University of Crete
Computer Science Department

AN INTEGRATED PLATFORM FOR LOCATION–AWARE INFORMATION SYSTEMS: MOBILE MULTIMEDIA NAVIGATOR

MASTER'S THESIS

by
Emmanouel Zidianakis

Heraklion, November 2008

Στους γονείς μου Γιώργο, Πόπη

στην αδερφή μου Ιωάννα

INTEGRATED PLATFORM FOR
LOCATION-AWARE INFORMATION SYSTEMS:
MOBILE MULTIMEDIA NAVIGATOR

By
ZIDIANAKIS MANOS

A thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science

Author: ______________________________
Zidianakis Manos, Department Of Computer Science

Board of enquiry:

Supervisor ______________________________
Anthony Savidis, Associate Professor

Member ______________________________
Constantine Stephanidis, Professor

Member ______________________________
Maria Papadopouli, Assistant Professor

Approved by: ______________________________

Heraklion, November 2008

University Of Crete
Computer Science Department

AN INTEGRATED PLATFORM FOR LOCATION–AWARE INFORMATION SYSTEMS: MOBILE MULTIMEDIA NAVIGATOR

MASTER'S THESIS

Emmanouel Zidianakis

Abstract

Mobile location-aware information systems are capable to deliver position-dependent information over a portable device for users that are primarily on the move. Their support requires a combination of core or User Interface subsystems, ranging from authoring, database, middleware, monitoring, visualization, network and navigation.

In this Thesis we report on the design and implementation of the *Mobile multimedia navigator*, being the end-user system running at the portable device side, offering location-based information delivery and site navigation, supporting:

- Advanced User Interface for exhibit information
- Extensible and prioritized location-sensing
- Scenario-driven navigation
- Combining location triggered with user initiated information presentation
- Position visualization with area maps

The current implementation of the navigator encompasses Wireless LAN position sensing (path tracking) combined with Infrared beacons (area detection) for enhanced accuracy. Additional features include: automatic system update through the network, auto locking in case the device is used outside the expected area boundaries, internal communication with a device surveillance system, and automatic database synchronization.

University Of Crete
Computer Science Department

AN INTEGRATED PLATFORM FOR LOCATION–AWARE INFORMATION SYSTEMS: MOBILE MULTIMEDIA NAVIGATOR

ΟΛΟΚΛΗΡΩΜΕΝΗ ΠΛΑΤΦΟΡΜΑ ΓΙΑ ΠΛΗΡΟΦΟΡΙΚΑ ΣΥΣΤΗΜΑΤΑ ΜΕ ΕΠΙΓΝΩΣΗ ΘΕΣΗΣ:

ΦΟΡΗΤΟΣ ΠΟΛΥΜΕΣΙΚΟΣ ΠΛΟΗΓΟΣ

Μεταπτυχιακή Εργασία

Εμμανουήλ Ζηδιανάκης

Περίληψη

Τα φορητά πληροφοριακά συστήματα με επίγνωση θέσης μπορούν να προσφέρουν πληροφορίες οι οποίες θα εξαρτώνται από τη εκάστοτε θέση, σε χρήστες που βρίσκονται κυρίως εν κινήσει, με την βοήθεια φορητών συσκευών. Η υποστήριξη τους απαιτεί ένα συνδυασμό από βασικά υποσυστήματα ή διεπαφές χρήστη, που περιλαμβάνουν υποσυστήματα δικτύου, βάσεων δεδομένων, συγγραφής περιεχομένου, παρακολούθησης συστήματος, πλοήγησης, οπτικής αναπαράστασης δεδομένων καθώς και υποσυστήματα διαμεσολάβησης τους.

Στην παρούσα εργασία περιγράφουμε το σχεδιασμό και την υλοποίηση ενός *Φορητού πολυμεσικού πλοηγού* (*Mobile multimedia navigator)* ο οποίος αφόρα ένα σύστημα τελικών χρηστών, κάνοντας χρήση φορητών συσκευών πλοήγησης. Κύριο χαρακτηριστικό είναι η επίγνωση της εκάστοτε θέσης του χρήστη και η

αναπαράσταση της αντίστοιχης πληροφορίας. Άλλα χαρακτηριστικά που υποστηρίζονται:

- Προηγμένο μέσο αλληλεπίδρασης με τον χρήση για την παρουσίαση σημείων ενδιαφέροντος.
- Επεκτάσιμο σύστημα υπολογισμού θέσης, μέσω αξιόλογης των μεθόδων χωρικής ανεύρεσης θέσης.
- Πλοήγηση βάση σεναρίου ξενάγησης
- Αυτόματη παρουσίαση περιεχομένου καθοδηγούμενη είτε από την εκάστοτε θέση, είτε από τον ίδιο τον χρήστη.
- Απεικόνιση εκάστοτε θέσης με την βοήθεια συστήματος χαρτών.

Η τρέχουσα υλοποίηση του φορητού πολυμεσικού πλοηγού περικλείει σύστημα εντοπισμού θέσης κάνοντας χρήση ασύρματου δικτύου (επίγνωση διαδρομής) συνδυαζόμενο με πομπούς υπέρυθρων (ανίχνευση περιοχών) για περισσότερη ακρίβεια. Επιπρόσθετα χαρακτηριστικά περιλαμβάνουν: αυτοματοποιημένη διαδικασία αναβάθμισης λογισμικού μέσω δικτύου, προστασία συσκευής με αυτόματο κλείδωμα σε περίπτωση χωρικής υπέρβασης, εσωτερική επικοινωνία με σύστημα παρακολούθησης και εποπτείας όλων των φορητών πλοηγών, καθώς και αυτοματοποιημένο συγχρονισμό δεδομένων.

Ευχαριστίες (Acknowledgements)

Θα ήθελα να ευχαριστήσω τον επόπτη της μεταπτυχιακής μου εργασίας Αντώνη Σαββίδη για την συνεχή καθοδήγηση και υποστήριξή του τα τελευταία δύο χρόνια στο πλαίσιο της συνεργασίας μας στο Εργαστήριο Επικοινωνίας Ανθρώπου-Μηχανής, του Ινστιτούτου Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας και ειδικότερα στο πλαίσιο της εκπόνησης της μεταπτυχιακής μου εργασίας.

Θα ήθελα επίσης να ευχαριστήσω τον Γραμμένο Δημήτρη για την ουσιαστική του υποστήριξη στην δομή της μεταπτυχιακής μου εργασίας, όπως επίσης και την Μαργαρίτα Αντώνα για την διόρθωση της ορθογραφίας της. Θα ήθελα να αναφέρω και την ουσιαστική συμβολή του αδελφικού μου φίλου Στέλιου Σημαιάκη δίνοντας μου πάνω απ’ όλα το απαιτούμενο κουράγιο.

Τέλος, ένα μεγάλο ευχαριστώ στους φίλους μου για τη βοήθεια και τη στήριξη που μου παρείχαν όλα αυτά τα χρόνια. Τους εύχομαι ότι καλύτερο στην σταδιοδρομία τους.

Table of Contents

List of Figures

List of Tables

1. Introduction

In times of increasing possibilities to access information of all kinds, it is often difficult to find efficiently what one is interested in. Thus, a growing demand for methods to structure information according to individual needs can be observed. Location aware applications offer an approach for this purpose: they are running on mobile computing devices, measure the positions of their users and support location-based services.

Since this platform is designed to contain huge amounts of data and various services which many users can access simultaneously, it is organized as a distributed system. Starting from a use case analysis, architecture with all necessary system components was designed for the infrastructure.

In this context, it has to be considered that different kinds of mobile devices (equipped with different sensors to capture the current contexts of the users) are accessing the infrastructure. Therefore, the Mobile Navigator offers an adequate interface to cover the demands of all possible. Within the integrated platform, the spatial data are offered in multiple representations in order to meet the individual needs of different applications.

In the Mobile Navigator multiple sources for user-location information are embedded, including explicit positioning as coordinates in a 2D place (e.g., X, Y and optionally a DIR vector) or implicit positioning as the result of interpretation data coming from sensory modules. The latter is managed by various technologies like special tags (e.g., infrared beacons). The former is possible mainly via GPS, for outdoor environments, and WLAN positioning engines, relying on signal fingerprints, for both indoor and outdoor setups.

Within this thesis, the term “**information item**” is used to describe any individual part of a place, such as a museum, for which location-based information is available. E.g.: a single exhibit, a collection of exhibits, a thematic area, a whole room, or even a museum floor.

In many cases exhibits must to be placed very close to each other, with relative distances that cannot be handled by the precision of positioning systems. For instance, it is common to place ancient helmets, shields and arcs close to each other, each having a separate information unit. For such scenarios, the platform supports the grouping of multiple information items into a single chunk called an **information area**, physically being defined by a polygon encompassing the locations of all respective information items. Generally, this has as result each different information area consist of different information items (not information area overlapping) while this approach cutting out the capability of context-depended splitting of the information items.

The Mobile Navigator architecture’s key constituent components rely on a generic location-sensing interface enabling sensing APIs to be dynamically loaded (as DLLs). As mentioned, the followed method splits the global information plane into independent maps, where maps encompass polygonal information areas, which enclose the actual information points (information items), i.e. the real exhibits (***Figure 1***).

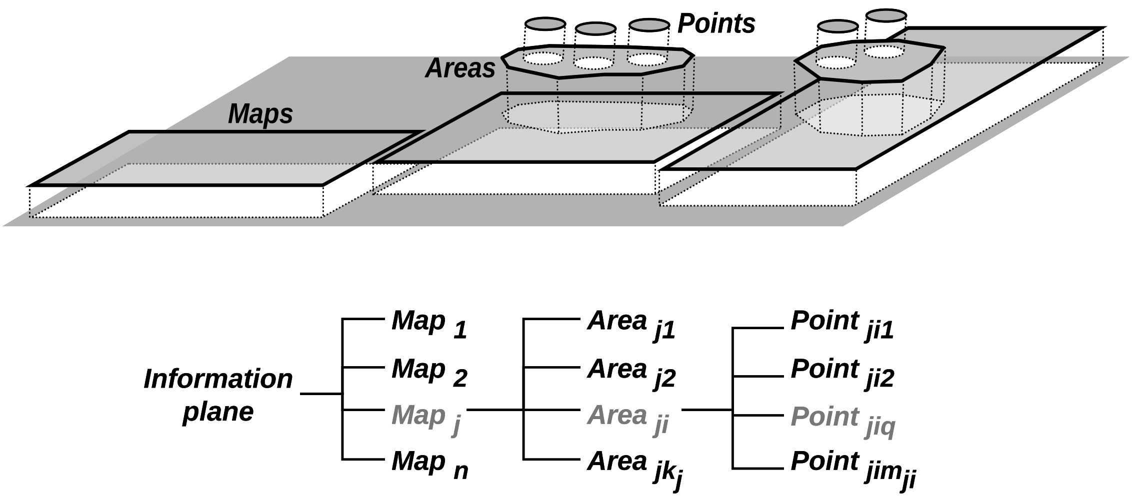


Figure 1: Split of the information plane to maps, areas and information items.

In this context, the location-sensing interface allows distinct technologies to return either a point (higher precision) within the currently active map plane, or alternatively the identifier of a polygonal area (lower precision) that is associated to a particular physical area (e.g., room, hall, corridor, corner, stares, etc.).

Following this approach, alternative techniques (e.g., infrared beacons or radio tags) than the common point-based (e.g., GPS) can be deployed as carriers of area identification information (e.g., an infrared beacon can be programmed to simply emit the logical identifier of an area denoting a specific room).

