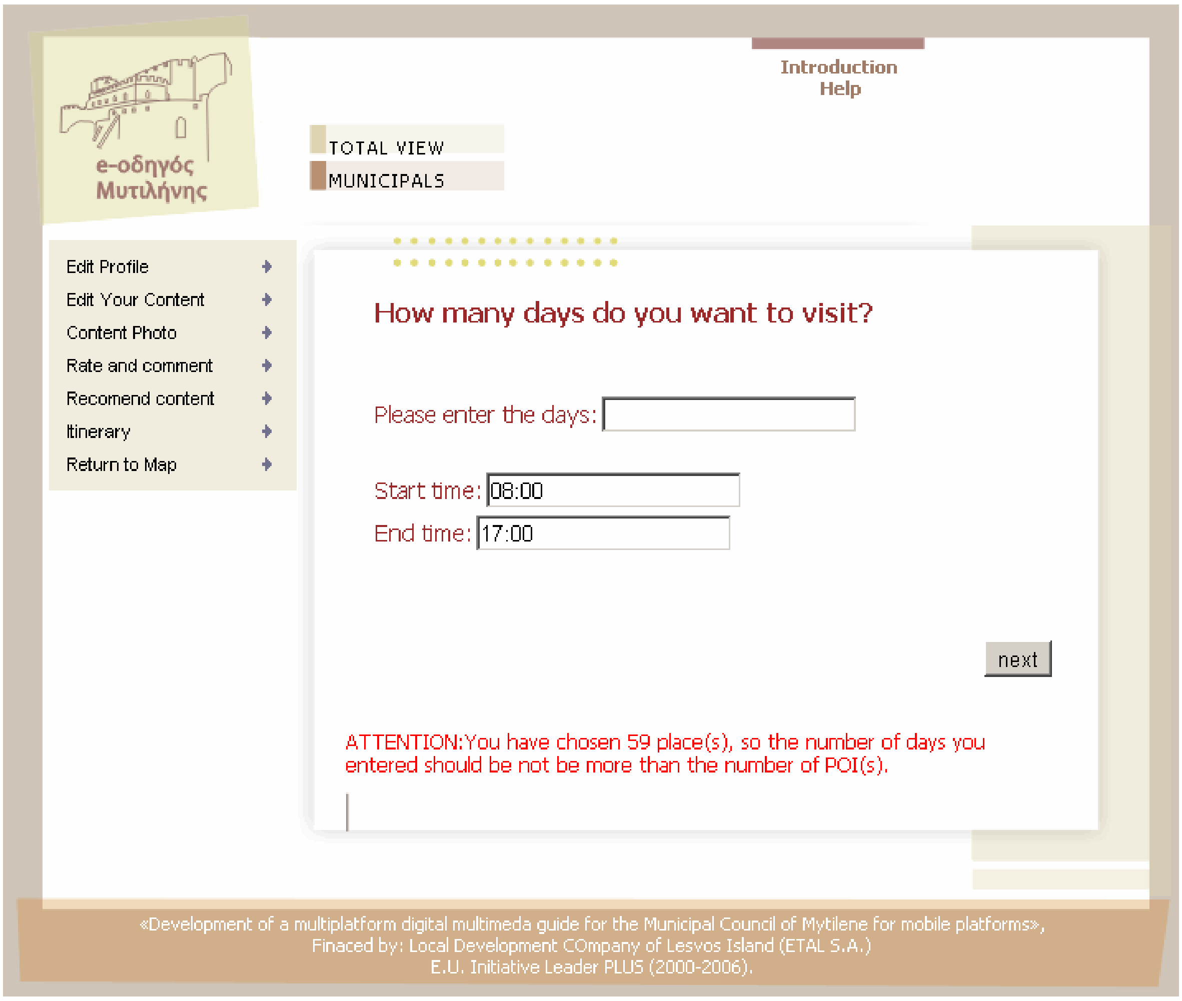
DailyTRIP has been developed using JSP/MySQL web technologies and Google Maps as the main user interface. The web pages are accessed via the profile pages of the web application of the Mytilene e-guide system discussed in Chapter 4 and Chapter 5. The user's profile pages have been created in the process of the "Download now" section. In this process the user has created a personal account (explicit profile), has already selected some content (implicit profile), and has been recommended POIs via the recommendation system already implemented on this system, i.e. personal interests parameters to selected POIs and to categories of POIs have been formed.



Please click on your hotel's position on the map.

Figure 6-3 Screen 1: Select hotel position

When the user selects the Itinerary menu (see Figure 6-3), the user is prompted to click the location *H* of their accommodation on the map, and following is prompted to enter the period of visit, the hours available to visit POIs each day etc (see Figure 6-4). The following screens the user can select the means of transport which the user will be using (as such only Car or Walking is available now) and the radius around the hotel she is willing to move in order to visit a POI (this depends upon the means of transport chosen).



How many days do you want to visit?

Please enter the days:

Start time: 08:00

End time: 17:00

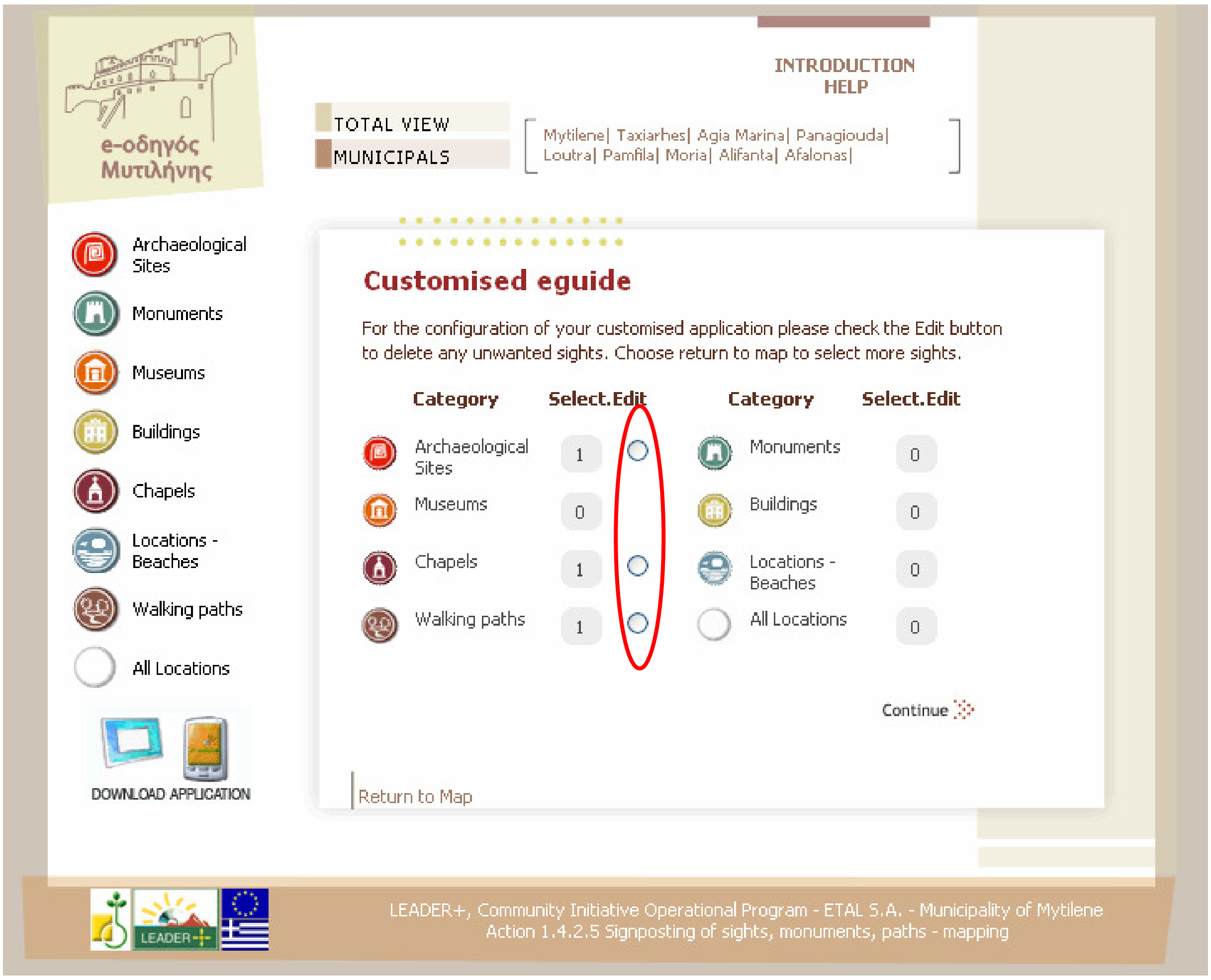
ATTENTION:You have chosen 59 place(s), so the number of days you entered should be not be more than the number of POI(s).

«Development of a multiplatform digital multimeda guide for the Municipal Council of Mytilene for mobile platforms», Finaced by: Local Development COmpany of Lesvos Island (ETAL S.A.) E.U. Initiative Leader PLUS (2000-2006).

Figure 6-4. Example of user data input screens

The user is then shown a list of the initially selected POIs based on her preferences, which the user is allowed to modify by adding or removing POIs (see Figure 6-5). As such, at any given time, the user has access to the personal profile pages to edit POI list and review the entered personal preferences.

The algorithm filters the POIs left out from the problem's space (due to the distance from the user's accommodation, their incompatibility with the user's preferences or the intentional removal by the user) and populates the travelling time matrix for the remaining nodes through first computing the distance matrix entries and considering the average expected velocity v of the selected means of transport. Distances amongst pairs of nodes are found by means of using the shortest-route functionality of the Google Maps API[21] [64] which refers to Topological distances and takes into account one-way roads.



Customised eguide

For the configuration of your customised application please check the Edit button to delete any unwanted sights. Choose return to map to select more sights.

Category	Select.Edit		Category	Select.Edit
Archaeological Sites	1	○	Monuments	0
Museums	0		Buildings	0
Chapels	1	○	Locations - Beaches	0
Walking paths	1	○	All Locations	0

LEADER+, Community Initiative Operational Program - ETAL S.A. - Municipality of Mytilene
Action 1.4.2.5 Signposting of sights, monuments, paths - mapping

Figure 6-5 How the user can choose POI category (a)and delete (b) POIs from their selection

Our implementation is based on the following assumptions: (a) each daily itinerary starts and ends at the same node, which coincides with the user's accommodation; (b) among all possible routes between a pair of nodes we only consider the shortest route in terms of length, although this might not be shortest in time; (c) the user is assumed to move with constant velocity regardless of the traversed edge or the time of day (admittedly, this is a valid assumption only for

[21] To determine the best route to a destination, Google map's routing software borrows from graph theory and employs a variation of Dijkstra's Algorithm. This algorithm solves the problem of finding the shortest paths from any point to all other points on a weighted, directed graph. The algorithm starts at the source vertex and grows a tree that eventually spans all nodes reachable from the starting node. The vertices are added to the tree in their order of distance. That is, the closest node is added first, followed by the next closest, and so on. Part of the formula is a relaxation procedure that continually updates the distance costs of all the vertices. It checks whether the current best estimate of the shortest distance can be improved by going through a different node.

tourists walking around a city); (d) the POIs are assumed to be open for visiting during the hours available to the tourist for sightseeing.

As stated in Section 6.5, phase 6 of DailyTRIP involves the conversion of the |D| trees to multipoint lines I_i through a heuristic TSP algorithm. The latter is based on the Lin-Kernighan heuristic [69] and was implemented in Java. The Lin-Kernighan heuristic is considered as one of the most efficient algorithms for deriving near optimal solutions for the symmetric travelling salesman problem.

6.6.1. The Java Application

The Java application which implements the above aforementioned algorithm comprises four main components: the database containing the list of possible POIs or nodes, a backend library that performs the calculations, API calls to the Google Maps service, and a front-end user interface implemented as a Java web application. The open-source integrated development environment (IDE) Netbeans 6 [124] was used to perform this development.

6.6.2. Process Flow

Figure 6-6 provides a high-level overview of the system components. The process flow of the web application is described below the diagram.

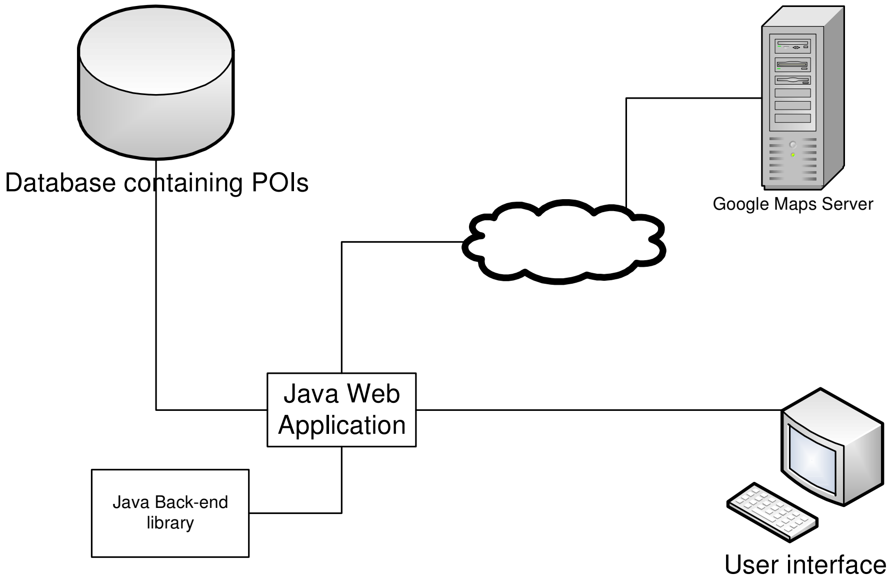


Figure 6-6 High level diagram of application system architecture.

- The Java Web Application uses the Google Maps API (application programming interface) to display a map of the area of interest. It does so by means of a web call to Google Maps, using JavaScript and Ajax [75] in the web page that the Web Application presents to the end user. The user, by means of a web browser (see Figure 6-3), clicks on the map to indicate her position of origin (e.g. her hotel.).

- The Java Web Application obtains from the database the list of POIs selected by the user, along with the user-dependent weighting of each POI.

- Using the Google Maps API, the Java Web Application creates the distance matrix. This is done by reading the coordinates of each of the user's chosen POIs from the database, and for each POI and the origin as the nodes, by calling the Google Maps API to calculate the shortest route for each pair of nodes.