Effectively, when these are combined with typical point-based location-sensing technologies, it helps to resolve possible ambiguities or precision problems, or overcome other types of practical obstacles. The capability to support multiple prioritized location-sensing technologies is crucial, since, in some cases, the particular characteristics of the installation site, or other types of restrictions concerning possible physical interventions, may directly exclude specific technologies. For instance, inside archeological sites it is forbidden to use any type of equipment (i.e. no tags, no power supply). Additionally, the prioritized use of location-sensing APIs reflects the different reliability of the position outcome in different technologies.

2. Background and Related Work

2.1 Related projects

The existing implemented location-aware applications can be classified based on several criteria, namely, the services they provide (information guides vs. navigation systems), the environment they work in (indoor vs. outdoor), the type of the interface (map-based vs. text-based, interactive vs. non-interactive), the content type (predefined or not) and their adaptation capabilities.

C-Map [1] is a tour guidance system which, based on location and individual interests, provides information to visitors at exhibitions. It is based on a prototype of the mobile assistant. In C-Map a personal guide agent with a life-like animated character on a mobile computer guides users using exhibition maps which are personalized depending on their physical and mental contexts.

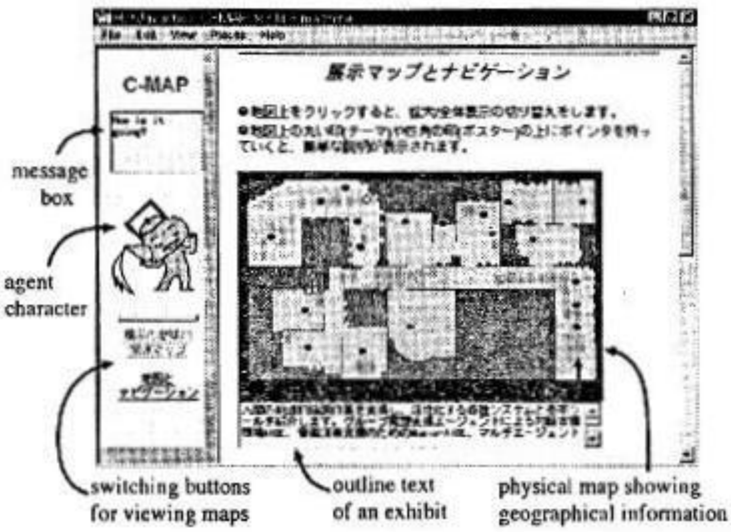


Figure 2: The mobile assistant display showing a map

C-Map is based on the Active Badge System [2] (ABS) for user location detection. The ABS server has many sensors at the exhibits sites to detect the locations of the users by infra-red linking to the badges they wear (***Figure 3***). The server gathers the latest sensor data and updates the location data of all users. The sensors can detect badges within a 1 to 2 meter perimeter.

Figure 3: Active Badge System

Cyberguide [3] is a collection of intelligent tour guides which provide information to tourists based on knowledge of their position and orientation. The Cyberguide Project consists of building prototypes (***Figure 4***) of a mobile context-aware tour guide. Knowledge of the user's current location, as well as the history of past locations is used to provide more of the kind of services that are expected from a real tour guide. The architecture and features of a variety of Cyberguide prototypes were developed for indoor and outdoor use on a number of different hand-held platforms.

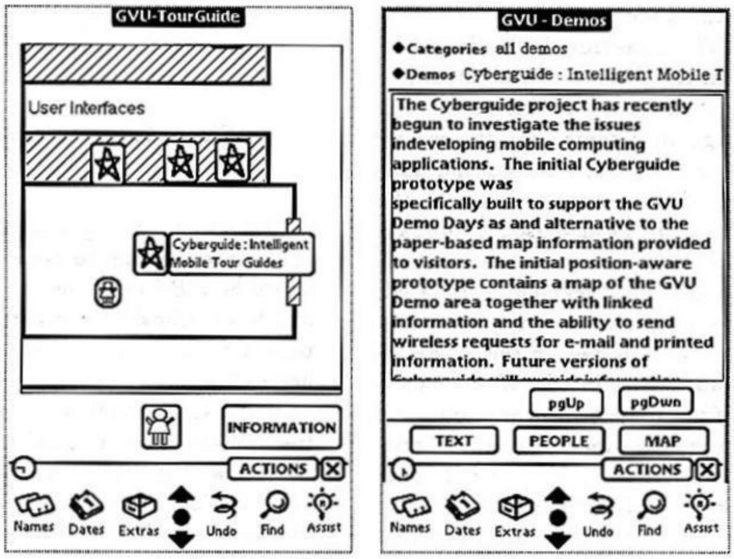


Figure 4: The map (left) and information (right) interfaces of the initial Cyberguide prototype

Metronaut [4] is an application developed for schedule management and guidance instructions for a visitor to a university campus. Metronaut is a novel wearable computer which captures information, senses position, provides wide range communications, consumes less than one watt of power, and weighs less than one pound. Metronaut employs a bar code reader for information input and position location, a two-way pager for communications, and an ARM processor for computation. Metronaut's application is schedule negotiation and guidance instructions for a visitor to the CMU campus. The visitor's position is determined from reading bar codes at information signs around campus. Modifications to the schedule are negotiated using the two-way pager for communications with the campus computing infrastructure. Metronaut is alternatively powered by a mechanical flywheel converting kinetic energy to electrical energy.

City Guide [5] enables a user to see his position on a map and request restaurant and hotel information. City Guide enables a user to know their geographical position by looking at their terminal screen, which displays a map with their position marked on it. City Guide is based on GPS and GSM network to measure the user position (***Figure 5***).

Figure 5: City Guide v1

Augmentable Reality [6] allows users to dynamically attach digital information such as voice notes or photographs to the physical environment. Audio Aura [7] provides information via auditory cues based on people's physical actions in the workplace. These systems use predefined locations and are designed for users to find each other or objects in the environment. Forget-Me-Not [8] is a wearable device which records interactions with people and devices, and stored this information in a database for later query.

The Remembrance Agent [9] provides text information relevant to the user's context, for example class notes when entering a specific classroom. It is a continuously running proactive memory aid that uses the physical context of a wearable computer to provide notes that might be relevant in that context. A currently running prototype is described, along with future directions for research inspired by using the prototype (***Figure 8***).

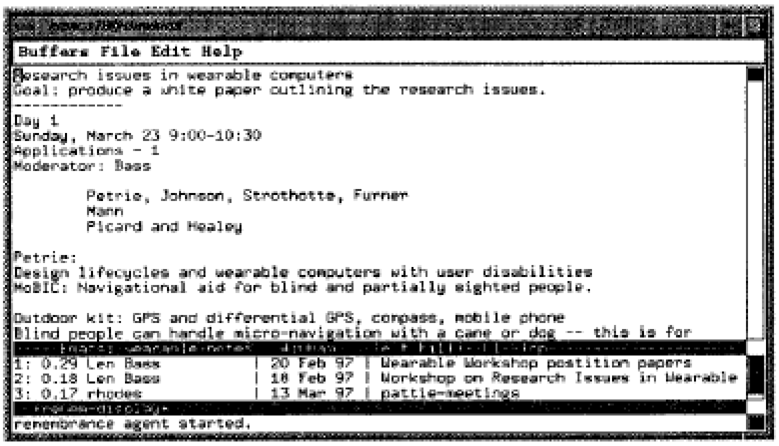
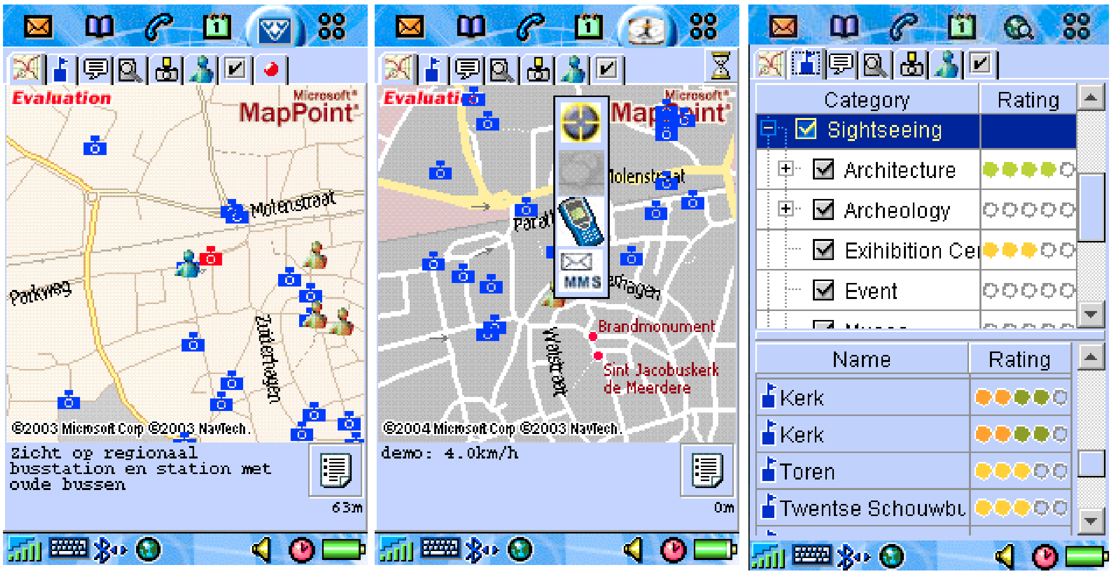


Figure 6: A screenshot of the remembrance agent

A Semantic-based Framework for Context-aware Pedestrian Guiding Services [10] offers a Context Pedestrian Guiding platform (CG) to explore intelligent pedestrian navigation. This approach, focusing on indoor environments, is structured around an RDFS-ontology for capturing and formally modeling contextual information and also utilizes user-centered design for developing interactive map-based presentations for accommodating pedestrians. For the CG prototype, an IEEE 802.11 network infrastructure is used for both positioning and communication purposes, exploiting the existence of a positioning system that runs on the client (e.g., CLS [11]).

Compass [12] is a context-aware mobile tourist application that adapts its services to the user's needs based on both the user's interests and his current context. In order to provide context-aware recommendations, a recommender system has been integrated with a context-aware application platform. The location is either obtained from the mobile network or from other devices such as GPS receivers. It uses various external map services through proprietary interfaces, a map service providing orthophotos and a map service providing cadastral maps. It builds on the open Web Architectures for Services Platform [13], which supports context-aware applications based on web services, described in OWL [14]. Compass serves a tourist with information and services (ranging from buildings to buddies) needed in his specific context that is interesting to him given his goal for that moment.



Sotto Voce [15] is an electronic guidebook prototype developed for guiding visitors in a historic house in Woodside, California. The system does not include any automatic location-awareness technology. Visitors can obtain information about objects in their environment using a visual interface. The interface consists of photographs taken facing the four walls of the room. Visitors can change the viewing perspective (i.e., see a different wall) by pressing a button on the device. Each photograph is actually an image map, in which different objects represent active areas. When visitors tap on an active area (i.e., an object) an audio description about it is provided. In the second version of the system, an *eavesdropping* function was added, where visitors are paired and each one of them can hear the content accessed by their companion (***Figure 7***).

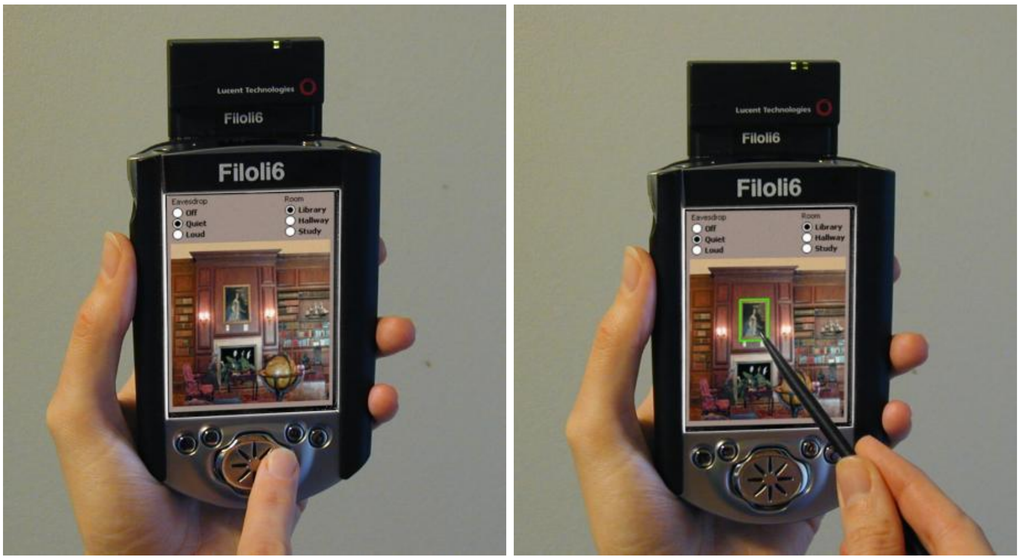


Figure 7: Sotto Voce

The IrReal [16] is a building information and navigation system based on Palm Pilot PDAs and a set of IR emitters located throughout a building. When the users are walking in the building, the IR sends them information related to their current position. All information, which can include formatted text and vector graphics which may contain hyperlinks to each other, is transmitted by the beacons using a proprietary format as a set of interconnected pages. This system supports implicit tracking; the mobile device displays whatever information is received by the transmitter in sight. IrReal was installed and tested at the German computer fair CeBIT 2000 in Hannover. Two basic drawbacks of the system are that the power consumption of the transmitter hardware is high and that each beacon is must be connected to a PC in order to broadcast the pages (***Figure 8***).

Figure 8: The IrReal building information and navigation

The Hippie prototype system [17], developed in the HIPS project, was tested at an art exhibition. The system uses infrared to locate the position of the visitor and an electronic compass to detect orientation. IR beacons are installed on the walls underneath each exhibit, as well as above each door of the museum. The beacons send an ID to a receiver being fastened on the jacket of the visitor or attached to the user's headphones and connected to the handheld computer. Based on the current user location related information is presented. The system tries to build a user model using predefined knowledge about user interests and knowledge inferred by history of user interactions and the choices. This user model is employed in order to adapt the presented information to the interests, the knowledge and the presentation preferences of the user. Since, by the time the system was developed, there were no PDAs available with a PCMCIA slot for the wireless and for infrared receiver the system was tested using a laptop computer (***Figure 9***).

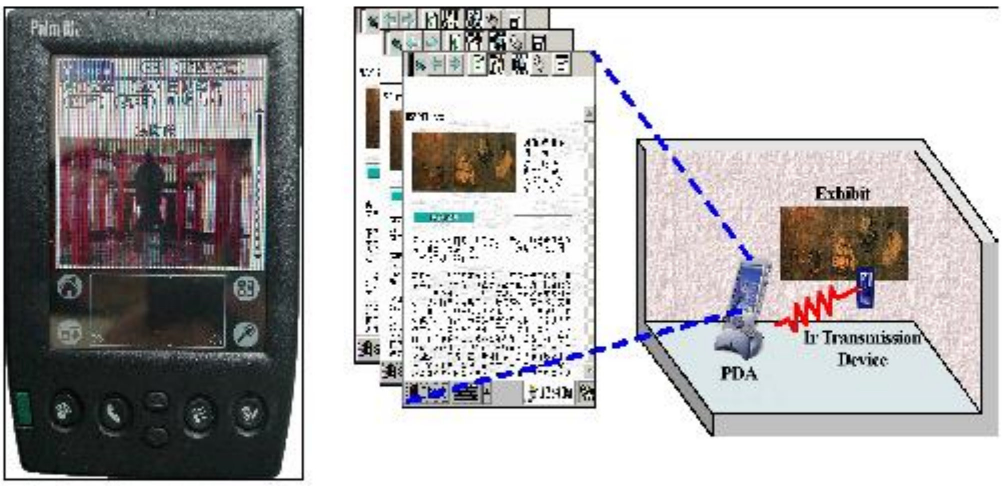
Figure 9: The Hippie prototype system

The Marble Museum of Carrara in Italy uses IR emitters to determine user position [18]. The emitters are located at the entrance of each room, on the ceiling, so that the presence of visitors does not interfere with the IR signal access.

When a user enters a new room the emitters sends an identifier to the PDA, and the application detects it and presents a museum map, where the current section is highlighted. After that, an audio presentation of the main characteristics of the section and a map indicating the location of the artworks in that section are provided. This system was one of the first fully operating systems in a museum (***Figure 10***).

Figure 10: The system used by the Marble Museum of Carrara in Italy

The Taipei Astronomical Museum in Taiwan used a Palm IIIc PDA running Palm OS which retrieves information about the position of the visitor through infrared emitting devices [19]. When the PDA receives an identifier of an exhibit it automatically presents a related HTML page containing text and graphics



(

Figure 11).

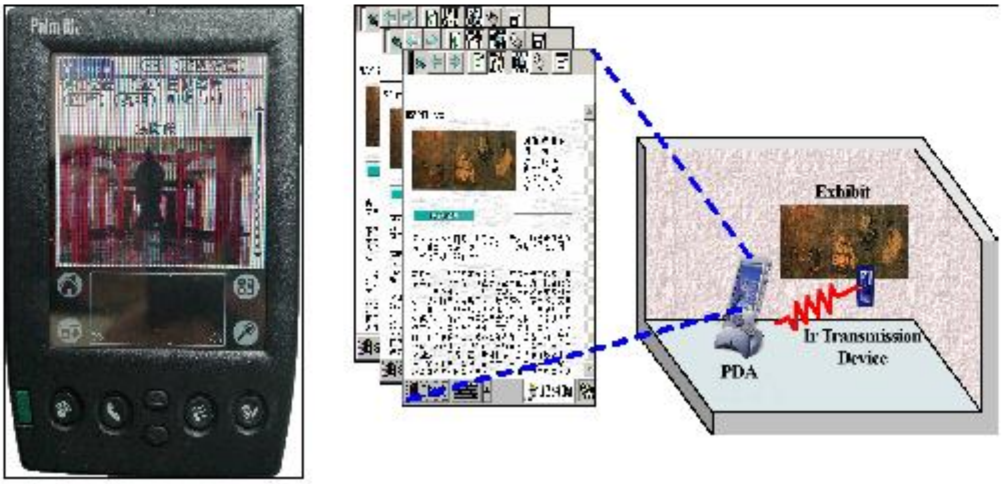


Figure 11: The tour guide at the Taipei Astronomical Museum in Taiwan

At the National Museum of History of Taiwan an experimental system was tested where the physical environment is represented on a handheld as a set of panoramas [20]. The handheld is equipped with an orientation sensor which is used to automatically align the panorama with the real world. The user can interact with an object within the environment by selecting the corresponding item on the display. This system has several drawbacks, since: (a) the orientation sensor is too big and hanging outside the hand-held device; (b) there are no data about the actual location of the visitor, so the same representation will be provided irrespectively of his/her relative position in the exhibition area; (c) the application of this approach may be restricted by the museum's layout; (d) the system provides a 1-to-1 photographic representation of the "real" museum; this means that whenever there is even a minor change to the museum all the photographic data will have to be recreated and reentered to the system in order to be accurate (***Figure 12***).

Figure 12: The panorama-based guide at the National Museum of History of Taiwan

The Exploratorium, an interactive science museum housed in San Francisco's Palace of Fine Arts, developed the Electronic Guidebook, through which visitors can access additional information about the exhibits but also dynamically create a Web-page during their visit, containing their personal photos and suggestions for home activities [21]. The device used was a HP Jornada 690/720 (***Figure 13***). Radiofrequency identification (RFID) tags are used to link visitors with exhibit-related content delivered by a Web-based server, while to enable the handheld to be context-aware, a point-of-information station was designed to hold a HP Cooltown infrared beacon which sends the URL to a handheld that is in close proximity to an exhibit. The web page corresponding to this URL is then automatically downloaded from the content server and displayed by the PDA's browser.

Figure 13: The Electronic Guidebook at the Exploratorium science museum in San Francisco's Palace

Further research with the Electronic Guidebook revealed several drawbacks of the system, such as [22]: (a) visitors found it difficult and cumbersome to carry the device while operating the museum exhibits; (b) visitors found that a handheld promoted a sense of isolation; (c) multimedia content was difficult to hear in the Exploratorium's noisy environment; and (d) visitors preferred having their hands free to manipulate exhibits. Thus, they designed a new application called the I-Guide (His, 2004) which instead of guiding the visitors it aims at recording and capturing user experiences at the museum for later reflection. Using an RFID device (e.g., an electronic watch, a card, a yo-yo, a remote clicker), a visitor can bookmark the exhibit he/she is visiting (which sends his/her unique ID to the network via an RFID transceiver mounted on the exhibit), capture a memorable photo of himself/herself at or near an exhibit by activating a camera, and/or trigger a printer to create a souvenir of a visit. An RFID transceiver records and sends the visitor identification number to the network and database system while also tracking a visitor's conceptual pathway through the museum. After the museum visit, the visitor later reviews additional science articles, conducts personally relevant science investigations, explores online exhibits, and downloads hands-on kits and other science activities at home via a personalized Web page (***Figure 14***).

Figure 14: The I-Guide. Prototypes for RFID token (left) and a prototype of a personalized Web page created during a museum visit (right)

A rather bulky research prototype guide [23] is based on a tablet PC with a touch screen, a webcam and a Bluetooth receiver. The system used computer vision techniques to recognize the object a visitor is interested in by analyzing a photo taken by the visitor. In order to improve recognition time and accuracy, Bluetooth nodes were used to pinpoint the room in which the photo was taken. The prototype has been demonstrated to visitors of the Swiss National Museum in Zurich (***Figure 15***).

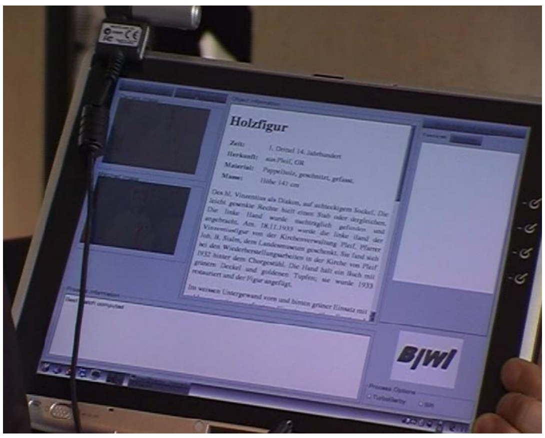


Figure 15: Tablet PC-based system by Bay, Fasel and Van Gool

A similar approach was followed by Albertini [23], for developing an experimental prototype for retrieving information related to paintings. The visitor carries a PDA equipped with a webcam and headphones. In order to retrieve information about a painting, or a detail, the visitor points the web cam toward the painting and feedback (e.g., labels) are overlaid on the camera view presented on the PDA's screen (***Figure 16***). The visitor can select these labels to access a relevant multimedia presentation. The vision recognition engine runs on a dedicated Linux server and communicates with the PDA through a Wi-Fi network.

Figure 16: Retrieving context-related information using machine vision

A collaborative children's "question and answer" museum game, called Musex [25], was developed using PDAs equipped with an RFID tag reader/writer and a wireless LAN card. An RFID tag with a unique ID was added to each museum exhibit. Whenever a child approaches an RFID, his/her PDA reads the ID number from the tag and requests from a server the corresponding question data (***Figure 17***).