- The algorithm described in section 6.3 is implemented by a set of Java classes that reside in a Java Back-end Library. These classes perform the calculations based on the user's chosen POIs, as well as the distance matrix derived by Google Maps. It returns the calculated routes as an array to the Java Web Application, which displays it to the user. (see Figure 6-9)

6.6.3. Java Back-End Library

The Java Back-end Library contains a collection of Java classes that implement the algorithm detailed in a previous section of this document. The UML [130] class diagram below (see Figure 6-7) contains the classes in the back-end library.

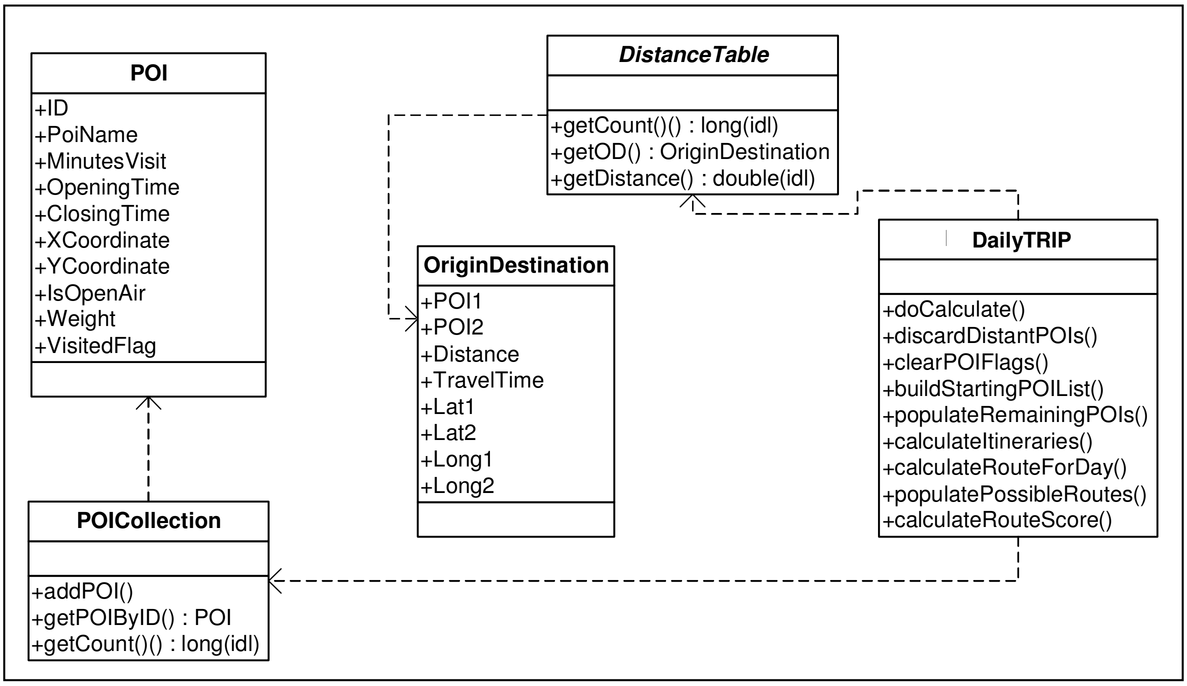


Figure 6-7. UML Class diagram of the itinerary system

The main classes of the Java Back-end Library are the following:

- POI: encapsulates a single node or place of interest.
- POICollection: includes a list of places of interest
- OriginDestination: includes a single origin-destination pair of nodes, with their coordinates and the travel time between them.
- DistanceTable: includes the list of Origin Destinations calculated by Google Maps.

- DailyTRIP: this is the main class that implements the DailyTRIP algorithm.

6.6.4. Java Web Application

The web application runs on an Apache Tomcat web server. It contains the web pages (Java Server Pages) [85] with which the user interacts with the system, as well as embedding the Google Maps calls. Google Maps' API is called by means of JavaScript calls embedded in the JSP pages. The diagram below (see Figure 6-8) illustrates the application flow by means of the Java Server Pages.

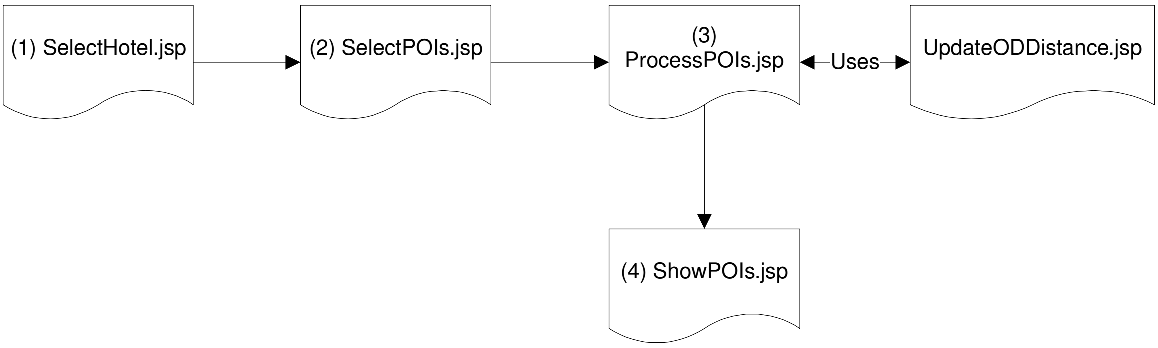
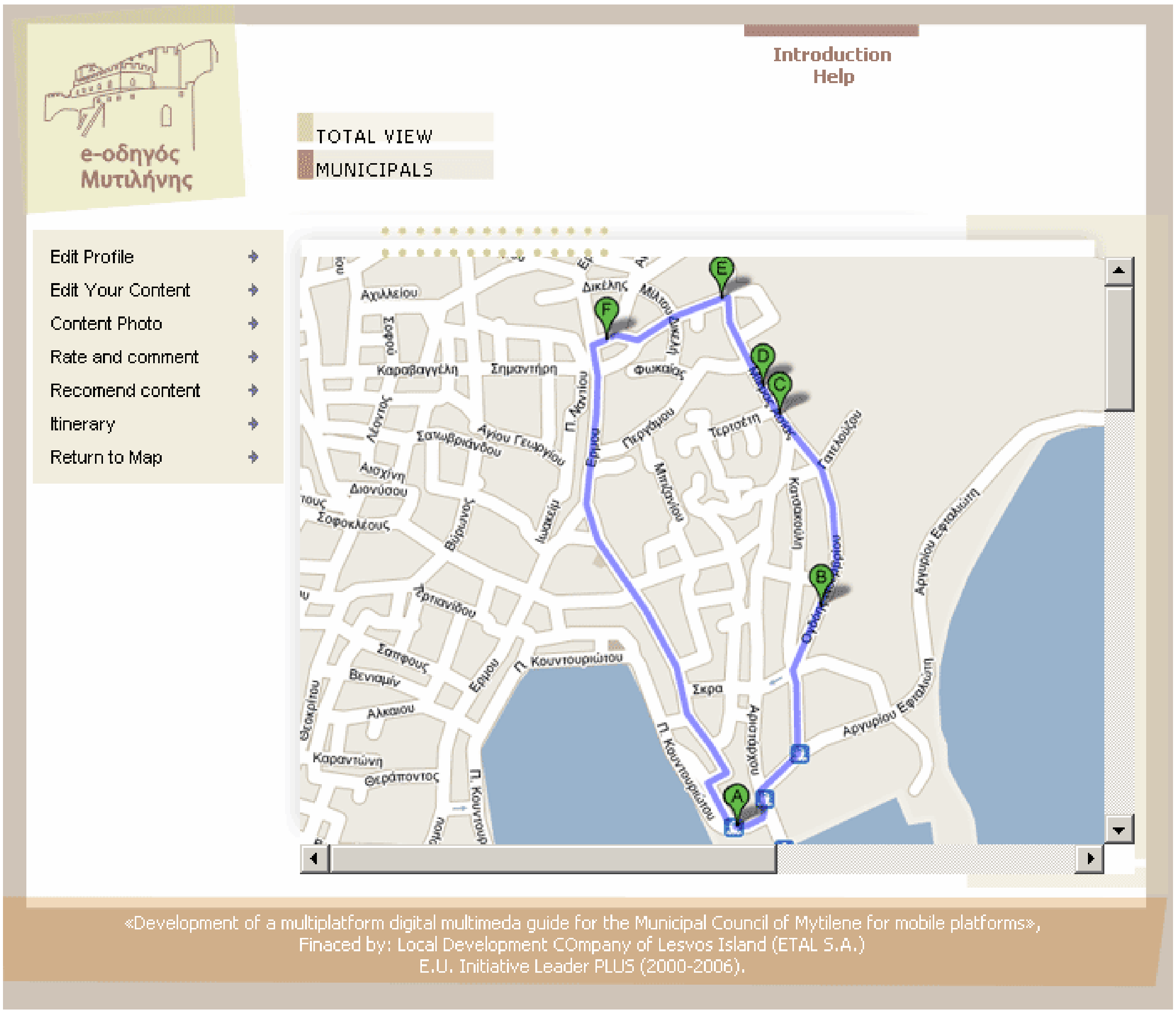


Figure 6-8. Application flow diagram of the DailyTRIP algorithm implementation

Jsp page	Functionality
SelectHotel.jsp	this displays a Google Maps map of the area of interest, and expects the user to click on the position of his hotel or starting point and then the user can also modify the proposed length of stay used for input parameters
SelectPOIs.jsp	this page reads the list of places of interest from the database, and allows the user to modify the POI list she would like to visit by means of clicking checkboxes next to each place of interest. The user can also view and re-assign then importance of the calculated weight of each POI
ProcessPOIs.jsp	is a JSP page that contains an embedded Google Maps map. It uses the list of POIs selected by the user in the previous page and creates a distance table by using the Google Maps API calls for calculating shortest routes. Once all the selected POIs have been processed, it calls the DailyTRIP (algorithm) class in the Java Backend Library in order to perform the calculations according to the algorithm described in Section 6.3.
UpdateODDistance.jsp	this is a helper JSP page that is called by ProcessPOIs.jsp which populates the distance table. It is called by ProcessPOIs.jsp as Google Maps calculates each distance between each origin-destination pair
ShowPOIs.jsp	this is the final JSP page that displays the results of the algorithm’s calculations to the user in a tabular format

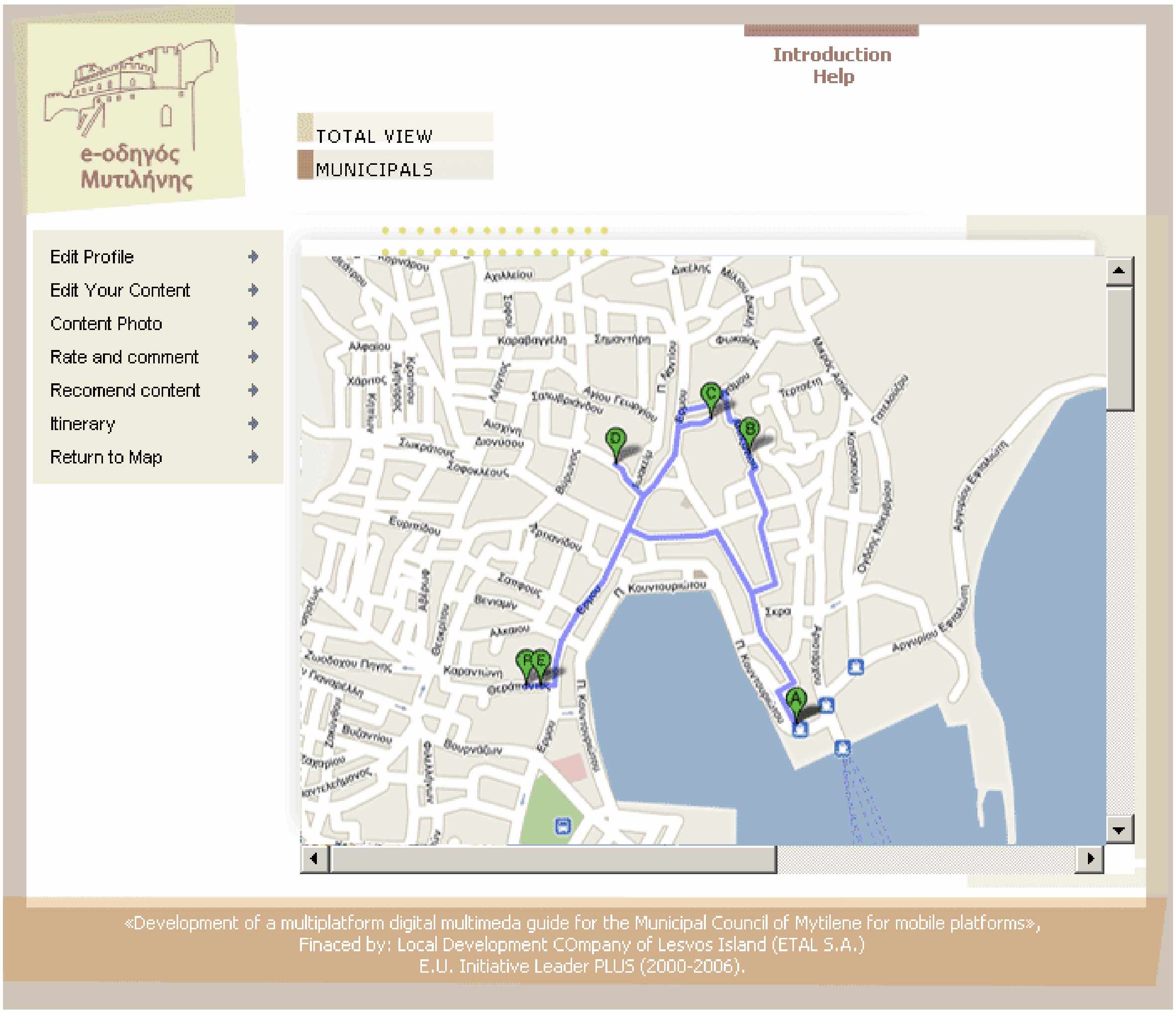
Table 6-1. Table of jsp page and of functionality description.

The output of DailyTRIP is sketched on a Google Maps interface, with each itinerary drawn on separate screen and the order of visiting POIs denoted by the alphabetical order of characters representing POIs (see Figure 6-9). The maps derived by the web application are then converted to static images using the Google Static Maps API [65] in order to display on mobile phone screens.