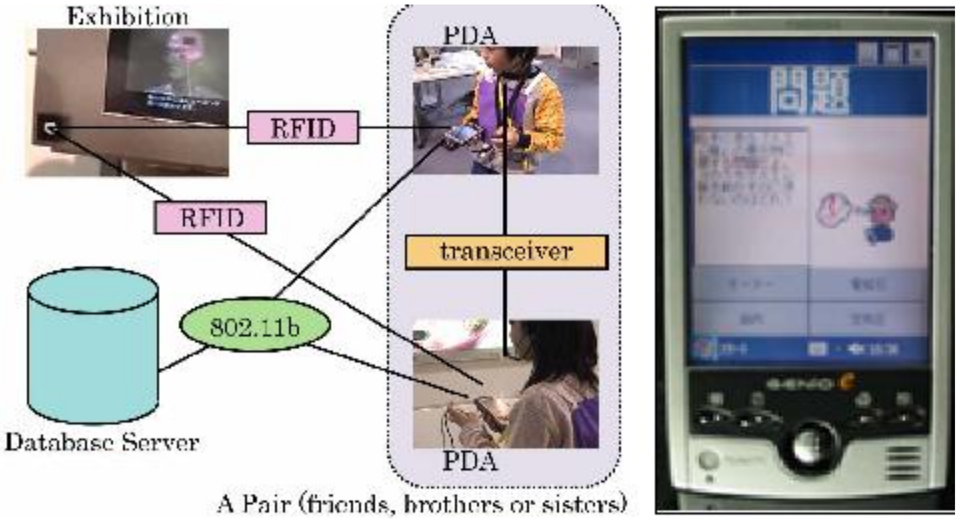


Figure 17: The Musex

The Active Print project [26] has developed an application in which users use the camera of their mobile phones to read barcode signs in order to obtain content about their surroundings at seaside locations. The application was made available to the general public in support of "Coast", a BBC television series about the UK coastline (***Figure 18***).

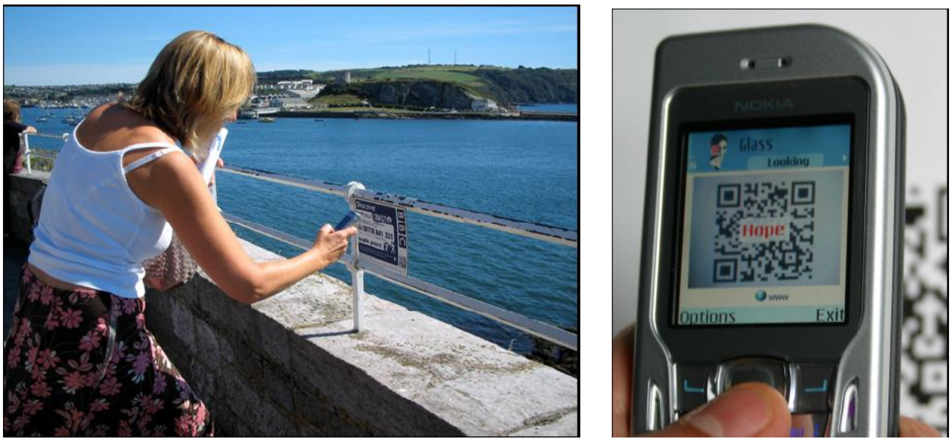


Figure 18: Using a mobile phone camera to read barcode signs

The key distinct features of the Mobile Multimedia Navigator are the extensible location-sensing system and the interactive graphic user interface. More specifically, the first one includes extensible and prioritized location sensing technologies such as WLAN, GPS, IR beacons and allows for the integration of additional technologies in a flexible and efficient way (extensible). The second one, refers to an advanced user interface for viewing information related to exhibits (i.e., information points).

The interface supports both location-triggered and user-initiated information presentation and provides user position indication through area maps, supporting zooming, scrolling and controlling level-of-detail. Additional features comprise scenario-driven navigation, multilingual content, audio and speech feedback, automatic content and system update through the data synchronization mechanism, auto locking when used outside predefined boundaries, communication with a surveillance (monitoring) system and finally recording detailed navigation statistics.

2.2 Technologies used

In most existing systems the basic interaction device used is a PDA. The main advantages of PDAs are can be summarized as follows:

- They are lightweight devices that can easily be carried using a single hand by both adults and children.
- They can render any type of multimedia at a high quality.
- They now have quite strong processors and large memory storage capacity, which can also be extended through memory cards.
- Most of them are (or can be easily and inexpensively) equipped with numerous communication and location-sensing technologies, such as Bluetooth, IrDA, wireless network, radio-frequency reader, as well as GPS.

- Some of them are also equipped with photo/video cameras that can be used by the visitors for taking souvenir photos, but also by the system for employing computer vision techniques.

Except PDAs, in a very few projects tablet PCs have been used, but mainly as a means for experimenting with technologies that were not yet available for PDAs.

Existing approaches for achieving location-awareness can be classified in two basic categories:

(a) User-driven position awareness: The end-users (visitors) are responsible for feeding into the system their current location or, the information item they are looking at. The most prominent related methods/technologies that have been employed up to now are following:

a.1. Numbered labels: Each information item carries a distinctive label with a unique numeric id. Visitors have to enter this number to their PDA, using a hardware or software (i.e., on-screen) keyboard in order to retrieve information (typically a narration or a video) related to the information item.

- **Pros**
  - High accuracy (except if the user types in the wrong number).
  - Inexpensive.
  - Very easy to implement.
  - Practically no maintenance cost.
- **Cons**
  - Visitors have to discover the labels. Since not all information items may have labels, visitors will not be sure where and when to look for them. Furthermore, locating a label may be difficult when there are other people standing in front of it.
  - It cannot be used by people with motor and vision impairments.
  - Its ease of use is questionable.

- It shifts the visitors' focus from the museum exhibits to their labels, thus completely disrupting their main task which is to visit and experience the museum and diminishing their overall experience.

a.2. Barcode labels: Each information item carries a distinctive label with a barcode. PDAs are equipped with a barcode reader. Visitors have to scan the label using the barcode reader.

- **Pros**
  - High accuracy.
  - Low cost.
  - Easy to implement.
  - Very low maintenance cost.
- **Cons**
  - Visitors have to discover the labels. Since not all information items may have labels, visitors will not be sure where and when to look for them. Furthermore, locating a label may be difficult when there are other people standing in front of it.
  - The process of label scanning can become tiring after a while.
  - Its ease of use is questionable.
  - There is no optimum strategy for placing the labels so that people with diverse requirements and physical characteristics can (easily) reach them.
  - It cannot be used by people with motor and vision impairments.
  - It requires that the visitor's PDA approaches to a very short distance from the tag. In crowded situations this may result in long waiting lines, or just in visitors avoiding the most popular exhibits.

(b) System-driven (automatic) position awareness: The system employs one or more types of positioning technologies to automatically identify the visitor's location.

The most prominent related methods/technologies that have been employed up to now are following:

b.1. Infrared (IrDa): One, or more, infrared emission devices (beacons) are placed at each information items. Each beacon emits a unique ID. The PDA reads the emitted signal through an infrared port and accordingly provides a presentation of the related PoI.

- **Pros**
    - Inexpensive solution, almost all PDAs are equipped with an infrared port.
    - The extra energy consumption required by the PDA is very low.
    - Location accuracy is high (since line of sight is required and the emitting direction is controllable).
    - Except location, direction information can also be inferred (since line of sight is required).
    - Easy to maintain.
- **Cons**
    - It requires line of site, so it is possible that some visitors may be blocking the signal (this can be overcome by placing the beacons on the ceiling and/or using multiple beacons). Furthermore the user has to point the IR port to the beacon and be careful not to block it with his/her fingers.
    - The beacons require an energy source; i.e., a battery or a power cable. If batteries are used then several practical issues may be arise, e.g., how and when these should be replaced and by whom, and there is also an associated cost, which – depending on the number of the individual triggers – may be considerable. On the other hand, power cables require making an electrical installation at each information item, a fact that may not always be possible – especially in archeological sites and museums.

- When there are beacons with overlapping emission ranges the PDA may receive conflicting signals. In this case, some heuristic conflict resolution strategies will need to be devised.
- High purchasing cost of the infrared transmitters/beacons.

b.2. Bluetooth: Bluetooth access points are placed at each information item. Each access point emits a unique ID. The PDA reads the emitted signal through a Bluetooth receiver and accordingly provides a presentation of the related information item.

- **Pros**
  - Inexpensive solution, many PDAs are equipped with Bluetooth.
  - The extra energy consumption required by the PDA is low.
  - Easy to maintain.
  - It does not require line of site, so it works in crowded situations.
- **Cons**
  - Location accuracy is low (it can also transmit through walls giving inaccurate location).
  - A single access point can be concurrently used by a maximum of 8 users. This means that in order to scale the system up multiple access points are required.
  - The access points require an energy source; i.e., a battery or a power cable. If batteries are used then several practical issues may be arise, e.g., how and when these should be replaced and by whom, and there is also an associated cost, which – depending on the number of the individual triggers – may be considerable. On the other hand, power cables require making an electrical installation at each information item, a fact that may not always be possible – especially in archeological sites and museums.

- When there are access points with overlapping emission ranges the PDA may receive conflicting signals. In this case, some heuristic conflict resolution strategies will need to be devised.
- The process of "discovering" a Bluetooth signal may take up to 10 seconds.

b.3. Radio-Frequency Identification (RFID): RFID passive tags are placed at each PoI. The PDA is equipped with an RFID reader. The reader has an antenna that emits radio waves. The RFID tag's antenna picks up this signal and returns it, augmented with a unique ID number.

- **Pros**
  - RFID tags can be quite small, do not require an energy source and are inexpensive.
  - Easy to maintain.
  - The extra energy consumption required by the PDA is low.
  - Short-range RFID readers for the PDA can be found at an acceptable cost.
  - If very short-range tags are used with no overlapping areas, location accuracy can be high.
- **Cons**
  - It requires that the visitor's PDA approaches to a very short distance from the tag (5-10 cm).
  - Visitors will have to learn to discover the RFID tags.
  - Does not work well in crowded situations.
  - When there are triggers with overlapping emission ranges the PDA may receive conflicting signals. In this case, some heuristic conflict resolution strategies will need to be devised.

***b.4.* Wi-Fi (802.11):** This approach uses wireless network access points in order to pinpoint user position. More specifically, as the user moves around the museum his/her PDA receives the signals of the surrounding APs through its wireless network card. These signals are received from multiple access points at a variable strength. The signal strength information of each access point is used by the system to triangulate the location of the user.

- **Pros**
  - Nowadays, practically all PDAs are equipped with a wireless network access card.
  - It is easy and rather inexpensive to scale up the whole system.
  - It does not require any interference with the exhibits such as electrical installation or tags.
  - The addition of new exhibits does not require the purchase or installation of any equipment.
  - Can be used both indoors and outdoors.
- **Cons**
  - Accuracy is medium.
  - Interference with the exhibits concerning the signal deformity because of possible signal interjections.
  - The extra energy consumption required by the PDA is medium.
  - An elaborate and time-consuming procedure must be followed in order to calibrate the system.
  - Significant purchase cost of the network infrastructure (e.g., wireless access points).

b.5. Global Positioning System (GPS): The PDA is equipped with a GPS receiver that determines the user's location (also speed and direction) in within a few meters using time signals transmitted along a line of sight by radio from satellites orbiting the earth.

- **Pros**
  - Inexpensive, many PDAs have a built-in GPS receiver.

  - Easy to implement and maintain.
  - Practically no maintenance cost.
- **Cons**
  - It does not work in indoor environments and in general in any areas that have no direct contact with the open sky (e.g., are covered with any type of material).
  - In order to achieve the best possible accuracy an extended antenna is required.
  - Due to a feature called Selective Availability (SA) that is used for restraining publicly available GPS data to be used for guiding long range missiles to precise targets, intentional errors maybe introduced to the signal up to about 10 meters horizontally and 30 meters vertically.
  - Startup delay is sometimes more than 2 minutes.

The following table (***Table 1***) illustrates the advantages and disadvantages of all aforementioned location-sensing technologies.

	RFID (short)	**RFID (long)**	**WLan**	**GPS**	**Infrared**	**Bluetooth**
Purchase Cost	Low	Medium	Medium	Low	Medium	Low
Line of sight	No	No	No	Yes	Yes	No
Accuracy	5-25 cm	3-5 m	3-5 m	~5 m	High	Medium
Functions indoors	Yes	Yes	Yes	No	Yes	Yes
Energy Required on PDA	Low	Medium	Medium	Low	Low	Low

Table 1: Comparison of location-sensing technologies

3. Architecture

The architecture of the Mobile Navigator is a planning and control operation in which a collection of indepe ndent modules collectively determines all necessary functionality and performance needed. It consists of a group of components communicating with a centralized main component known as "arbiter", either by sending votes in favor of actions that satisfy its objectives, or by indicating the utility of various possible world states.

The "arbiter" is responsible for combining the components' votes and generating actions which reflects their objectives and priorities. Thus, the Mobile Navigator provides coherent, rational, goal-directed behavior while preserving real-time responsiveness to the user.

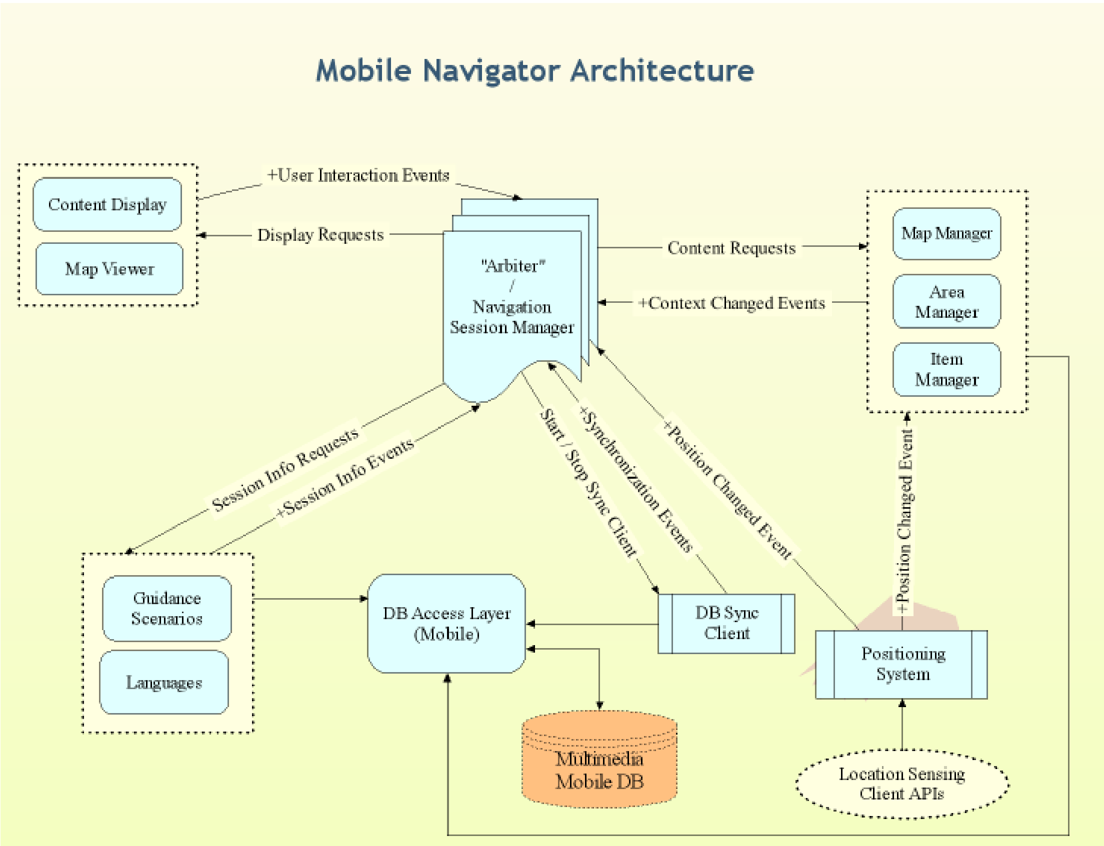


Figure 19: Mobile Navigator Architecture

The architecture consists of major components, such as the "*arbiter*", being responsible for all navigation processes. Furthermore, the *Map Manager,* the *Area Manager* and the *Item Manager* constitute the *Data store Module.* In case of context change, the "*arbiter*" is informed due to position information given by the *Positioning System*. Moreover, the *Data store Module* is the mediator between the "arbiter" and the Multimedia Mobile Database in order to get context data grouped and packed for further actions, such as *content or map representation*. The Multimedia Mobile Database is accessible through the *Database Access Layer (Mobile)* to all components that are related with data retrieval, such as *Languages*, *Guidance Scenarios*, *DB Sync Client,* and, as mentioned before, to *Data store Module*. More information is presented in section "Components".

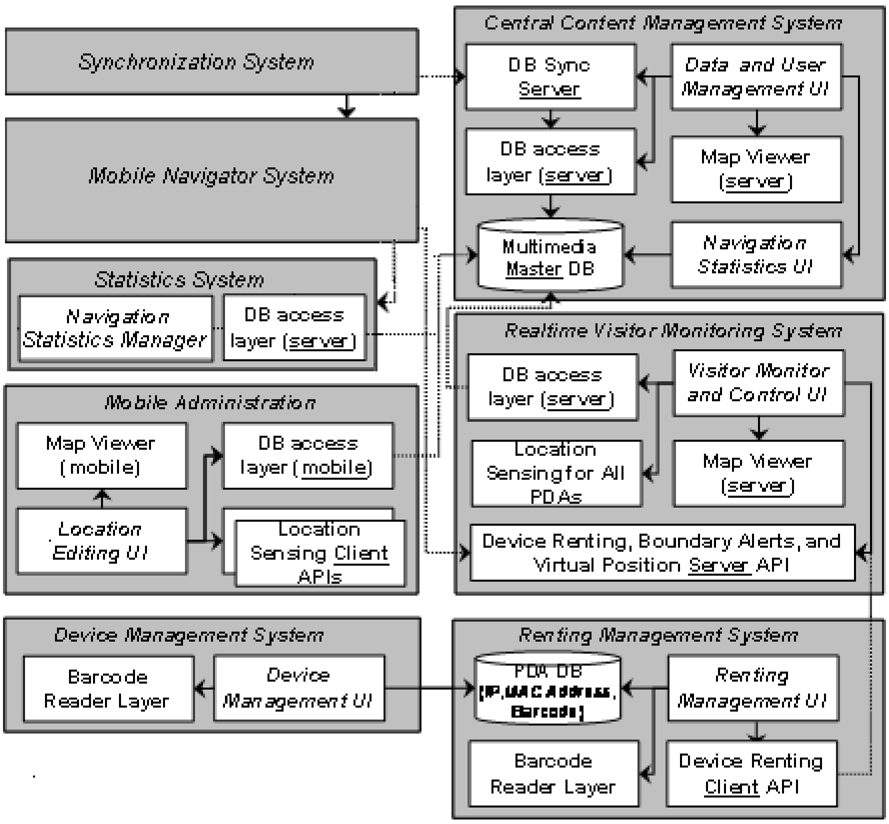


Figure 20: Overall platform architecture

Important sub-systems are not incorporated within the architectural diagram, but they are part of the whole integrated platform (see Figure 9):

- **Central Content Management System** [27] has as a key feature that content administration is facilitated through spatial data editing, providing a direct-manipulation graphical editor to administer the semantic content and the geometrical data of information areas and information items directly over the displayed maps.

- **Mobile Administration System** provides novel features for location data editing, except from the traditional non-mobile content management and form-based data entry.

- **Device and Renting Management System.** Existing location-aware mobile information systems do not deal with the issue of explicit device renting and returning processes. The Device and Renting Management System is applicable both to protected and commercial settings requiring quick service, as for example museums, galleries and exhibitions, by offering very efficient device management and renting processes.

- **Real-time Visitor Monitoring System** [28] enables the real-time monitoring of users/visitors in their surroundings and the transfer of information between the users' handheld devices and other applications that perform location-sensing and activation, in the context of a mobile, location-aware information system. It handles hundreds of devices with efficiency and robustness, delivering the necessary information in time and providing an accurate real-time overview of the system. It provides knowledge of the location of any activated device and its functioning state, as well as management of device state and location for testing or security purposes.

- **Synchronization System**. On-demand synchronization of the master database to the mobile Database (Mobile DB) on the PDAs (i.e. DB Sync Client/Server). While one expects this facility to be offered by the Database Management System, it was implemented from scratch as the mobile edition of the Microsoft SQL Server restricts the size of the mobile Database to at most 100 MB. (see section 4.7 Data Synchronization)

- **Statistics System** (e.g., time spent at an information item, information items reviewed, elapsed time of use) supporting various queries (e.g., most popular information item in a selected period, total time of use for all users/visitors, average exploration time for users/visitors). In order to achieve this, the Mobile Multimedia Navigator records information about the visited information items as well as the time that the user spent on each of them. All the aforementioned data are collected for each individual session during the navigation process. After the end of each navigation session, the collected data are sent to the Statistic System for storage and further analysis.

4. Content User Interface

4.1 Design Requirements

This section presents the outcomes of the preliminary design activities of the Mobile Navigator that took place during the phase of planning the environmental exhibition in the physical context.

4.2 Task Analysis

In order to design the Mobile Navigator, an analysis of the basic tasks that the users of the hand-held device should be able to perform has been made. These tasks are presented at a high level in ***Figure 21***.

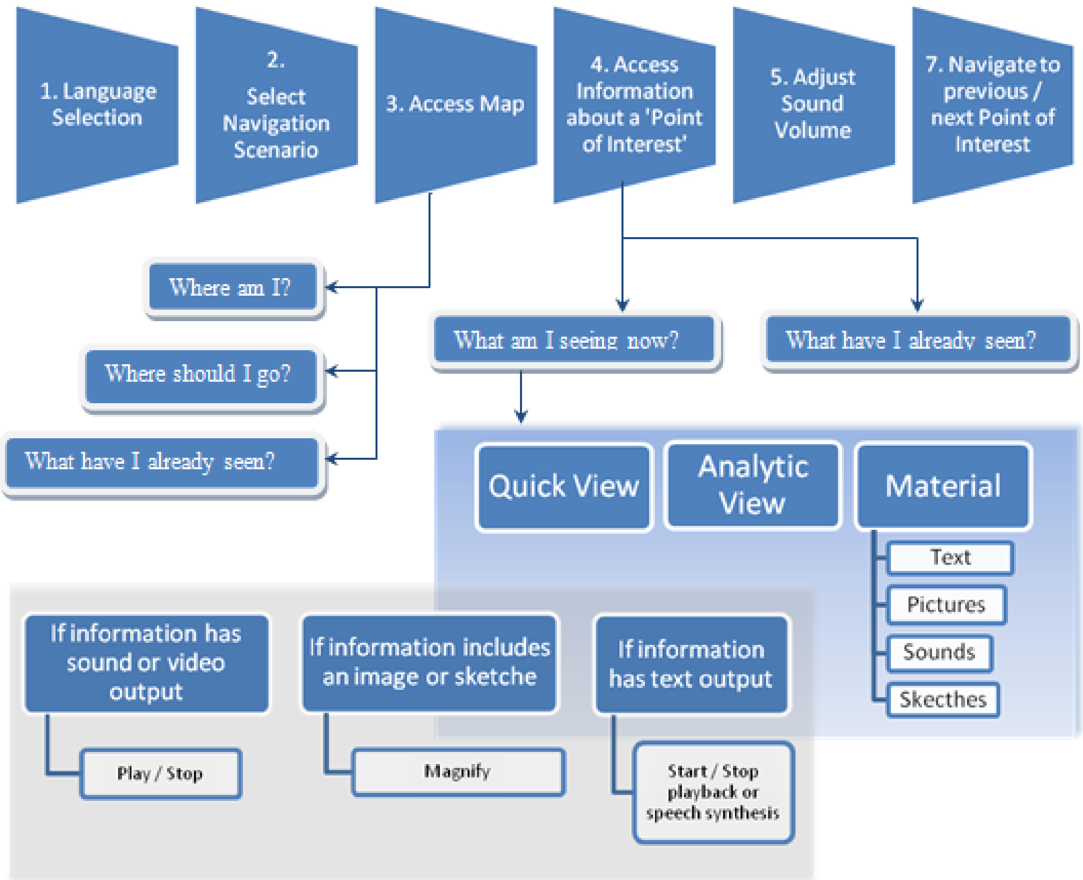


Figure 21: Mobile Navigator high-level task analysis

4.3 User Interface Master Design Layout

Following the design principle of *consistency*, all the different screens comprising the user interface of the Mobile Navigator adopt the same abstract design layout. This "master" design layout is illustrated in ***Figure 22***.