«Development of a multiplatform digital multimeda guide for the Municipal Council of Mytilene for mobile platforms»,
Finaced by: Local Development COmpany of Lesvos Island (ETAL S.A.)
E.U. Initiative Leader PLUS (2000-2006).

(a)



«Development of a multiplatform digital multimeda guide for the Municipal Council of Mytilene for mobile platforms»,
Finaced by: Local Development COmpany of Lesvos Island (ETAL S.A.)
E.U. Initiative Leader PLUS (2000-2006).

(b)

Figure 6-9. Output of DailyTRIP for two daily itineraries (a –day 1 b –day 2)on Google Maps (point 'A' denotes the user's accomodation location, i.e. the start/end point of the two itineraries).

6.7. ILS algorithm

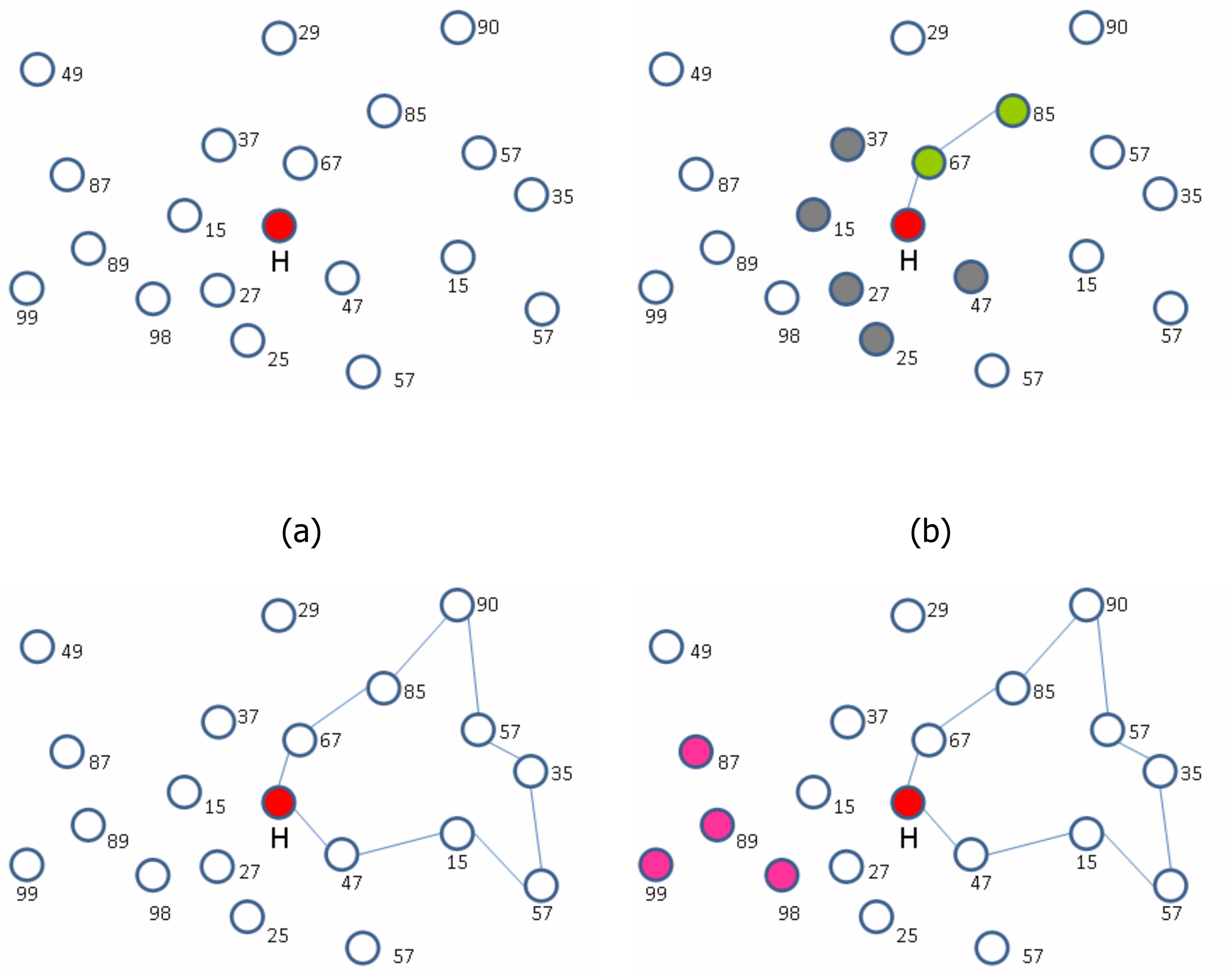
In order to have some basis of comparison for the DailyTRIP algorithm, an alternative algorithm -also dealing with the TIDP- called the ILS algorithm [157], has been implemented and integrated within the Java application discussed above. Therefore, ILS serves as a benchmark against which to compare the results of DailyTRIP for the same topologies.

The same input information, i.e. nodes to be visited, opening and closing times are fed to both DailyTRIP and ILS. Then they both derive output that specifies the route to be followed after their respective sets of calculations.

The ILS algorithm and the DailyTRIP algorithm differ substantially. DailyTRIP is based on trees for consolidating the POIs that belong to the same itinerary and then converts each 'itinerary tree' to a near-optimal route after running a TSP algorithm. In contrast, ILS uses a "greedy" approach [24], wherein the non-selected POI with the greatest score is chosen as the most suitable to be appended to the itinerary, thus removing it from the list of available candidate POIs.

Using this approach, the ILS algorithm does not have to go through a large number of inspections of combinations and permutations (it only considers links deriving from the last inserted node), but it fails to examine possibly more suitable candidate nodes in the global perspective. Thus, ILS represents a fair compromise in terms of speed versus deriving routes that approximate optimality.

Initial tests have shown that DailyTRIP derives better solutions in terms of the overall collected profit, while not significantly surpassed in terms of performance (i.e. execution time required to derive a valid solution). This is because ILS uses the 'greedy' approach, wherein the POI with higher profit is iteratively chosen to join an itinerary. Thus, ILS quickly spends the available daily time budget and thereby cannot afford to include important ('profitable') POIs within the itinerary solutions. Figure 6-10 illustrates an example of the ILS's greedy approach. In this example ILS tends to select POIs with high weight values located in proximity to the hotel. As seen in Figure 6-10d though, the path eventually derived by the ILS overall has neglected 'important' POIs (with high profit values). In contrast, DailyTRIP suggests a compromise in the trade-off between performance and deriving improved solutions.



(a)

(b)

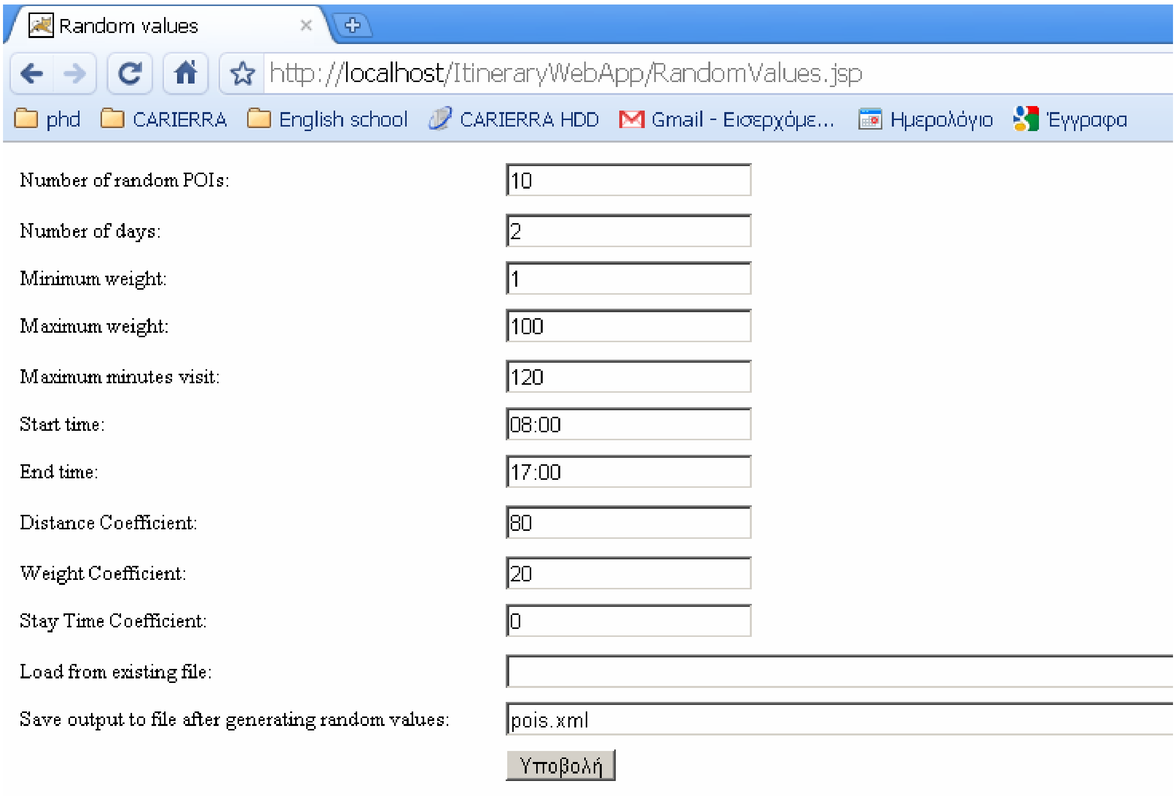
(c)

(d)

Figure 6-10. Execution Phases of the ILS showing a non global optimum using the greedy approach.

6.8. *Simulation Comparison between DailyTRip and ILS algorithms*

In order to perform extensive tests over ILS and DailyTRIP algorithms, we have implemented a web simulation tool in Java which takes into account all the TIDP parameters discussed in Section 6.4.1 and includes a visual interface for displaying the step-by-step output of the examined algorithms. To ensure a fair comparison, we have fed the algorithms the same input, i.e. they have been always executed over the same sets of POIs. The POIs as well as the itineraries' start/end points are placed at random locations on the plane. Similarly, the weight and the anticipated visiting time for each POI is randomly set (although the user may determine minimum/maximum permitted values).



Number of random POIs:	10
Number of days:	2
Minimum weight:	1
Maximum weight:	100
Maximum minutes visit:	120
Start time:	08:00
End time:	17:00
Distance Coefficient:	80
Weight Coefficient:	20
Stay Time Coefficient:	0
Load from existing file:	
Save output to file after generating random values:	pois.xml

Figure 6-11. A screenshot of the web-based simulation tool input screen

The web-based simulator allows easy specification of simulation parameters (see Figure 6-11) and graphically illustrates the output of the DailyTRIP and ILS algorithm, while also recording their respective overall itinerary length, visit order of POIs and respective travel times. It also takes into account a number of constraints, for instance the attributes a_1, a_2, a_3 that determine the weight of travelling time, visiting time and profit of POIs in the calculation of the cost matrix (stated in Section 6.4.1), the number of days to visit, the minimum/maximum weight to assign to each POI, the maximum time to visit a POI (stay time) and the disk location to save the xml topology file (so as to be able to repeat later on the simulation test upon the same topology).

```xml
<?xml version="1.0"?>
<pois>
<days count='2'/>
<start time='08:00'/>
<end time='17:00'/>
<poi name='Hotel' x='3379.8986542959815' y='11498.919247832173' weight='78.0' minutes='48' id='0'/>
<poi name='2' x='5422.925558645952' y='9090.505878937429' weight='31.0' minutes='91' id='1'/>
<poi name='3' x='10561.667683071331' y='13633.734552278185' weight='46.0' minutes='58' id='2'/>
<poi name='4' x='4049.5968576216087' y='6399.585253160639' weight='69.0' minutes='68' id='3'/>
<poi name='5' x='8788.997625381335' y='10587.909482438223' weight='95.0' minutes='47' id='4'/>
<poi name='6' x='6241.733435952213' y='9521.692996680576' weight='65.0' minutes='72' id='5'/>
<poi name='7' x='15503.97033912635' y='11783.23072728515' weight='50.0' minutes='67' id='6'/>
<poi name='8' x='14443.7807671722' y='13929.415697438204' weight='6.0' minutes='1' id='7'/>
<poi name='9' x='5441.18047522093' y='688.980527050795' weight='21.0' minutes='106' id='8'/>
<poi name='10' x='11410.772750421567' y='4310.141532688469' weight='97.0' minutes='30' id='9'/>
<poi name='11' x='7427.801337454328' y='2992.731559982257' weight='77.0' minutes='66' id='10'/>
<poi name='12' x='13419.038339165998' y='1104.5289525692076' weight='19.0' minutes='81' id='11'/>
<poi name='13' x='10020.559451215977' y='1147.1121998892677' weight='17.0' minutes='4' id='12'/>
<poi name='14' x='8845.075549860741' y='5202.199491882018' weight='63.0' minutes='6' id='13'/>
<poi name='15' x='14042.130489377772' y='7855.052952481051' weight='51.0' minutes='61' id='14'/>
<poi name='16' x='10175.08657437544' y='11357.158514537821' weight='86.0' minutes='114' id='15'/>
<poi name='17' x='14079.443990860329' y='7056.007533539546' weight='60.0' minutes='117' id='16'/>
<poi name='18' x='3507.6808926235685' y='8885.442747934288' weight='28.0' minutes='84' id='17'/>
<poi name='19' x='15006.4027252456' y='2975.6492641378854' weight='77.0' minutes='77' id='18'/>
<poi name='20' x='15612.237605818558' y='4207.49328571844' weight='78.0' minutes='55' id='19'/>
<poi name='21' x='1593.5185632870184' y='5801.5475708334225' weight='95.0' minutes='56' id='20'/>
</pois>
```

Figure 6-12. XML excerpt of the file output of the derived topology.

Figure 6-12 shows an excerpt of the XML output file which records the days count, representing the number of days the user has stated they will be visiting, the first POI represents the user’s hotel and each POI has been recorded with the X, Y coordinates, weight (profit) and visiting time (in minutes).