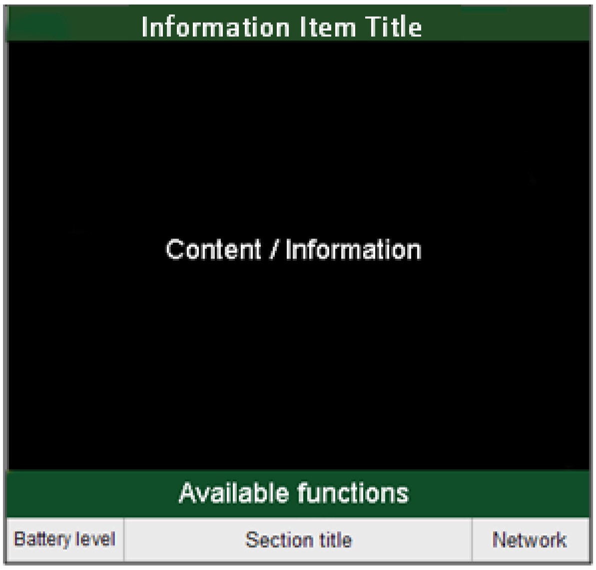


Figure 22: User Interface Master Design Layout

The master design layout comprises the following parts:

Information item title*:* A descriptive designation for the information item for which information is currently presented on the screen (*if no information item is currently presented on the screen, e.g., the user is looking at the map, this part does not appear*).

Content/Information: This is the main part of the interface where multimedia content is rendered, depending on the current application context.

Available functions: All the functions that the user can currently execute. To improve the ease of use and learning of the Mobile Navigator, these options are kept to a minimum and at no point will be more than five.

Section title: A brief description of the user interface section that the user is currently interacting with (e.g., Language Selection, Available Scenarios, Map, Photos). In case the current section comprises several pages (e.g., a long description of a specific information item), then the number of the current pages and total number of pages is appended to the title (e.g., Description 3/6).

Battery level: The current battery level, so that both the staff and the visitors know when the device needs to be recharged.

Network: The wireless communication status.

4.3.1 Design Principles

The design of the Mobile Navigator's user interface is based on the following principles:

User-centred design: Collection and analysis of the requirements, needs and preferences of the end-users are the main goals of the design process as well as evaluation and testing with representative end-users.

Simplicity: The user interface does not require that the visitors/users are familiar with the use of hand-held computers (PDAs). The functional choices provided at each point are the absolutely necessary ones. The whole interaction process is designed in such a way that the need for user input at any point is minimized. The user can even fully explore all the available information by using just a single button or (in "hands-free mode") without having to do anything (i.e., fully automated information presentation).

Consistency: All the different screens that appear on the Mobile Navigator's user interface adopt a common master design layout scheme. Additionally, there is consistency in the way the user interacts with the device, as well as in all the symbols and colors presented in any part of the interface.

Visible controls: At any time, all the currently available interaction options are made explicit and visible to the user (i.e., there are no hidden or obscure choices or functions).

Provision of adequate feedback: Whenever there is an automatic change to the user interface (e.g., the user approaches or leaves an information item) both visual and aural feedback is provided. Feedback is also provided regarding the current battery and network communication, as well as when there is a critical change in it (e.g., battery low).

Memory and anticipation: The system keeps track and thus can "remember" which information item the visitor has already seen, as well as which part of the available information has been presented about them. This feature is used, on the one hand, to present this information to the user (e.g., so that s/he knows which information item s/he has already visited) and, on the other hand, to enable the system to "suggest" subsequent information item that could be visited according to the selected guidance scenario.

Adequacy for the environment context: During the user interface design process, the particular characteristics of the environment as the context of use have been also taken into consideration. For example, an integrated feature namely "single button interaction" gives the ability to the visitors to interact with the application using one hand (it is very common that in the context of an environment, such as a museum, one of the visitor's hand is "occupied" by a handbag, a camera, a child, etc.).

4.3.2 Interacting with the Mobile Navigator

Figure 23: Schematic representation of the Mobile Navigator hand-held device

Visitors can interact with the Mobile Navigator using the buttons residing on the device and/or its touch screen. The functions of the device's hardware buttons (***Figure 23***) are the following:

(1) Arrow buttons: Change the currently selected option (using up/down, when options are presented vertically and left/right, when options are positioned horizontally). The currently selected option is highlighted through a red rectangle (***Figure 24***).

(2) OK: Activate the current selection.

The remaining hardware buttons are used to make the following functions available from any screen of the Mobile Navigator:

(3) Volume: Increase sound volume.

(4) Volume: Decrease sound volume.

(5) Map: Access a map that also pinpoints the user's current position.

(6) Physical located information item: Visit the latest physically visited information item.

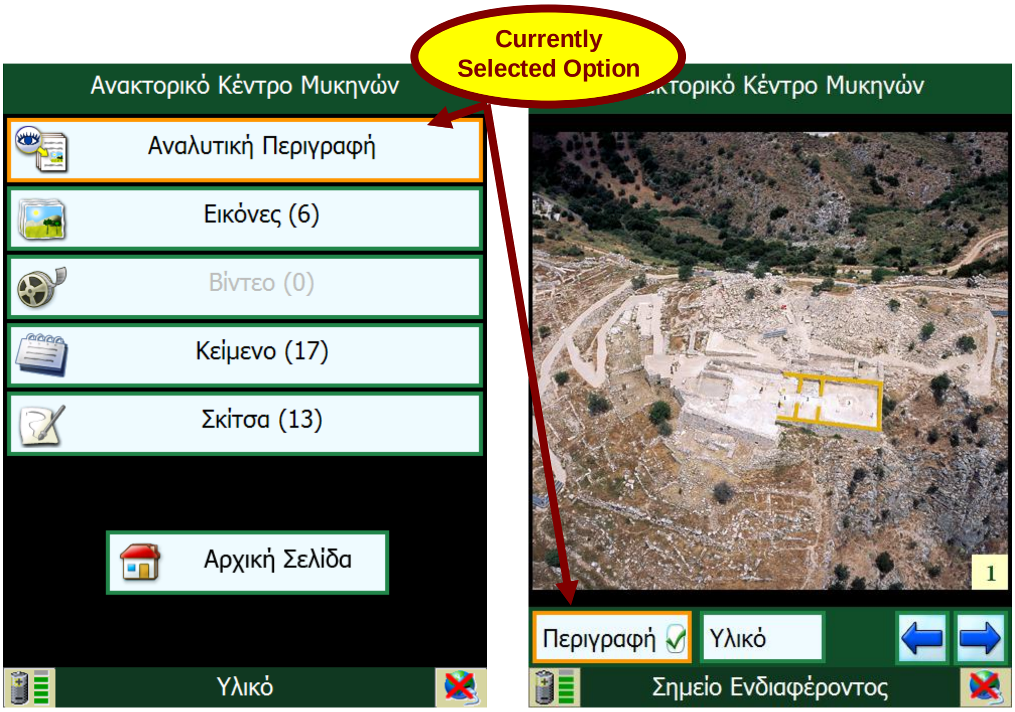


Figure 24: Highlighting the current selection

4.3.3 Single button interaction

The user interface allows the exploration of all the information that is available for any single information item page by page (e.g., short and long descriptions, additional images, videos and texts) through the use of a single hardware button (the OK button). In other words, by just pressing the OK button, the user will be able to see all the information contents of the Mobile Navigator, without missing a single bit. This can be achieved due to the "memory and anticipation" principles that were followed for the design of the user interface (see Section "4.3.1 Design Principles"). Hands-free interaction

In case of long time period user inactivity (i.e., user not interacts with the Mobile Navigator), the Mobile Navigator takes the initiative to automatically present the information to the user. Thus, a visitor has the option to passively view all the available information, without having to do any action. Of course, at any time the user can interrupt the automated presentation and take over the control of the Mobile Navigator.

4.4 Content Display

4.4.1 Information Item Presentation

Whenever a user approaches a designated information item, the Mobile Navigator plays a characteristic sound in order to notify him/her and automatically presents on its screen the information item's "home page" (see ***Figure 25***).

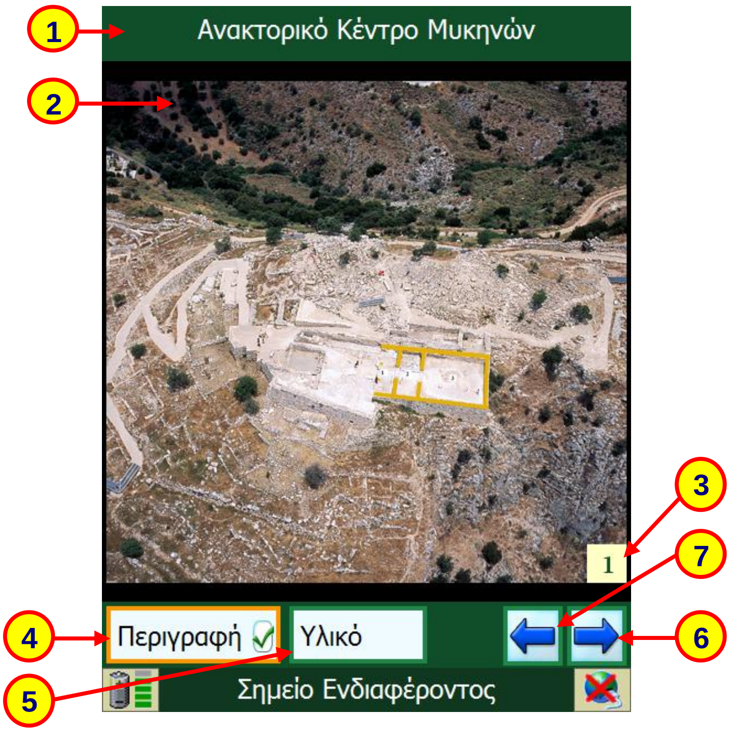


Figure 25: Sample home page for a information item

The home page of any information item comprises the following parts:

(1) The title of the current information item. A representative photo, so that the user can match the presented information item with the real one.

(2) A unique identification number, which is also used to represent the information item on the museum map.

(3) Option "Description". Presents a short description of the information item which can comprise a sequel of several different screens each one containing narrated text, images or video.

(4) Option "Material". Through this option users can access additional material that is related to the current information item (see ***Figure 26***):

(5) In-depth description.

- Videos.
- Photos.
- Sketches.
- Other texts.

(6) Option "Next Information Item". With this option the user can visit the next information item according to the selected guidance scenario.

(7) Option "Previous Information Item". Through this option the user can visit the previous information item according to history or the selected guidance scenario.

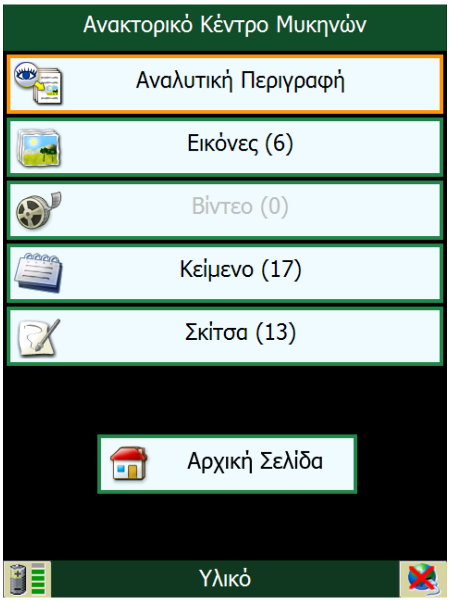


Figure 26: Accessing additional material for an information item

The number in the parenthesis next to each category indicates the number of the currently available items. In case of a category does not contain any items, it is rendered using a different visual style and the user cannot select it (disabled).

When the visitor has seen all the available information of a specific category, then a green mark appears next to the item (see ***Figure 26***).

4.4.2 Information Item Description

The description of an information item consists of a sequence of pages, each of which can contain:

(a) Text

(b) Image/Sketch

(c) Video

The first button of each page's toolbar is used for "returning to the home page". The last is for "going to the next page". Before the latter, there is another button for "going to the previous page". In case that there is no next or previous page, the corresponding button is invisible.

The user interface for each category page is presented below.

4.4.2.1 Text Page

When a text page is presented the text is automatically read. If the presented text does not fit in the available screen space (a situation that is usually avoided by design – but this is not always possible), the user can scroll the text up and down using the corresponding arrow buttons. The toolbar contains a button for playing/pausing speech.

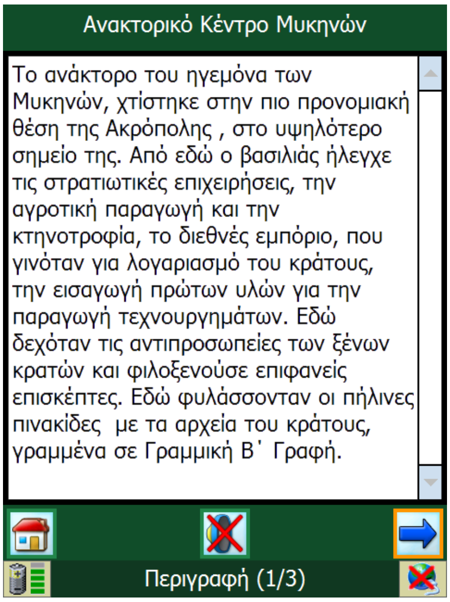


Figure 27: Text Page

4.4.2.2 Image/Sketch Page

The Image/Sketch page consist of the image and a label which is announced automatically. The user can magnify the images so that they take up the whole available screen space (***Figure 28***).

Figure 28: Image/Sketch Page

4.4.2.3 Video Page

The user can pause/play or stop the current video, as well as magnify it in order to take up the whole available screen space. In this case, if any button is pressed the video playback returns to its original size. Since, in the context of the Mobile Navigator, videos are expected to be quite short in length, there is no need for fast-forward and fast-backward buttons, thus simplifying the available interaction options (***Figure 29***).

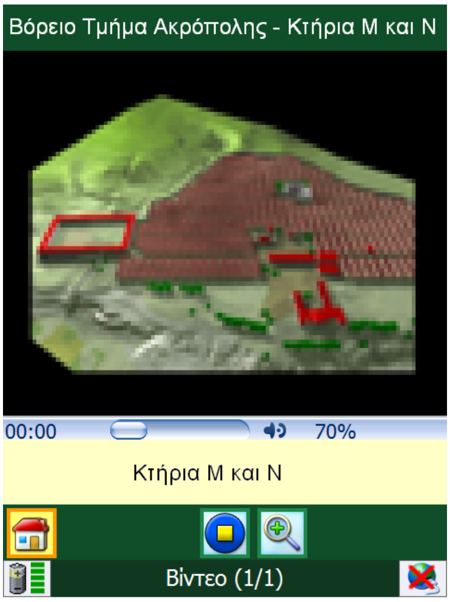


Figure 29: Video Page

4.5 Components

This section examines the structure and role of all the developed Mobile Navigators' components, as well as the interaction among them. Additionally, implementation issues are presented along with related code examples.

4.5.1 Languages

Because of the expected cultural user diversity, a multi-lingual user interface was developed. This was facilitated through the implementation and integration of a

"Language Manager" software module. This module pre-loads all the dictionaries available in the hand-held device. It is important to note that the whole platform supports the addition of any new language by creating a new dictionary through the Central Content Management System. The new dictionary will be available after the synchronization process to all running instances of the Mobile Navigator. The Language Manager is based on a dictionary, as presented in Figure 10. It comprises a simple text file which includes: (i) the language identification number according to the database entry, and, (ii) the correspondences between the phrase in English and the translated phrase in the respective dictionary language (see ***Figure 30***).

```
<phrase in English> = <phrase to each time dictionary language>.
```

```
Language ID = 2
Material = Υλικό
Language Selection = Επιλογή γλώσσας
Available Languages = Διαθέσιμες Γλώσσες
Detailed Description = Αναλυτική Περιγραφή
Navigation Strategy = Σενάριο Ξενάγησης
Available Navigation Schemes = Διαθέσιμα Σενάρια Ξενάγησης
Information Item = Σημείο Ενδιαφέροντος
Description = Περιγραφή
Database Synchronization = Συγχρονισμός με Βάση Δεδομένων
```

Figure 30: Language dictionary template: the case of the Greek dictionary

The Language Manager provides its functionality through a set of public methods that are always accessible by any component of the Mobile Navigator. The Language Manager class diagram is provided in C# in ***Figure 31***. It should be noted that the Language Manager is used only for translating the menus of the user interface and not for translating the information retrieved from the database (i.e., the content).

Figure 31: Language Manager Class Diagram

4.5.2 Data Store Module

The role of the Data Store Module is a) data retrieval and, b) to check for context changes. The Data Store Module consists of three independent components: (a) Map Manager, (b) Area Manager, and (c) Item Manager. All these components, except for the Item Manager, monitor and use the information provided by the Positioning System.

4.5.2.1 Map Manager

The Map Manager provides all the appropriate data and information about Maps, retrieving the latter through the Database Access Layer (DB Access Layer). When the position information refers to a different map than the current one, the Map Manager sends: (a) a notification to the Area Manager in order to update the list of the currently available areas of interest and, (b) a notification to the "arbiter" module in order to act accordingly (e.g., by displaying the appropriate map).

4.5.2.2 Area Manager

The Area Manager keeps a list of the available areas of interest of each map. When a map change event occurs, the renewal of the list is triggered by the Map Manager. This update step is necessary since different maps contain different information areas (see ***Figure 1***).

4.5.2.3 Item Manager

Any information related to an information item is provided by the Item Manager. The latter retrieves any content requested by the "arbiter" through the DataBase Access Layer (DB Access Layer). As already mentioned, information has to be grouped and packed according to the current information presentation requirements. The Item Manager meets these requirements by providing the data through a set of public methods (***Figure 32***). Moreover, the Item Manager takes into account the currently selected language. This is easily achieved by adding the language id as a parameter to the retrieval query (provided by the Language Manager).

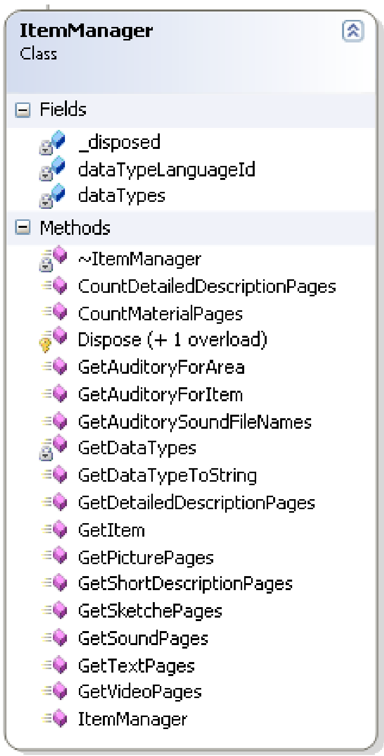


Figure 32: Item Manager Class Diagram

4.5.3 Navigation Sessions

The "arbiter" is the main system architecture component that handles the navigation process. It provides functions for controlling the navigation session (e.g., start, stop). These functions are triggered by the Device and Renting Management System (see Architecture). Before a new navigation session starts, the desirable language and the navigation mode are set by the Renting Management System.

The "arbiter" handles any context changes which are triggered by the data store module. Additionally, it retrieves the appropriate data and prepares the representation of the information item. Generally, the role of the "arbiter" is to identify events, process them, and finally perform appropriate actions to satisfy the requestors (e.g., update the user location on the viewed map after a "position change" event).

During the navigation process, there is continuous communication between the Mobile Navigator and the Realtime Visitor Monitoring System (see Architecture). The role of the Realtime Visitor Monitoring System is to keep track of the location (provided by any location-sensing API) of each hand-held device. In the case of boundaries violation (e.g., a visitor does not return the hand-held device when leaving an archeological site), an alert is triggered informing both the system administrator and the user. At the same time, a warning message appears on the screen of the hand-held device (***Figure 33***).

Figure 33: Warning message during boundaries violation

The navigation process starts when the user selects between the two available types of navigation sessions:

- Multimedia
- Auditory

4.5.3.1 Multimedia

In this case, the user has access to all the available functionality of the Mobile Navigator, including: (a) the positioning map and, (b) full access to any multimedia data of the current information item (see *Information Item Presentation*).

4.5.3.2 Auditory

In the auditory mode the map is continuously displayed. The map depicts the user's position. As a result, the process of access in any location-relative information item is simplified and automated.

The map comprises the following parts:

1) The yellow X represents the location of the user.
2) The yellow arrow shows the direction to the next information item according to the selected guidance scenario. On the other hand, the user is free to follow or not the prompts that Mobile Navigator provides. However, the chance is given to the user to return at anytime to the predetermined guidance tour.
3) The red circles refer to information items that belong to the guidance scenario.
4) Green circles refer to information items from the guidance scenario that the user has already visited.
5) Blue circles refer to information items out of the guidance scenario.
6) Map provides navigation info about the order of information items that are part of the guidance scenario, etc.
7) The section title contains the zooming factor percentage that follows the word “Map”.

Toolbar buttons (left to right):

8) The user can hear the description of the previous information item among them that belong to each visited information area.
9) The user can hear the description of the next information item among them that belong to each visited information area.
10) Stop or restart the playback of the description.
11) Performing a sequential transition among the available display options in a circular manner.
12) Map movement using the arrows (up/down/left/right)
13) Zoom reset. The zooming level is adjusted to a pre-defined zooming factor, according to information relative to each map.
14) Zoom out. Progressively zooming out with automatic focus in the center of each map.
15) Zoom in. In case of user is in the visible area, the progressive zoom in, automatically focuses on him. Otherwise, the center of the map is focused.