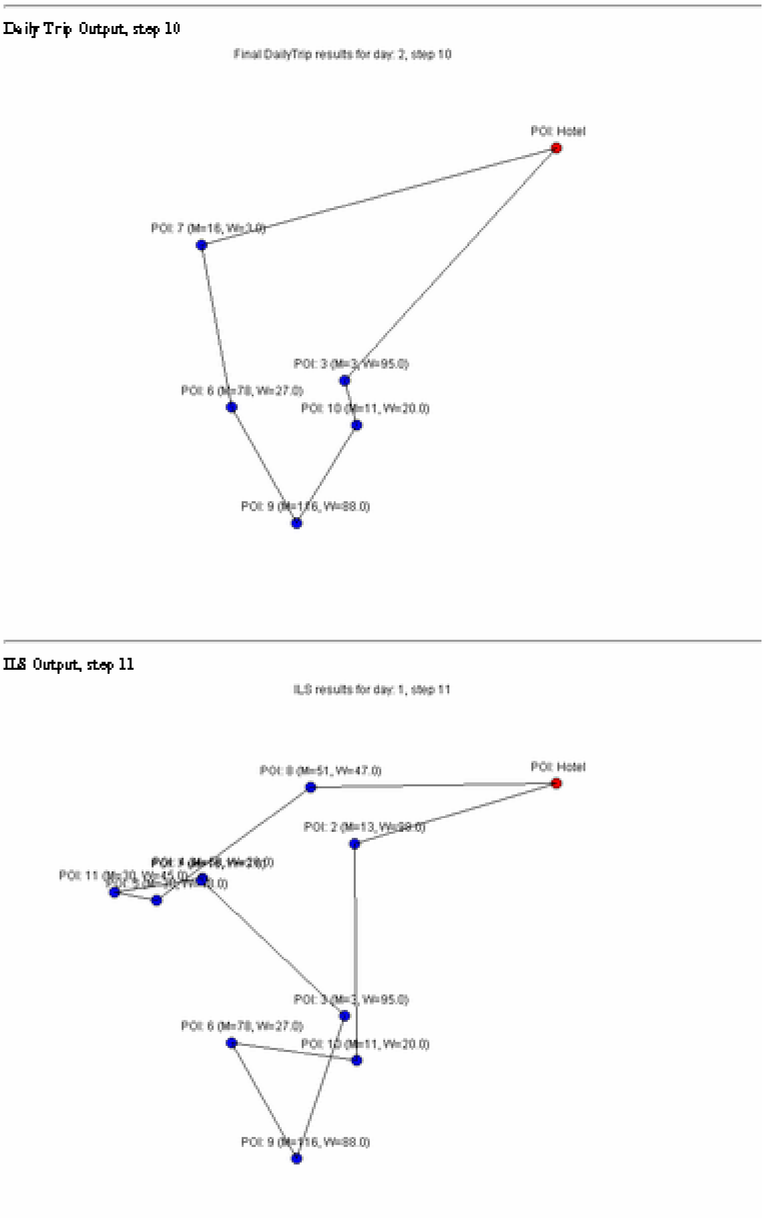


Figure 6-13. The web based simulation tool graphically simulates the algorithms output.

Unless otherwise stated, in all our simulation tests, the weight of each POI is a random number between 1 and 100, the maximum visiting time a user can spend at a POI is 60 minutes and each daily itinerary starts at 08:00 and ends at 17:00 (i.e. duration of 9 hours). In order to denote a higher importance in regards to distance, the distance coefficient has been set to 20% (0,2), while the profit coefficient was set to 80% (0,8) in order to increase the importance of the profit in the POIs selection process. As a result, the visiting time coefficient was set to 0. The main motivation for setting those values has been to come to agreement with the implementation of ILS algorithm, which does neither take into account the distance among POIs nor their corresponding visiting time (it only considers POIs' profit when designing the itineraries - see Section 6.4.1). The simulation results presented herein have been averaged over ten simulation runs (i.e. for ten different network topologies) to ensure their statistical validity.

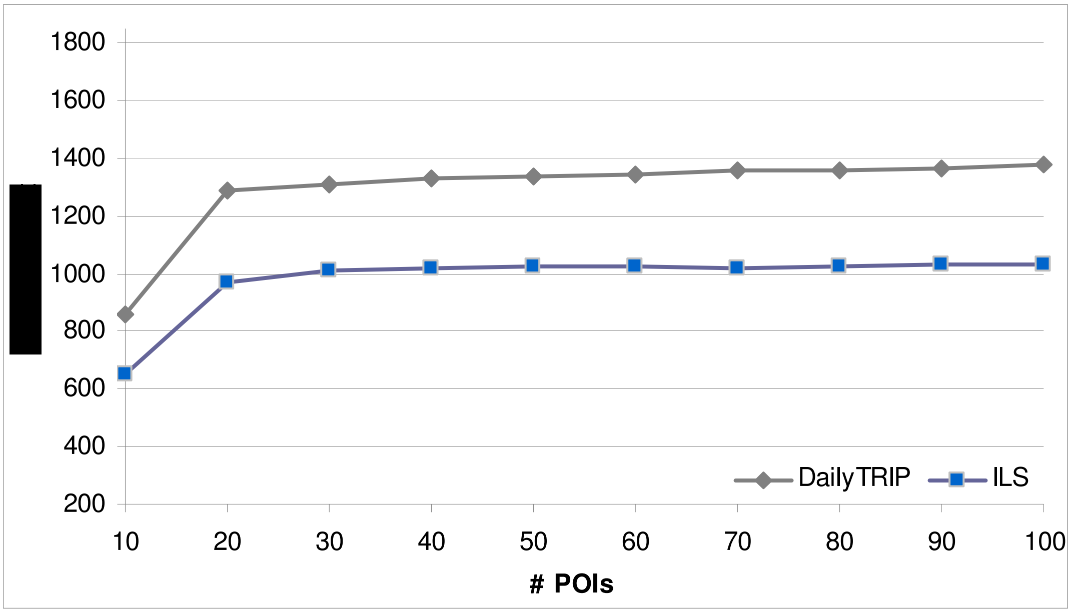


Figure 6-14. Comparison of the overall profit of DailyTRIP vs ILS Algorithm for 2 itineraries.

Figure 6-14 illustrates the overall profit for the DailyTRIP in comparison to ILS Algorithm as the number of POIs increase. For this simulation we assumed 2 days of stay (i.e. two itineraries). The overall profit of DailyTRIP is shown to improve the solutions obtained from ILS. This is basically due to the 'greedy' nature of ILS, as detailed in 6.7. It should be stressed that the total itinerary profit of DailyTRIP very much depends on the value of profit coefficient, that is decreasing the profit coefficient results in decreased total profit (it signifies that the profit of POIs to visit becomes less important) and vice versa. Both algorithms' curves increase more mildly when crossing a threshold of around 20 POIs. This is because no more POIs may be accommodated within the 2 itineraries (typically a maximum of around 10 POIs per itinerary), however larger topologies create opportunities to substitute some POIs with others having larger profit values.

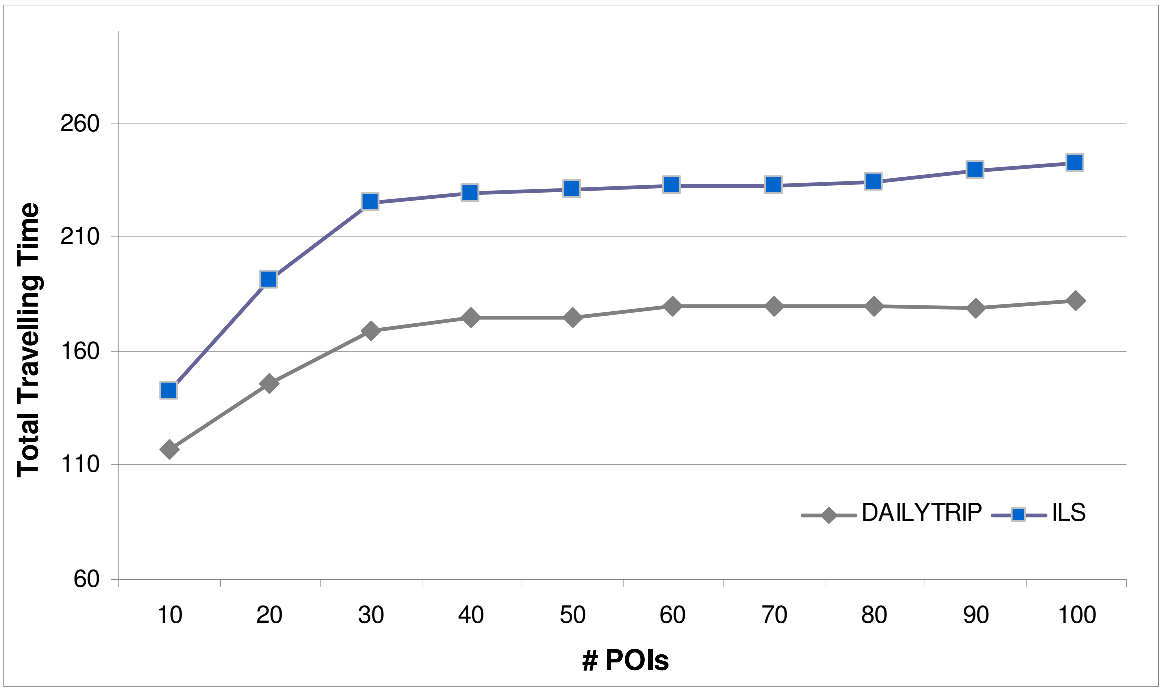


Figure 6-15. Comparison of the total travel time derived from DailyTRIP and ILS algorithms with 3 Itineraries.

Figure 6-15 shows the total travel time for all 3 days (i.e. 3 itineraries) as a function of the increasing number of POIs. It is obvious that as the number of POIs increase (i.e. the total itinerary is prolonged) so to does the total travel time. The DailyTRIP yields improved results which are mainly due to the TSP algorithm executed on its last phase, which ensures near-optimality of each daily itinerary in terms of length. However, the total itinerary length of DailyTRIP very much depends on the value of distance coefficient, that is decreasing the traveling time coefficient resulting in increased total itinerary length (it signifies that the time spent by the user for traveling is less important) and vice versa. That represents an interesting trade-off among profit and travelling time. It also noted that after crossing a threshold number of POIs (around 30 POIs) small fluctuations are observed on both algorithms in the total travel time (as some POIs are substituted by others).

Figure 6-16 compares DailyTRIP against ILS in terms of their respective overall itinerary time. That sums up the overall time spent for travelling (i.e. moving from one POI to another) and visiting POIs. As expected, this figure resembles Figure 6-15 as the total visiting time is added to the total travelling time values.

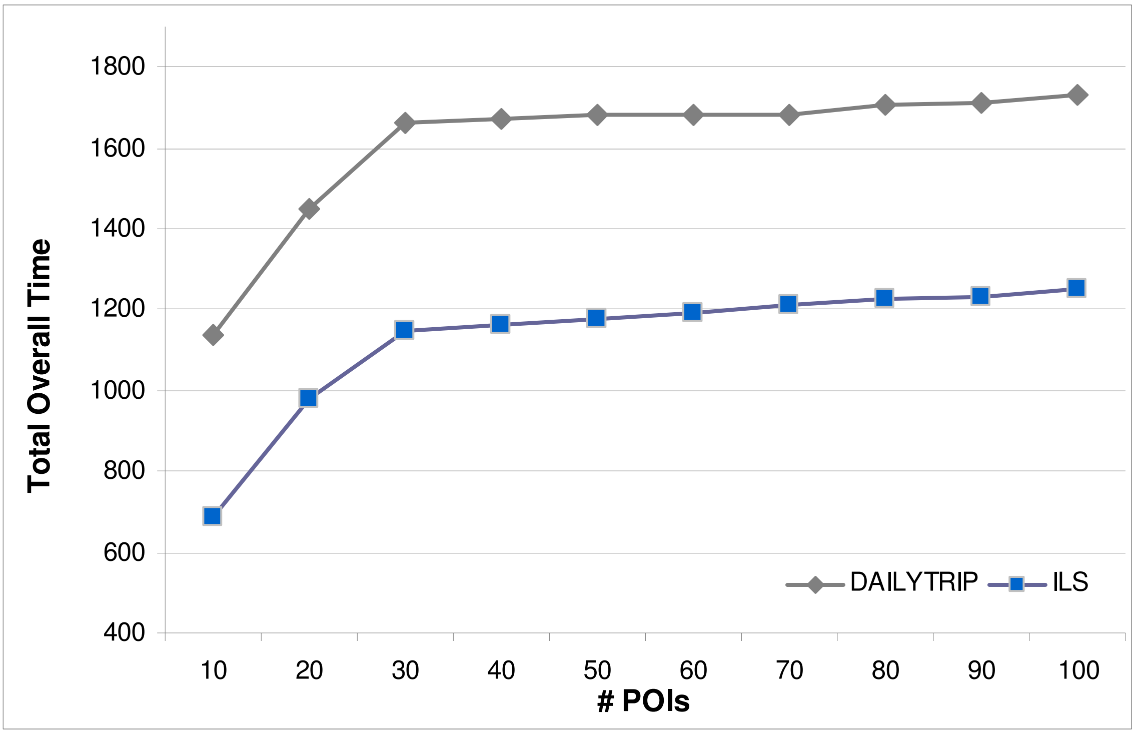


Figure 6-16. Comparison of the overall time derived from DailyTRIP and ILS algorithms with 3 itineraries.

Below, Figure 6-17 compares the total itineraries length of DailyTRIP and ILS (that is the sum of individual daily itinerary lengths). Evidently, the illustrated distance values are proportional to the travelling times of Figure 6-15.

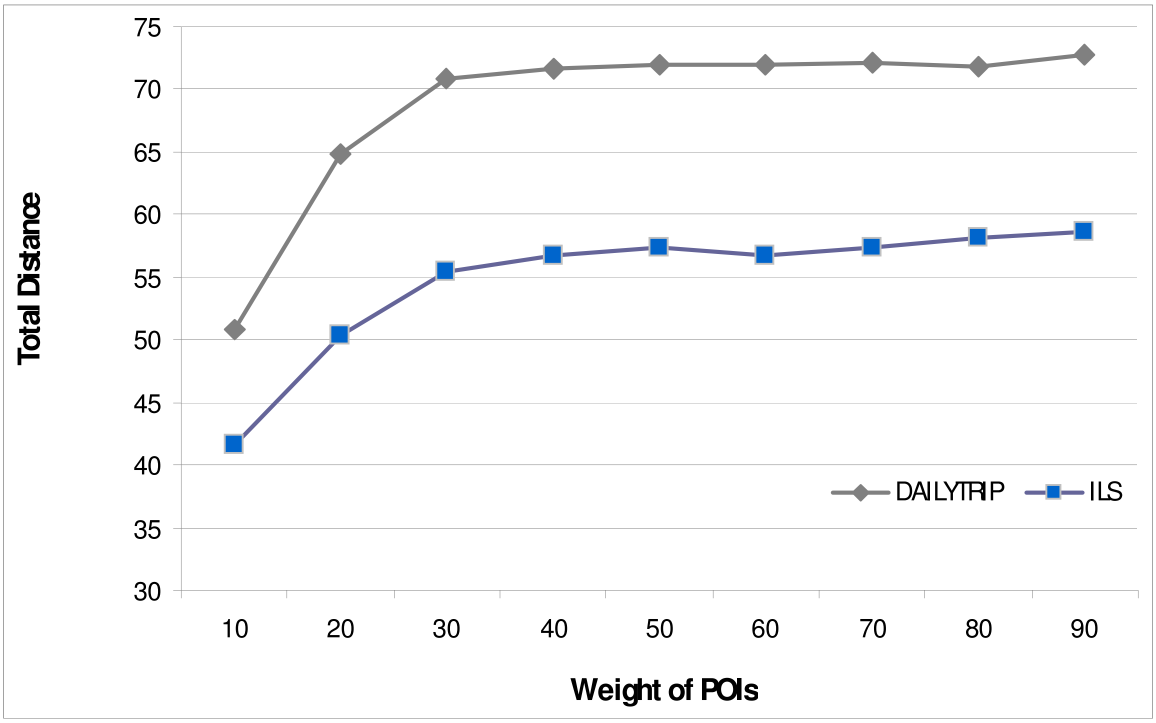


Figure 6-17. Simulation output of the total distance compared to the increasing number of POIs per run for 3 itineraries.

Figure 6-18 compares the DailyTRIP against ILS with regards to the total travelling time, that is, the sum of individual daily itineraries travelling time. In this test, we consider 4 days of stay (i.e. 4 daily itineraries). DailyTRIP performs better than ILS as it yields reduced total travelling time. This is because ILS only considers POIs' profit as the only criteria for appending a new POI in an itinerary, whereas DailyTRIP considers both profit and topological distance (i.e. travelling time). Both algorithms are shown to increase their travelling time up to a threshold of ~40 POIs, which is the maximum number of POIs that can be accommodated within the 4 itineraries. Above this threshold, both DailyTRIP and ILS travelling time exhibit some fluctuations as the itineraries are modified (some POIs are substituted by others, hence, the itineraries' length is modified).

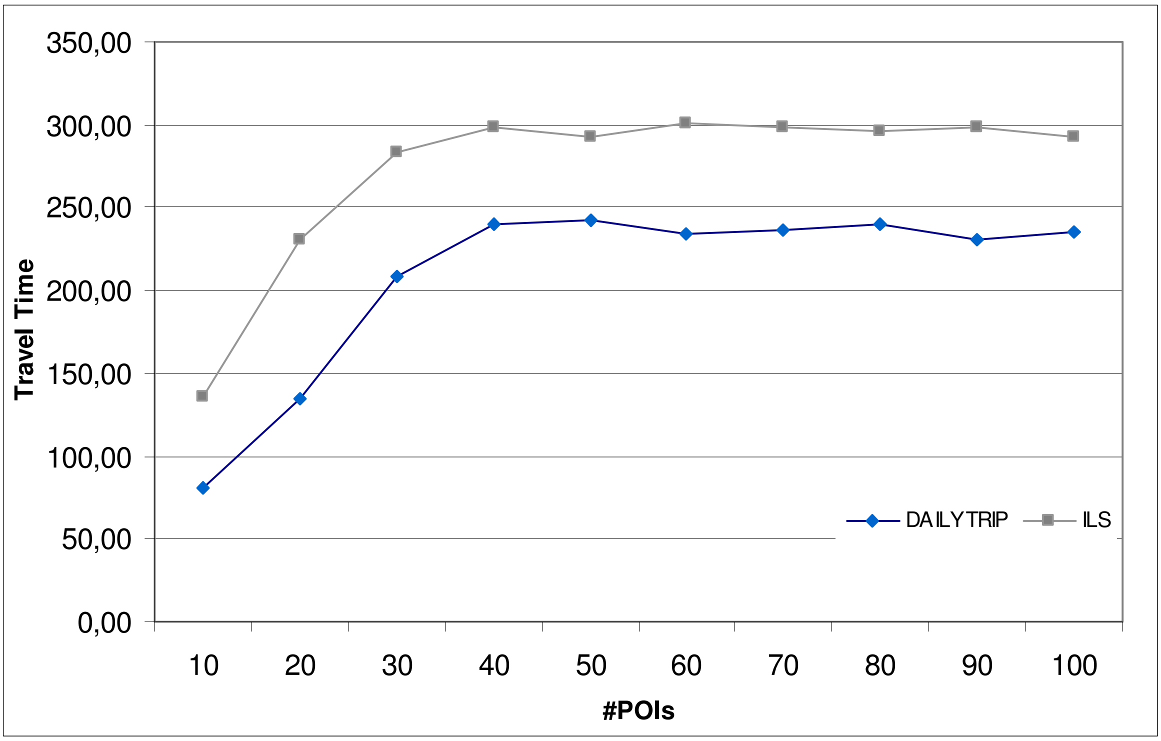


Figure 6-18. Total travel time using 4 days itinerary simulation

Last, Figure 6-19 portrays the variance of the overall profit per day (i.e. the sum of profits of POIs visited on each day). Herein we assume 4 daily itineraries (i.e. staying 4 days at the destination). Small variance values denote fairly balanced itinerary profit values over the four itineraries while larger variance values denote that some itineraries are much more 'profitable' than others. Notably, DailyTRIP yields smaller variance values in comparison to ILS. This is because DailyTRIP considers the four itineraries interchangeably while inserting new nodes, hence, derived itineraries comprise approximately the same number of POIs (± 1) , ensuring a balanced overall itinerary profits. On the other hand, ILS first completes an itinerary before proceeding to the next one, which implies that in scenarios with few POIs and relatively large number of itineraries, ILS derives some 'full' and some 'empty' itineraries (hence, large variance values). In other words, DailyTRIP distributes the 'burden' of visiting POIs over all available days of stay, while ILS derives 'overloaded' days in the beginning and relatively 'relaxed' days in the end of the staying time. Notably, as the number of POIs increase, ILS obtains improved distributions of POIs into individual itineraries (i.e. its itineraries are gradually 'filled' with POIs) which in turn decreases variance values.

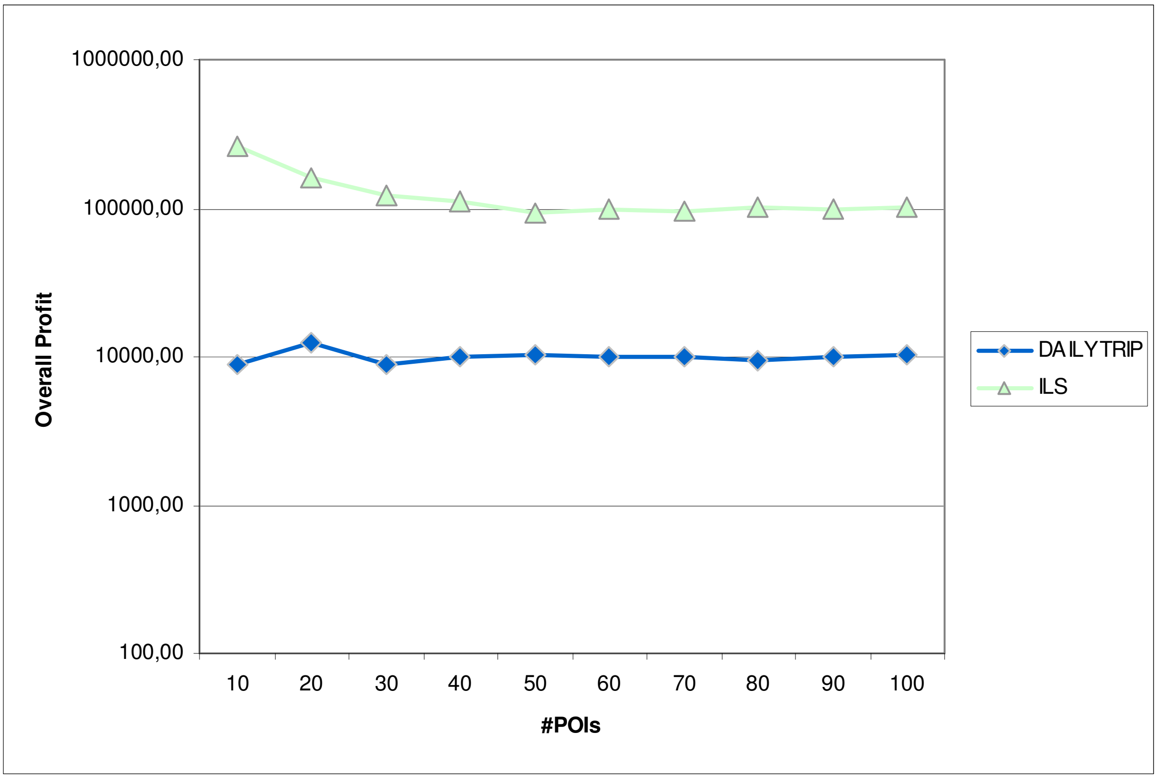


Figure 6-19. Variance of the individual itineraries' overall profit per day for both algorithms

The satisfactory performance of DailyTRIP suggests it is suitable for online usage. In particular the algorithm requires less than 2,5 sec for topologies of 15 nodes, which represents a reasonable number of POIs to visit at any destination to derive a solution (excluding the time required to draw the solution on Google Maps) that deviates less than 7% from the optimal solution, considering problem spaces spanning up to 25 nodes.

6.9. Summary

This chapter introduced DailyTRIP, a heuristic approach for deriving personalized recommendations of daily tourist itineraries for tourists visiting any tourist destination. DailyTRIP considers selected POIs that a traveller would potentially like to visit and derives a near-optimal itinerary for the traveller for each day of visit. Our approach takes into account user preferences, time available for visiting sights in daily basis, opening days of sites and average visiting times for these sites. The objective of DailyTRIP is to maximize the overall profit associated with visited POIs (whereby individual profits are calculated as a function of the POIs' 'objective' importance and the user's potential interest for the POI) while not violating the daily time budget for sightseeing. Our algorithm has been implemented and proved suitable for online applications (real-time design of itineraries).

Part of research presented in this chapter was published in the Proceedings of the IEEE ISCC'2010 (see Appendix A).

CHAPTER 7

SUMMARY, RESEARCH CONTRIBUTIONS & DIRECTIONS FOR FUTURE WORK

This chapter brings together the work described in the preceding chapters of this thesis. Section 7.1 summarizes the main contributions and findings of this thesis. Section 7.2 concludes and identifies areas in which this work could be developed further.

7.1. Summary of Main Research Contributions

A first objective of this thesis has been to evaluate the assets and limitations related to the practical use of 'mobile tourism' applications. Based on this evaluation, we focused on the investigation, design and implementation of a multiplatform tourist framework which allowed tourists to use the web in order to dynamically 'build' customized mobile standalone guides. Furthermore, we investigated the use of personalized tourist services in this framework; this involved the design and implementation of the MTRS (Mobile Tourism Recommender System) which employed 'context-aware collaborative filtering' techniques. Last, we implemented the DailyTRIP algorithm for deriving personalized recommendations of daily tourist itineraries for tourists visiting any tourist destination.

The main research contributions and findings of this thesis are summarised in the following:

- Survey of the technological landscape of mobile computing and review of development environments currently available for mobile applications. The survey led to the identification of a coherent set of requirements that authoring tools and application development platforms should satisfy in order to allow the efficient and cost-effective development of cultural tourist applications on mobile devices.
- Thorough evaluation of mobile applications used by tourists to get information, navigation and guidance. The evaluated projects were divided into four main groups: mobile guide applications, web-to-mobile guides, mobile phone navigational assistants and mobile web-based applications. The evaluation extracted design guidelines in respect to information models, position and map technologies, architecture/network infrastructure, input/output modalities and unique services.
- Design and implementation of a novel mobile tourism application framework. A web application was implemented to allow tourists to create personalized guides and later install them on their mobile phone. The users can "tag" content of interest via the web application and then by selecting their mobile device model, download the dynamically generated mobile application.
- Design and implementation of MTRS (Mobile Travel Recommender System), used to assist tourists in choosing places to visit. This system utilizes collaborative filtering techniques while also taking into account contextual information (e.g. the time, season, weather conditions and location of the user at the time she 'publishes' information related to a POI) for deriving improved recommendations in a pervasive environment.

- Implementation and evaluation of DailyTRIP, a heuristic approach for deriving personalized recommendations of daily tourist itineraries for tourists visiting any tourist destination. DailyTRIP considers POIs that a traveller would potentially like to visit and yields a near-optimal itinerary for the traveller for each day of visit. Our approach takes into account user preferences as well as several constraints liketime available for visiting sights on a daily basis, opening days of sites, anticipated visiting times for these sites, etc.

7.2. Directions for Future Work

Even at the time of writing this thesis after 4 years of research, widely known companies like Google and Facebook still offer a choice in how users access their fully fledged rich-user interfaced static web applications via mobile devices. They either offer mobile users the ability to directly access cut down versions of the web application via a microbrowser or they offer users the ability to download and install a ported thick mobile application which can offer the same rich user interface and enhanced facilities which their web based counterparts offer. Even though these thick clients work without the need of a constant connection they will always have problems of porting mobile applications to vast number of fragmented mobile devices. The problem of porting must be dealt with by mobile application developers which, in turn, results in many man hours and incurs large costs.

In this context, our web-to-mobile model currently represents a convincing answer to such problems. Yet, given the changing trend, on decreasing prices for the mobile internet and the increasing trend towards the mobile coverage of suburban and rural areas, the use of constant connection selected by mobile device users will increase gradually. This along with the evolution of mobile micro browsers (which seems, will eventually support the rich functionality of web browsers) shape the ground for a near-future turn towards the mobile web. However, in the scope of the never ending presentations of new mobile platforms and business paradigms into the market e.g. Google Android 2.2, Apple's iPhone 4, Samsung Bada, etc, proprietary development platforms and proprietary based market places might exceed the evolution of microbrowsers and be the future of mobile tourism guides. This gives us the incentive to continue expanding the web-to-mobile model to be able to port applications to new mobile platforms directly from the web interface.

Indeed, getting back to the present, a number of issues were raised from participants of our usability studies whilst evaluating the multiplatform tourist guide framework. This alone suggested future enhancements of the current mobile application. Some of these enhancements are summarised below:

- Enhancement of digitized maps so as to provide three zoom levels (instead of two), target highlighting, street names and clickable objects, like city attractions, to provide quick access to important tourist information;
- Inclusion of emergency contacts in the city map (pharmacies, hospitals, port & tourist police authorities, etc) and search facilities to locate the shortest contacts, depending on the current location of the user;
- Provision of daily/weakly weather forecast reports based on the user's location.

Apart from these enhancements, we plan to direct our research of the multiplatform mobile tourist guide towards to the following directions:

- Expand our system so as to include recommendations for information other than local attractions, such as lodging, restaurants, entertainment, travel agencies, souvenir shops, local authorities, etc.
- Study the feasibility of a mobile peer-to-peer approach for the direct exchange of multimedia content, reviews, recommendations, etc, among users while onsite.

- Consult domain specialists (e.g. experts in the field of tourism) to manually create appropriate user stereotypes; along the same line, we will investigate machine learning techniques for automating the generation of stereotypes as proposed in [136].

- Focus on algorithmic solutions that will provide daily, personalized tourist itineraries that will depend on a variety of criteria, such as the overall journey duration and day-to-day duration using current weather conditions, POIs opening hours, physical distances among POIs, etc.

- Investigate new multimedia mobile platforms to host the mobile guide application, including: Android (Linux), BlackBerry, iPhone (Mac OS X), Symbian, and Windows Mobile, Nokia OVI etc

- Focus on variations of DailyTRIP algorithm that will incorporate additional TIDP problem parameters and constraints, e.g. weather conditions while on travel, financial budget (for transport and POIs admission charges), etc.

- Investigate the use of a combination of means of transport, e.g. walking and bus service, taking into account various aspects of alternative transport services (e.g. walking time to the nearest metro station, day and time-dependent metro service frequencies, etc).

- Consider methods for fast itinerary updates, wherein derived itineraries are subject to modifications due to sudden weather changes, taking longer than anticipated to visit or arrive at POIs, etc.

- Incorporate location-awareness in the DailyTRIP, deriving itineraries that start at the user’s current position.

REFERENCES

[1] Abowd A, Atkeson C, Hong J, Long S, Pinkerton M. Cyberguide: A Mobile Context-Aware Tour Guide. Wireless Networks. 1996, 3(5):421–433.

[2] Abbaspour R.A., and Samadzadegan F. Itinerary Planning in multi-modal Urban Transportation Network. Journal of Applied Sciences, 9(10): 1898-1906, 2009.

[3] Adobe: Mobile & Devices, last visited 30/10/2006

[4] Adobe Flash lite product page, HTTP://WWW.ADOBE.COM/ PRODUCTS/FLASHLITE/ ?PROMOID=DIOCQ, last visited August 2010

[5] Alencar P, Cowan D. Handbook of Research on Mobile Software Engineering: Design, Implementation and Applications. ISBN 9781615206551, IGI Global, 2010

[6] Akyildiz I.F., Su W., Sankarasubramaniam Y., Cayirci E. (2002). Wireless Sensor Networks: A Survey. Computer Networks, 38(4): 393-422.

[7] Amirian S, Hand-held Mobile Computing in Museums. A Background Paper, 2001.

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

[9] Android - An Open Handset Alliance Project. http://code.google.com/android/, Last visited December 2009.

[10] Android Platform, http://www.android.com/index.html, Last visited July 2010.

[11] Apple – iTunes web page, http://www.apple.com/itunes/, Last visited July 2010.

[12] ASP.NET Mobile Controls, 2006, http://www.asp.net/mobile/. Last visited August 2010.

[13] Bada platform- Samsung, http://www.bada.com/, Last visited July 2010.

[14] bango.com. Mobile web use in the US surges ahead with three fold increase in last 12 months. http://bango.com/news/pressreleases/143_mobileWebgrowth.aspx. Last visited October 2008.

[15] Baus J, Kray C & Krueger A, Visualization of Route Descriptions in as Resource-Adaptive Navigation Aid. Cognitive Processing, 2 (2-3). pp. 323-245, 2001.

[16] Baus J, Kruger A, Wahlster W. A resource-adaptive mobile navigation system. International Conference on Intelligent User Interfaces archive, Proceedings of the 7th international conference on Intelligent user interfaces. 2002, pp. 15-22.

[17] Bederson B, Clamage A, Czerwinski M & Robertson G, A Fisheye Calendar Interface for PDAs: Providing Overviews for Small Displays. In Proceedings of the International Conference on Human Factors in Computing Systems (CHI'03), pp. 618-619, 2003.

[18] Belsis P., Konstantopoulos C., Mamalis B., Pantziou G., Skourlas C. (2008). Interactive Cluster-based Personalized Retrieval on Large Document Collections, Proceedings of the 1st International Symposium on Intelligent Interactive Multimedia Systems and Services (KES-IIMSS'08), Vol. 142: 211-220.

[19] Benelli G., Bianchi A., Marti P., Not E., Sennati S.. (1999). HIPS: Hyper-Interaction within Physical Space. Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMC'99).

[20] Berger S., Lehmann H., Lehner F. (2003). Location-Based Services in the Tourist Industry. Information Technology & Tourism 5(4): 243-256.

[21] Berka T., Plößnig M. (2004). Designing Recommender Systems for Tourism. Proceedings of the 11th International Conference on Information Technology in Travel & Tourism (ENTER'2004).

[22] Bertini, E., Gabrielli, S., Kimani, S. (2006). Appropriating and Assessing Heuristics for Mobile Computing. Proceedings of the Working Conference on Advanced Visual Interfaces (AVI'06): 119-126.

[23] BlueBlitz Proximity & Mobile Marketing. Building a Platform for Mobile Services, http://www.blueblitz.com/mbMagicBeamer.htm, last visited 12/01/2008.

[24] Black P.E. (2005). Greedy algorithm. Dictionary of Algorithms and Data Structures, U.S.National Institute of Standards and Technology, http://www.itl.nist.gov/div897/sqg/dads/HTML/greedyalgo.html. Last accessed: Junuary 10, 2010.

[25] Brown B & Chalmers M, Tourism and Mobile Technology. In Proceedings of the 8th European Conference on Computer Supported Cooperative Work (CSCW'03), pp. 335-354, 2003.

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

[27] Buhalis D.. (2003). eTourism: Information Technology for Strategic Tourism Management, Prentice Hall.

[28] Burke, D., (2007) A Short Position Paper for W3C/OpenAjax Alliance Workshop on Mobile AJAX.

[29] Buyukkokten O, Garcia-Molina H, Paepcke A & Winograd T, Efficient Web Browsing on Handheld Devices Using Page and Form Summarization. ACM Transaction on Information Systems, 20(1), pp. 82-115, 2002.

[30] Canadian Heritage Information Network (CHIN): Tip Sheets, Personal Digital Assistants (PDA), PDA Aesthetics and Interface Design, Creating and Managing Digital Content, 2004. http://www.chin.gc.ca/English/Digital_Content/index.html, Last visited May 2009

[31] Celtnet Mobile Phones- Introduction to the mobile phone, http://www.celtnet.org.uk/mobile-phone/mobile-phone-information.php last visited September 2010

[32] Checkland P & Scholes J, Soft Systems Methodology in Action. Chichester-NY: John Wiley & Sons, 1990.

[33] Chen G, Kotz D. A Survey of Context-Aware Mobile Computing Research. Dartmouth Computer Science Technical Report TR2000-381. 2001.

[34] Cheverst K, Mitchell K, Davies N. The role of adaptive hypermedia in a context-aware tourist GUIDE. Commun. ACM 45(5). 2002, pp. 47-51.

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

[36] Chao, I.-M., Golden B. L., Wasil E. A. (1996). The Team Orienteering Problem. European Journal of Operational Research, 88(3): 475-489.

[37] Chittaro L & Cin P.D , Evaluating Interface Design Choices on WAP Phones: Navigation and Selection, Personal and Ubiquitous Computing, 6, pp. 237-244, 2002.

[38] Chu D.C & Humphrey M, Mobile OGSI.NET: Grid Computing on Mobile Devices. In Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, pp. 182-191, 2004.

[39] Ciavarella C & 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.

[40] Conner M., Herlocker J. (1999). Clustering Items for Collaborative Filtering. Proceedings of the ACM SIGIR Workshop on Recommender Systems.

[41] Damith R., Fragmentation of mobile applications, http://www.comp.nus.edu.sg/~damithch/df/device-fragmentation.htm, Last Visited August 2010

[42] Davies N, Cheverst K, Mitchell K, Friday A. 'Caches in the Air': Disseminating Tourist Information in the Guide System. WMCSA. 1999, pp. 11-19.

[43] Dey A.K, Salber D & Abowd G.D, A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Human-Computer Interaction, 16, pp. 2-4, 2001.

[44] E. W. Dijkstra Archive, http://userweb.cs.utexas.edu/users/EWD/, Last visited August 2010

[45] Dublin Tourism (2009) Dublin Tourism on the Wireless Web. http://www.visitdublin.com/wireless/. Accessed: October 2009.

[46] Dumas Y., Desrosiers J., Gelinas E., Solomon M. M. (1995). An Optimal Algorithm for the Traveling Salesman Problem with Time Windows. Operational Research, 43(2): 367-371.

[47] Dunlop M & Brewster S, The challenge of mobile devices for human computer interaction. Personal and Ubiquitous Computing, 6 (4). pp. 235 -236, 2002.

[48] Dunlop M., Ptasinski P., Morrison A., McCallum S., Risbey C., Stewart F. (2004). Design and Development of Taeneb City Guide - From Paper Maps and Guidebooks to Electronic Guides. Proceedings of the International Conference Information and Communication Technologies in Tourism (ENTER’2004): 58-64.

[49] Dublin.mobi. Welcome To Dublin.Mobi, http://www.dublin.mobi/mobi/, last visited March 2009

[50] Economou, D., Varitimiadis, S., Samaroudi, M., Gavalas, D. (2006). The Effectiveness of Personal Digital Assistants (PDAs), Audio-Guides or Paper Leaflets to Enhance the Museum Visit Experience. Proceedings of the 3rd International Conference on Museology (ICM’2006).

[51] eIndia.mobi. eIndia.mobi India’s Mobile Portal, http://eindia.mobi/, last visited March 2009

[52] Europa - Information Society, (2009) Roaming: The Roaming Regulation. http://ec.europa.eu/information_society/activities/roaming/regulation/index_en.htm. Accessed: October 2009.

[53] Europe, Nokia. Nokia Europe. Nokia Europe - Maps - Main. Nokia, HTTP://EUROPE.NOKIA.COM/MAPS. last visited October 2008.

[54] Evans J & 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.

[55] Felfernig A., Gordea S., Jannach D., Teppan E., Zanker M. (2007). A Short Survey of Recommendation Technologies in Travel and Tourism. ÖGAI Journal, 25(7): 17-22.

[56] Garcia A., Linaza M.T., Arbelaitz O. and Vansteenwegen P. Intelligent Routing System for a Personalised Electronic Tourist Guide. Proceedings of the International Conference in Information and Communication Technologies in Tourism 2009 (ENTER'2009):185-197, 2009.

[57] Gavalas D., Economou, D. (2007). The Technology Landscape of Wireless Web. International Journal of Mobile Communications, 5(5): 508-527.

[58] Gavalas D., Economou D. (2010). Development Platforms for Mobile Applications: Status and Trends. IEEE Software, in press.

[59] Garcνa-Crespo A., Chamizo J., Rivera I., Mencke M., Colomo-Palacios R., Gσmez-Berbνs J.M. (2009). SPETA: Social Pervasive e-Tourism Advisor. Telematics and Informatics, 26(3): 306-315.

[60] Gefflaut A., van Megen, Siegemund F., Sugar R., (2005). Porting the .NET Compact Framework to Symbian Phones-A Feasibility Assessment, Proceedings of the 2005 International Conference on .NET Technologies.

[61] Google. Google. Google mobile maps. HTTP://WWW.GOOGLE.COM/MOBILE/DEFAULT/MAPS.HTML.

[62] Google, Google Maps Navigation on your Mobile phone, http://www.google.com/mobile/navigation/#p=default, last visited January 2010.

[63] Goldman, S.V., Pea, R., Maldonado, H., Martin, L., White, T. (2004) Functioning in the Wireless Classroom. Proceedings of the 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'04): 75-82.

[64] Google Maps. Google Maps API Concepts, http://code.google.com/apis/maps/documentation/. Last accessed: January 10, 2010.

[65] Google Static Maps API, http://code.google.com/apis/maps/documentation/staticmaps/. Last accessed: January 10, 2010.

[66] Grun, C., Werthner, H., Proll, B., Retschitzegger, W., Schwinger, W. (2008). Assisting Tourists on the Move-An Evaluation of Mobile Tourist Guides. Proceedings of the 7th International Conference on Mobile Business (ICMB'08): 171-180.

[67] Gupta A & Srivastava M, Integrated Java Technology for End-to-End m-Commerce: http://developers.sun.com/techtopics/mobility/midp/articles/mcommerce/. Last visited in 12/01/2008

[68] Hagen K., Kramer R., Hermkes M., Schumann B. Mueller P (2005). Semantic Matching and Heuristic Search for a Dynamic Tour Guide. In: Information and Communication Technologies in Tourism..

[69] Helsgaun K, (2000). An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic, European Journal of Operational Research, Volume 126, Issue 1, Pages 106-130

[70] Herlocker J., Konstan J., Terveen L., Riedl J., Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems, 22(1), pp. 5-53, 2004.

[71] Huerta, Juan M. / Stern, Richard M. (1998): Speech recognition from GSM codec parameters, In ICSLP-1998, paper 0626

[72] Hinze, A., Buchanan, G. (2005). Context-Awareness in Mobile Tourist Information Systems: Challenges for User Interaction. Proceedings of Workshop on Context in Mobile HCI at the Conference for 7th International Conference on Human Computer Interaction with Mobile Devices and Services.

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

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

[75] Holzner, S. (2007). Ajax Bible. Wiley.

[76] Holtsblatt K, Designing for mobile devices: Experiences, Challenges and Methods. In Communications of the ACM, 48(7). pp. 33-35, 2005.

[77] Harakawa, T., Kamada, T., (2007). Embedded Ajax - Web 2.0 Optimized for Mobile Devices. W3C/OpenAjax Workshop on Mobile Ajax (white paper).

[78] ICT/telecommunication data, http://www.itu.int/ITU-D/ict/statistics/, Last visited July 2010

[79] Jain, A.K, Dubes, R.C, (1988). Algorithms for Clustering Data. Prentice Hall.

[80] Jeff, Axup. Methods Of Understanding And Designing For Mobile Communications PhD Thesis. s.l. : The University Of Queensland, 2006-2007.

[81] JAVA ABOUT – Learn about Java technology, http://www.java.com/en/about/, Last visited August 2010

[82] Java 2 Platform Micro Edition (J2ME). Last visited in 12/01/2008: http://java.sun.com/j2me/.

[83] Java API for XML Processing (JAXP). Last visited in 12/01/2008: http://java.sun.com/webservices/jaxp/index.jsp.

[84] JavaFX Mobile. Retrieved December 13, 2009, from: HTTP://JAVAFX.COM/.

[85] JavaServer Pages Technology. Last visited in 12/01/2008: http://java.sun.com/products/jsp/.

[86] J2ME Polish (2009). J2ME Polish home page. http://www.j2mepolish.org/. Last visited October 2009.

[87] J2ME Polish GPL License. J2ME Polish home page. http://www.j2mepolish.org/cms/leftsection/licensing.html#gpl. Last Visited August 2010.

[88] Janecek A & Hlavacs H, Programming Interactive Real-Time Games over WLAN for Pocket PCs with J2ME and .NET CF. In Proceedings of the 4th ACM SIGCOMM workshop on Network and system support for games, pp. 1-8, 2005.

[89] Jones M & Marsden G, Mobile Interaction Design, Wiley, 2005, ISBN: 978-0-470-09089-3.

[90] JSR 179: Location API for J2ME. Last visited in 12/01/2008: http://jcp.org/en/jsr/detail?id=179.

[91] JSR 184: Mobile 3D Graphics API for J2ME. Last visited in 12/01/2008: http://www.jcp.org/en/jsr/detail?id=184

[92] Kabassi K. (2010). Personalizing Recommendations for Tourists. Telematics and Informatics, 27(1): 51-66.

[93] Konstantopoulos C., Gavalas D., Pantziou G. (2008). Clustering in Mobile Ad-Hoc Networks through Neighborhood Stability-Based Mobility Prediction. Computer Networks, 52(9): 1797-1824.

[94] Kjeldskov, J., Stage, J. (2003). New Techniques for Usability Evaluation of Mobile Systems, International Journal of Human-Computer Studies, 60:599-620.

[95] Kramer R., Modsching M., ten Hagen K. (2006). A City Guide Agent Creating and Adapting Individual Sightseeing Tours Based on Field Trial Results. International Journal of Computational Intelligence Research 2(2): 191-206.

[96] Kray, C., Baus, J. (2003). A Survey of Mobile Guides. Proceedings of the 5th International Symposium on Human Computer Interaction with Mobile Devices and Services (Mobile HCI'2003): 1-5.

[97] Kray C., Baus J., Cheverst K. "A survey of map-based mobile Guides" in: Map-based mobile services – Theories, Methods and Implementations. 2005, pp. 197-216.

[98] Kray, C., Laakso, K., Elting, C., Coors, V. (2003). Presenting Route Instructions on Mobile Devices. Proceedings of the 2003 International Conference on Intelligent User Interfaces (IUI'03): 117-124.

[99] Krüger A., Malaka R. (2004). Artificial Intelligence Goes Mobile. Applied Artificial Intelligence, 18: 469-476.

[100] kXML parser. Last visited in 10/4/2006: http://kxmlrpc.objectweb.org/.

[101] Lawton G, Moving Java into Mobile Phones (2002). IEEE Computer 35(6): 17-20

[102] Larsena J., Urryb J., Axhausenc K.W. Networks and Tourism: Mobile Social Life. Annals of Tourism Research, 34(1), pp. 244-262, 2007.

[103] Lehn D & 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 12/01/2008: http://www.ichim.org/ichim03/PDF/088C.pdf.

[104] Li M., Yang B.. A Survey on Topology Issues in Wireless Sensor Network. Proceedings of the International Conference on Wireless Networks (ICWN'06), 2006.

[105] Long S, Kooper R, Abowd G D, Atkeson C G. Rapid prototyping of mobile context-aware applications: the Cyberguide case study. MobiCom '96: Proceedings of the 2nd annual international conference on Mobile computing and networking. 1996.

[106] Luyten K, Coninx K, Houben G, Winters F. Blended Maps and Layered Semantics for a Tourist Mobile Guide. The 11th International Conference on Human-Computer Interaction, Las Vegas, Nevada, USA. July 2005, pp. 22-27.

[107] Malaka R., Zipf A. DEEP MAP - Challenging IT Research in the Framework of a Tourist Information System. Information and Communication Technologies in Tourism, Proceedings of ENTER 2000. 7th, 2000.

[108] Maruyama A., Shibata N., Murata Y., Yasumoto K., and Ito M. P-tour: A Personal Navigation System for Tourism. Proceedings of 11th World Congress on ITS:18-21, 2004.

[109] Maw S.Y., Naing M.M. (2006). Personalization Agent for Multi-Agent Tourism System. Proceedings of the 4th International Conference on Computer Application: 117-124.

[110] Meng L, Zipf A, Reichenbacher T. Map-based Mobile Services: Theories, Methods and Implementations. Springer-Verlag. 2005.

[111] meteo.gr Greek weather forecast service, http://www.meteo.gr/.

[112] Mobile Tourism and Mobile Government - An inventory of European Projects, Research Report (2006), http://www.mgovworld.org/whitepapersandcasestudies, Last visited October 2009

[113] Micha K & 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.

[114] Microsoft Map Point Web service, http://msdn.microsoft.com/mappoint/.

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

[116] Microsystems, Sun. The Java ME Platform - the Most Ubiquitous Application Platform for Mobile Devices. Sun Microsystems. http://java.sun.com/javame/index.jsp.

[117] Mobasher B., Cooley R., Srivastava J. (2000). Automatic Personalization Based on Web Usage Mining. Communications of the ACM, 43(8): 142-151.

[118] Mobasher B., Dai H., Luo T., Nakagawa M. (2002). Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization. Data Mining and Knowledge Discovery, 6(1): 61-82.

[119] mobiletravelaide.com. Mobile Travel Aide - Mobile-Friendly Travel Information Center http://www.mobiletravelaide.com, last visited March 2009

[120] MyCityMate, MyCityMate homepage, http://www.mycitymate.com/, last visited January 2010.

[121] Mobile Google map (2009). Mobile Google Map site. http://www.google.com/gmm/index.html. Accessed October 2009.

[122] MyCityMate (2009). MyCityMate homepage. http://www.mycitymate.com/. Accessed: October 2009.

[123] MySQL, http://www.mysql.com/. Accessed 10 Jun 2009.

[124] Netbearns.org, Netbeans Intergrated Development Environment (2010), http://netbeans.org/, Last visited September 2010

[125] Nielson J, Jakob Nielsen's Alertbox, July 20, 2009: Mobile Usability, http://www.useit.com/alertbox/mobile-usability.html, Last visited August 2010

[126] Nielsen J, Heuristic Evaluation, in Nielsen, J. and Mack R.L. (eds.). Usability Inspection Methods. John Wiley & Sons, New York, 1994.

[127] Nielsen, J., Molich, R. (1990). Heuristic Evaluation of User Interfaces. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'90): 249-256.

[128] Nielsen, J. (1992). Finding Usability Problems through Heuristic Evaluation. Proceedings of the SIGCHI Conference on Human factors in Computing Systems (CHI'92): 373-380.

[129] Nokia Beta Labs, Ovi Maps Beta for mobile, http://betalabs.nokia.com/betas/view/nokia-maps-30-maps-ovi, last visited January 2010.

[130] Object Management Group. http://www.omg.org, last visited June 2009

[131] Oliver, E., (2008). A Survey of Platforms for Mobile Networks Research. ACM SIGMOBILE Mobile Computing and Communications Review, 12(4): 56-63.

[132] Ophrys Systèmes (2009). OPHRYS Systems: Home, discover our audioguide. http://www.ophrys.net/audioguide_english/accueil.html. Accessed: October 2009.

[133] Oppermann R, Specht M. A Nomadic Information System for Adaptive Exhibition Guidance. Proceedings of the International Conference on Hypermedia and Interactivity in Museums (ichim 99). September 1999, pp. 103-109.

[134] Oppermann R & 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.

[135] Ovi by Nokia – Mobile maps, music downloads, mobile games, photo sharing, file sharing, http://www.ovi.com/services/worldwide, last visited July 2010

[136] Paliouras, G., Karkaletsis, V., Papatheodorou, V. (1999). Exploiting Learning Techniques for the Acquisition of User Stereotypes and Communities. Proceedings of the 7th International Conference on User Modeling (UM'99): 169-178.

[137] Preece J, Rogers Y & Sharp H, Interaction design: Beyond human – computer interaction. Danvers, MA: Wiley, 2002.

[138] Proctor N & Tellis C, The state of the art in museum handhelds in 2003. In Proceedings of the International Conference of Museums and the Web 2003. D. Bearman and J.

Trant, (eds.). Archives & Museum Informatics, 2003. Last visited in 12/01/2008: HTTP://WWW.ARCHIMUSE.COM/MW2003/PAPERS/PROCTOR/PROCTOR.HTML.

[139] Project Capuchin. Retrieved December 13, 2009, from: HTTP://DEVELOPER.SONYERICSSON.COM/SITE/GLOBAL/DOCSTOOLS/PROJECTCAPUCHIN/P_PROJECTCAPUCHIN.JSP.

[140] Poonsuph, R., (2007). Enterprise User Interface and Technology Comparison. Proceedings of the 9th International Decision Science Institute Conference.

[141] Poslad S, Laamanen H, Malaka R, Nick A, Buckle P, Zipl A. CRUMPET: creation of user-friendly mobile services personalised for tourism. 3G Mobile Communication Technologies, 2001. Second International Conference on (Conf. Publ. No. 477). 2001, pp. 28-32.

[142] Pospischil G, Umlauft M, Michlmayr E. Designing LoL@, a Mobile Tourist Guide for UMTS. Proceedings 4th International Symposium on Human Computer Interaction with Mobile Devices (Mobile HCI '02), Pisa, Italy. September 2002.

[143] Przybilski M, Campadello S, Saridakis T. Mobile on Demand Access of Service-Annotated 3D Maps. Proceedings of the 22nd IASTED International Conference on SOFTWARE ENGINEERING (SE 2004). 2004, pp. 719-725.

[144] Sparacino Z, Larson K, MacNeil R, Davenport G & 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 12/01/2008: http://ic.media.mit.edu/Publications/Conferences/MuseumTracking/PDF/MuseumTracking.pdf.

[145] Studio 7.5 (2005), Designing for Small Screens, Ava Academia, ISBN: 2-940373-07-8

[146] Tenmoku R, Kanbara M & 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.

[147] Ricci F. (2002). Travel Recommender Systems. IEEE Intelligent Systems, 17: 55-57.

[148] Rich, E. (1998). User Modeling via Stereotypes. In: Readings in Intelligent User Interfaces, Maybury, M.T., Wahlster, W. (Eds.), Morgan Kaufmann Publishers Inc.: 329-342.

[149] s60blog.com. Nokia Maps 3.0 - Maps With Ovi - Mail on Ovi - Consumer Messaging. The S60 Blog. http://s60blog.com/2008/12/nokia-ovi-just-got-bigger-and-better. last visited October 2008

[150] Scharl A., Dickinger A., Murphy J. (2004). Diffusion and Success Factors of Mobile Marketing. Electronic Commerce Research and Applications, 4(2): 159-173.

[151] Schilling A, Coors V, Giersich M, Aasgaard R. Introducing 3D GIS for the Mobile Community Technical Aspects in the Case of TellMaris. IMC Workshop 2003 Proceedings : Assistance, mobility, applications Stuttgart: Fraunhofer IRB Verlag. 2003.

[152] Schmidt-Belz B., Poslad S., Nick A., Zipf A. (2002). Personalized and Location-based Mobile Tourism Services. Proceedings of the International Conference on Mobile HCI'02.

[153] Schmidt-Belz B, Laamanen H, Poslad S, Zipf A. Location-based mobile tourist services first user. Information and Communication Technologies in Tourism (ENTER 2003). 2003.

[154] Schwinger W., Grün C., Pröll B., Retschitzegger W., Schauerhuber A. (2002). Context-Awareness in Mobile Tourism Guides - A Comprehensive Survey. Technical Report, Johannes Kepler University Linz, IFS/TK.

[155] Sharp R, Rehman K. The 2005 UbiApp Workshop: What Makes Good Application-Led Research? IEEE Pervasive Computing, vol. 4, no. 3. 2005, pp. 80-82.

[156] Soffaiu, W., Maervoet, J. et al: (1999) A Mobile Tourist Decision Support System for Small Footprint Devices, Proceedings of the 10th International Work-Conference on Artificial Neural Networks (IWANN'2009): 1248-1255.

[157] Souffriau W., Vansteenwegen P., Vertommen J., Berghe G. V., Oudheusden D. A Personalized Tourist Trip Design Algorithm for Mobile Tourist Guides. Applied Artificial Intelligence, 22(10), pp. 964-985, 2008.

[158] Souffriau W., Maervoet J., Vansteenwegen P., Vanden Berghe G., Van Oudheusden D. (2009). A Mobile Tourist Decision Support System for Small Footprint Devices, Proceedings of the 10th International Work-Conference on Artificial Neural Networks (IWANN'2009): 1248-1255.

[159] Staab S., Werthner H. (2002). Intelligent Systems for Tourism - Introduction. IEEE Intelligent Systems,17(6): 53-55.

[160] Steinbock, D. (2007). The Mobile Revolution: The Making of Mobile Services Worldwide. Kogan Page.

[161] SunSPOT project, http://www.sunspotworld.com/, last visited January 2010.

[162] Sun Microsystems hot Java Browser. http://java.sun.com/products/archive/hotjava/, Sun Microsystems. Last visited August 2008.

[163] Sun Microsystems (2009). Java ME Platform. http://java.sun.com/javame/, last visited January 20

[164] Sun Developer Network (2006). Java ME: De-fragmentation. Retrieved December 13, 2009, from: HTTP://DEVELOPERS.SUN.COM/MOBILITY/REFERENCE/TECHART/DESIGN_GUIDELINES/.

[165] Su X., Khoshgoftaar T.M. (2009). A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence, Vol. 2009.

[166] Tsekouras G., Anagnostopoulos C., Gavalas D., Economou D. (2007). Classification of Web Documents using Fuzzy Logic Categorical Data Clustering. Proceedings of the 4th IFIP Conference on Artificial Intelligence Applications & Innovations (AIAI'2007), Vol. 247: 93-100.

[167] Ultra-Mobile PCs, The Official UMPC site, Last visited in 27/02/2007: http://umpc.com/default.aspx.

[168] Umlauft M, Pospischil G, Niklfeld G, Michlmayr E. LoL@, A Mobile Tourist Guide for UMTS. Journal on Information Technology & Toursim, H. Werthner and E. Veit (ed.) Congnizant, Vol. 5, No. 3. March 2003, pp. 151-164.

[169] Van Setten M., Pokraev S., Koolwaaij J. (2004). Context-Aware Recommendations in the Mobile Tourist Application COMPASS. Proceedings of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2004), LNCS 3137: 235-244.

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

[171] Vansteenwegen P. (2008). Planning in tourism and public transportation. PhD thesis, Katholieke Universiteit Leuven.

[172] Vansteenwegen P., Van Oudheusden D. (2007). The Mobile Tourist Guide: An OR Opportunity. Operational Research Insight, 20(3): 21-27.

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

[174] Virkus, R., Gülle, R., Rouffineau, T. (2009).Don't Panic – Mobile Developer's Guide to the Galaxy". White paper. http://www1.j2mepolish.org/downloads/MobileDevelopersGuideToTheGalaxy.pdf, Last visited August 2010.

[175] Vlahakis V, Ioannidis N& Karigiannis J, ARCHEOGUIDE: Challenges and Solutions of a Personalised 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.

[176] Vrdoljak, M., Vrdoljak, S.I., Skugor, G. (2000). Fixed-Mobile Convergence Strategy: Technologies and Market Opportunities. Communications Magazine, 38(2): 116-121.

[177] Windows market place for mobile, http://marketplace.windowsphone.com, Last visited July 2010

[178] W3C Mobile Web Initiative, http://www.w3.org/Mobile/, Last visited August 2010.

[179] 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 12/01/2008: http://www.archimuse.com/mw2004/papers/wilson/wilson.html.

[180] Woodruff A, P.M. Aoki, A. Hurst & T.M.H. Szymanski, Electronic guidebooks and visitor attention. In Proceedings of the 6th International Cultural Heritage Informatics Meeting. pp. 437-454, 2004. Last visited in 12/01/2008: http://www2.parc.com/csl/members/woodruff/publications/2001-Woodruff-ICHIM2001-VisitorAttention.pdf

[181] Wooseong K , Myungchul K , Kyounghee L , Chansu Y, Ben L, Link layer sssisted mobility support using SIP for real-time multimedia communications, Proceedings of the second international workshop on Mobility management & wireless access protocols, October 01-01, 2004, Philadelphia, PA, USA

[182] Wahlster, W., Reithinger, N., Blocher, A. (2001). SmartKom: Towards Multimodal Dialogues with Anthropomorphic Interface Agents. Proceedings of the International Status Conference: Lead Projects Human-Computer-Interaction: 23-34.

[183] Wahlster W. SmartKom A Transportable and Extensible Multimodal Dialogue System. International Seminar on Coordination and Fusion in MultiModal Interaction, Schloss Dagstuhl International Conference and Research Center for Computer Science, Wadern, Saarland, Germany. 2001.

[184] Yuan M.J, "Enterprise J2ME: Developing Mobile Java Applications", Prentice Hall PTR, 2004.

[185] Zancanaro M, Stock O & Alfaro I, Using Cinematic Techniques in a Multimedia Museum Guide. In Proceedings of the 9th International Conference on Intelligent User Interface, pp. 316 - 318, 2004.

[186] Zhang, D., Adipat, B. (2005). Challenges, Methodologies and Issues in the Usability Testing of Mobile Applications. International Journal of Human Computer Interaction, 18(3): 293-308.

LIST OF PUBLICATIONS & CITATIONS

International Journals (peer reviewed):

1. **M. Kenteris**, D. Gavalas and D. Economou, "Electronic Mobile Guides: A Survey", *accepted* from *Personal and Ubiquitous Computing* (IF: 1.554), Springer Science.

2. **M. Kenteris**, D. Gavalas and D. Economou, "Mytilene E-guide: A Multiplatform Mobile Application Tourist Guide Exemplar", *accepted* from *Multimedia Tools and Applications* (IF: 0.626), Springer Science.

3. **M. Kenteris**, D. Gavalas and D. Economou, "An Innovative Mobile Electronic Tourist Guide Application", *Personal and Ubiquitous Computing* (IF: 1.554), Springer Science, 13(2), pp. 103-118, February 2009.

4. D. Economou, D. Gavalas, **M. Kenteris** and G.E. Tsekouras, "Cultural Applications for Mobile Devices: Issues and Requirements for Authoring Tools and Development Platforms", *ACM SIGMOBILE Mobile Computing & Communications Review*, 12(3), pp. 18-33, July 2008.

5. D. Economou, D. Gavalas, **M. Kenteris** and K. Micha, "Multimedia Applications for Handheld Devices: Analysis of Requirements for Development Platforms and Application Authoring Tools", *Journal of Mobile Multimedia* (ISSN: 1550-4646), Rinton Press, 3(1), pp.65-87, March 2007.

Book chapter (peer reviewed):

1. **M. Kenteris**, D. Gavalas and D. Economou, "Mobile Tourist Applications: Design Criteria, Status and Trends", *accepted* as a book chapter for "Handbook of Research on Mobile Software Engineering: Design, Implementation and Emergent Applications", P. Alencar and D. Cowanto (Eds.), to be published by IGI Global (formerly Idea Group Inc.), USA.

International Conferences (peer reviewed):

1. **M. Kenteris**, D. Gavalas, G. Pantziou and C. Konstantopoulos, "Near-Optimal Personalized Daily Itineraries for a Mobile Tourist Guide", Proceedings of the *15th IEEE Symposium on Computers and Communications* (*ISCC'2010*), pp. 862-864, Riccione, Italy, 22-25 June 2010.

2. **M. Kenteris**, D. Gavalas and A. Mpitziopoulos, "A Mobile Tourism Recommender System", Proceedings of the *15th IEEE Symposium on Computers and Communications* (*ISCC'2010*), pp. 840-845, Riccione, Italy, 22-25 June 2010.

3. **M. Kenteris**, D. Economou, D. Gavalas and D. Zamplaras, "Deploying Thick Mobile Clients using Thin Client Architecture: a Case in Mobile Tourist Guides", Proceedings of the *1st World Summit on the Knowledge Society* (*WSKS'2008*), M.D. Lytras et al. (Eds.), Communications in Computer and Information Science (CCIS) Vol. 19, Springer, Athens, Greece, pp. 635-640, 24-28 September 2008.

4. **M. Kenteris**, D. Gavalas and D. Economou, "Evaluation of Mobile Tourist Guides", Proceedings of the *1st World Summit on the Knowledge Society* (*WSKS'2008*), M.D. Lytras et al. (Eds.), Communications in Computer and Information Science (CCIS) Vol. 19, Springer, Athens, Greece, pp. 603-610, 24-28 September 2008.

5. **M. Kenteris**, M. Vafopoulos and D. Gavalas, "Cultural Informatics in Web Science: A Case of Exploiting Local Cultural Content", Proceedings of the *12th Panhellenic Conference on Informatics* (*PCI'2008*), Samos, Greece, 28-30 August 2008.

6. **M. Kenteris**, D. Economou and D. Gavalas, "The Quest to Build a Multi-Platform Mobile Tourist Guide for the Municipal Council of Mytilene, Greece",

Proceedings of the *2nd International Scientific Conference of the E-Business Forum* in "*E-Business in Travel, Tourism and Hospitality*", Athens, Greece, 14 March 2008.

7. **M. Kenteris**, D. Gavalas and D. Economou, "Mobile Electronic Guides for the Masses: Optimizing Tourists Mobile Devices", Proceedings of the *11th Panhellenic Conference on Informatics* (*PCI'2007*), ISBN: 978-960-89784-0-9, Vol. B, pp. 625-634, Patras, Greece, 18-20 May 2007.

8. D. Economou, D. Gavalas and **M. Kenteris**, "Authoring Tools & Development Platforms: Requirements for Mobile Devices-Enabled Cultural Applications", Proceedings of the *11th Panhellenic Conference on Informatics* (*PCI'2007*), ISBN: 978-960-89784-0-9, Vol. B, pp. 419-432, Patras, Greece, 18-20 May 2007.

9. **M. Kenteris**, D. Gavalas and D. Economou, "Software Analysis and Design for Mobile Devices: A Case Study in Tourism", Proceedings of the *24th EuroCHRIE Congress* (*EuroCHRIE'2006*), Thessaloniki, Greece, 25-28 October 2006.

10. **M. Kenteris**, D. Gavalas and D. Economou, "Developing Tourist Guide Applications for Mobile Devices Using the J2ME Platform", Proceedings of the *2006 Conference on Mobile Computing and Wireless Communications* (*MCWC'2006*), O. Haddad (Eds), ISBN: 978-9957-486-00-6, pp. 178-183, Amman, Jordan, 17-20 September 2006.

11. D. Gavalas, D. Economou and **M. Kenteris**, "The Wireless Internet Technology Landscape", Proceedings of the *2006 Conference on Mobile Computing and Wireless Communications* (*MCWC'2006*), O. Haddad (Eds), ISBN: 978-9957-486-00-6, pp. 171 – 177, Amman, Jordan, 17-20 September 2006.

12. **M. Kenteris**, D. Gavalas and D. Economou, "A Novel Method for the Development of Personalized Mobile Tourist Applications", Proceedings of the *5th IASTED International Conference on Communication Systems and Networks* (*CSN'2006*), C.E. Palau Salvador (Eds.), ACTA Press, ISBN: 0-88986-606-6, pp. 208-212, Palma de Mallorca, Spain, 28-30 August 2006.

13. D. Gavalas, D. Economou and **M. Kenteris**, "Status and Trends of Wireless Web Technologies", Proceedings of the *5th IASTED International Conference on Communication Systems and Networks* (*CSN'2006*), C.E. Palau Salvador (Eds.), ACTA Press, ISBN: 0-88986-606-6, pp. 219-224, Palma de Mallorca, Spain, 28-30 August 2006.

14. **M. Kenteris**, D. Gavalas and D. Economou, "The Use of Mobile Telephones to Promote Cultural Information", Proceedings of the *3rd International Conference on Museology* (*ICM'2006*), *poster* presentation, Mytilene, Greece, 5-8 June 2006.

Under Review (υπό κρίση):

1. D. Gavalas and M. Kenteris, "A Mobile Tourism Pervasive Recommendation System", submitted to IEEE Pervasive Computing.

Citations

- M. Kenteris, D. Gavalas and D. Economou, "An Innovative Mobile Electronic Tourist Guide Application", Personal and Ubiquitous Computing (ISSN: 1617-4909), Springer Science, 13(2), pp. 103-118, February 2009

    1. H. Cramer, "People's Responses to Autonomous and Adaptive Systems", PhD Thesis, SIKS (Dutch Research School for Information and Knowledge Systems), April 2010.
    2. W. Premchaiswadi, "An image search for tourist information using a mobile phone", WSEAS Transactions on Information Science and Applications, 7(4), pp. 532-541, 2010.
    3. L. T. Yong, C. Y. Qi, C. S. Yee, A. Johnson, N. K. Hoong, "Designing and Developing a PDA Food Ordering System Using Interaction Design Approach: A Case Study", Proceedings of the 2009 International Conference on Computer Technology and Development, November 2009.
    4. R. Ranaweera, M. Cohen, N. Nagel, M. Frishkopf, "Virtual World, World Music-Folkways in Wonderland", Proceedings of the International

Workshop on the Principles and Applications of Spatial Hearing (IWASH'09), November 2009.

5. I. Aslan, D. Menon, "Component-based development of mobile assistants with the ELEPHANT system", Proceedings of the 6th ACM International Conference on Mobile Technology, Application & Systems (Mobility'2009), September 2009.
6. L. Laporte, P. Eyckerman, B. Zaman, "Designing a mobile task based UI for tourists", Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'09), September 2009.
7. K. Rechert, "MobIS: A pragmatic framework for location based services", Proceedings of the 6th Workshop on Positioning, Navigation and Communication (WPNC'2009), pp. 141-144, March 2009.
8. A.R. Binti Mohd Faizal Kim, "Mobile PLUS ExpressWays Travel Assistant System (M-PEX)", PhD Thesis, Universiti Utara, Malaysia, 2009.

- D. Economou, D. Gavalas, M. Kenteris and G.E. Tsekouras, "Cultural Applications for Mobile Devices: Issues and Requirements for Authoring Tools and Development Platforms", ACM SIGMOBILE Mobile Computing & Communications Review, 12(3), pp. 18-33, July 2008.

1. K. Rechert, "Challenges and Success Criteria for Mobile Services in Tourism", Proceedings of the 17th International Conference on Information Technology and Travel & Tourism (ENTER'2010), February 2010.
2. K. Rechert, "MobIS: A pragmatic framework for location based services", Proceedings of the 6th Workshop on Positioning, Navigation and Communication (WPNC'2009), pp. 141-144, March 2009.
3. K. Uotila, I. Huvila, J.-P.a Paalassalo, I. Helenius, J. Lindholm, M. Lääperi, L. Puolamäki, S. Wahlqvist, "Learning, Access, and Mobility (LAMB) for Cultural Heritage Education", Proceedings of the 37th Annual Computer Applications and Quantitative Methods in Archaeology Conference (CAA'2009), March 2009.

- M. Kenteris, D. Gavalas and D. Economou, "A Novel Method for the Development of Personalized Mobile Tourist Applications", Proceedings of the 5th IASTED International Conference on Communication Systems and Networks (CSN'2006), C.E. Palau Salvador (Eds.), ACTA Press, ISBN: 0-88986-606-6, pp. 208-212, Palma de Mallorca, Spain, 28-30 August 2006.

  1. C. Sahd, H. Thinyane, M. Thinyane, "An Evaluation of a Mobile Phone Platform as a Convergent Technology for Text Based Communication", Proceedings of the Southern Africa Telecommunication Networks and Applications Conference (SATNAC'2009), August 2009.