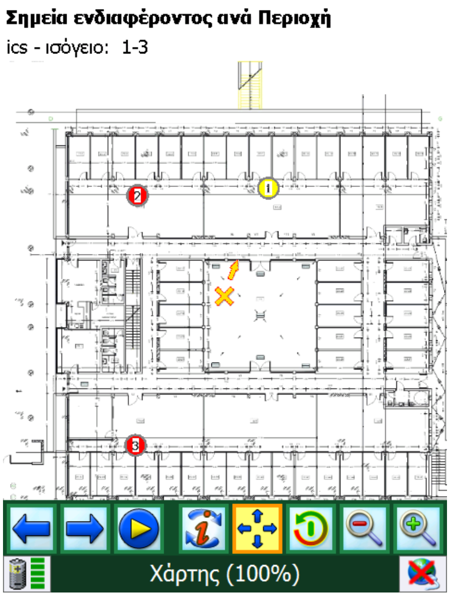


Figure 34: Map in auditory application mode

Whenever the user approaches an information item, the Mobile Navigator plays a characteristic sound in order to notify him/her. Then, the user will hear the title and the description of the visited information item.

If there are more information items that exist in the same information area the Mobile Navigator announces the following phrase: "In this information area, there are N information items", where N is the number of these. Afterwards, the user successively hears the title and description of each information item. This is described with the following expression:

```
Sound Collection:
(Information area with more than one information items) ⇔
Count_X_Items {
      item1 {title, description},
                  item2 {title, description},
                   …,
                  itemn {title, description}
         }
```

4.5.4 Application Modes

Application modes are the states that the Mobile Navigator could transits through its life cycle. There are five modes that are described below.

4.5.4.1 Suspended

In this mode (***Figure 35***), the Mobile Navigator is not available for use, because in most of the cases the hand-held device is running out of battery. In the Suspended state, the Mobile Navigator consumes the minimum energy in order to promptly charge the battery. Switching from this state to another one can be done manually. There is one state that the Mobile Navigator can transit to, namely Active.

Figure 35: Mobile Navigator in Suspended Mode

4.5.4.2 Active

This mode indicates that the Mobile Navigator is ready to be rented and used. In this state, the Wi-Fi is activated because of the following reasons: a) as mentioned, the Renting Management System (which is a standalone component) will trigger the “arbiter” in order to a new navigation session to begin. However,

b) a network communication must be established between the Visitor Monitoring System and the Mobile Navigator in order to start the monitoring management process.

Based on the above state, there are two possible transitions. The first is manually shifted, namely Suspended, while the second is the Rented state. During the transition from the Active state to the Inactive, it might be possible to intermediate another one, namely Synchronizing. To perform this specific transition, new data or content modification must occur to a pre-defined synchronization server (***Figure 36***).

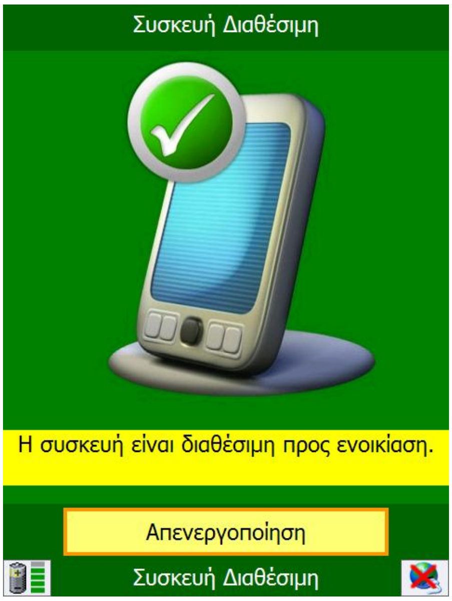


Figure 36: Mobile Navigator in Active Mode

4.5.4.3 Rented

In this mode, the Mobile Navigator is on use, and a navigation session has been started. The only state that application can automatically transit is the previous one, the Active described above. The transition takes place when a user finishes the navigation. When the latter happens, the Renting Management System triggers the “arbiter” to stop the navigation session by sending to the statistics server all information collected about navigation (e.g., visited information items).

4.5.4.4 Synchronizing

As mentioned, during the transition from the Active state to the Suspended, the Synchronizing state might take place. If necessary, the content is updated by downloading it through a predefined ftp server. This occurs when new images, videos or other type of contents are added to the database. It also occurs when data have been modified by the content administrator. During the synchronization process the Mobile Navigator is useless. After the completion of the synchronization process, the Mobile Navigator automatically shifts to the next state, namely Suspended (***Figure 37***).

Figure 37: Mobile Navigator on Synchronizing Mode

The following figure (Figure 38) shows the application life cycle through a transition diagram according to the aforementioned information.

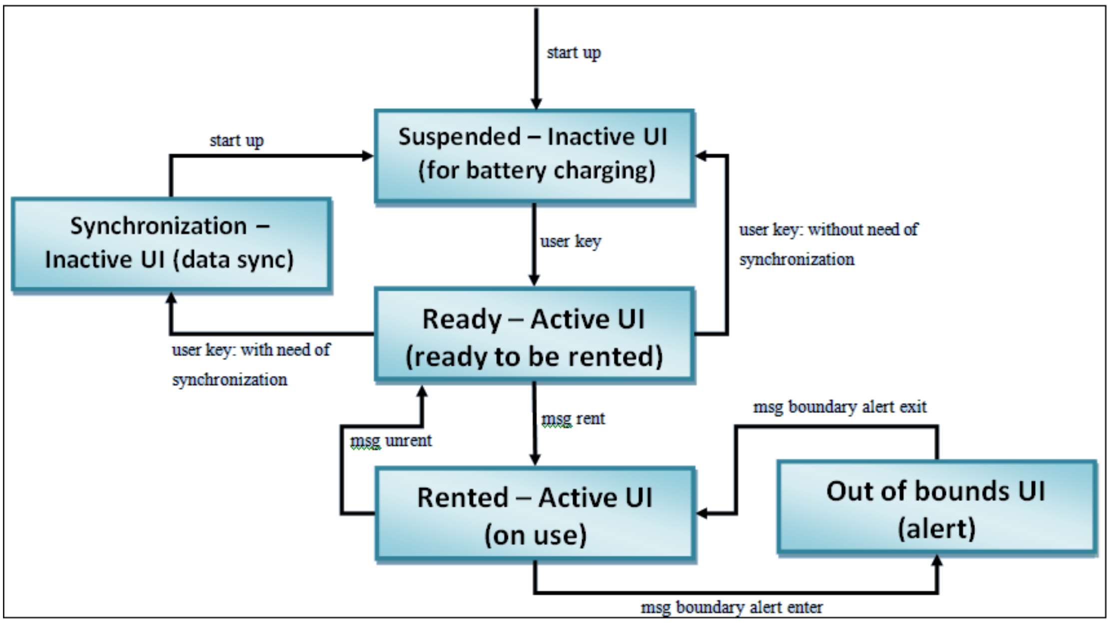


Figure 38: Application Modes

4.6 Data Infrastructure

Scalability and extensibility issues are the goal or in other words, a portion of it is very much depended on the underlying data infrastructure. This has as result a centralized database that contains the schema of information and is filled with data.

The database schema (***Figure 39***) was designed in a generic manner, aimed at storing information for the integral questions according to the fact that a context-aware pervasive system is supposed to have an answer for the following questions:

1. What are users doing? (Logging of user's activities)
2. Where am I? (Current location and current map)
3. What can I find near me? (Spatial information for proximate artifacts)
4. What can I learn about proximate items? (Data content)

As shown in ***Figure 39***, there are different tables for storing the actual data and different tables for storing the multilingual data description

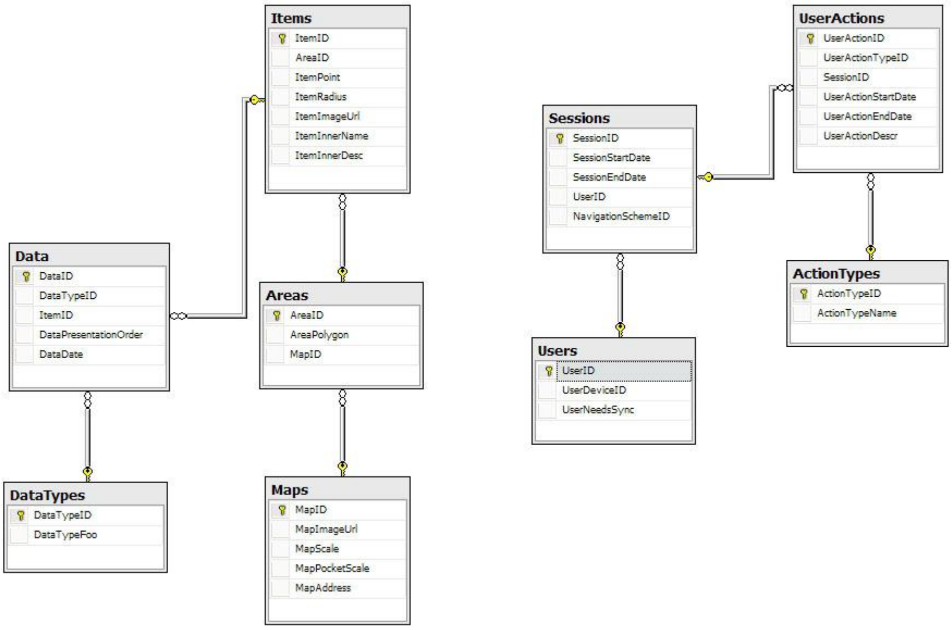


Figure 39: Part of the Database Diagram

4.7 Data Synchronization

The goal of this mechanism was to create a low-bandwidth cost solution for the replication of data between a centralized database and distributed mobile clients.

The reasons for employing such an elaborate scheme are:

- The distributed PDA clients are not always online since they move through a physical space in which there is no guarantee that each point is covered by an access point.
- There is a large number of mobile clients in the space (e.g., the Acropolis scenario has 1000 PDAs). This means that alternative content provisioning solutions such as streaming are not appropriate since they demand a large amount of bandwidth and create network congestion bottlenecks.
- As off-the-shelf pocket databases pose 100MB limit in content size, we cannot store multimedia content into blob database fields because this would definitely overflow the pocket database limit.

In order to deal with these challenges, a mechanism was developed that transforms the database synchronization problem into a well-defined file synchronization problem. This is achieved by using the database build-in synchronization facilities to take a snapshot of a central database into a pocket database file (.sdf format).

The multimedia files are not stored inside the database because of the limited capacity of the mobile database snapshot. The multimedia files (i.e. videos, images, etc) are marked with a file descriptor (URI) by the side of the database. They actually stored in public synchronization ftp server that the mobile devices can access during synchronization process. Additionally, the database mobile snapshot is uploaded into that synchronization ftp server.

In the synchronization ftp folder, there is an index xml file describing every transaction made to that folder according to the addition, deletion or update of each file along with the date it occurred (***Figure 40***).

```xml
<?xml version="1.0" standalone="yes" ?>
<DocumentElement>
<Remote>
  <Date>2008-10-28T18:45:19+02:00</Date>
  <FileName>1203F01.jpg</FileName>
</Remote>
<Remote>
  <Date>2008-10-28T18:45:19+02:00</Date>
  <FileName>1203F02.jpg</FileName>
</Remote>
```

Figure 40: The XML schema used for data synchronization

The multimedia content management and pocket database management is handled by the Central Content Management System which is described in more detail as above in chapter 3-Architecture. The Central Content Management System sends a signal to the Synchronization System, shown in Figure 20, notifying it that there has been a change in the database content and that the clients need to be synchronized. As mentioned before, during the deactivation of the Mobile Multimedia Navigator, it asks the Synchronization System if there is any need of data synchronization (modification or addition) by sending a message that contains the mobile device mac address (which is unique).

If there is need for synchronization, the Mobile Multimedia Navigator gets into Synchronization mode. During this process, the local xml file (located on the mobile device) must be compared against the XML file that resides on the synchronization FTP server side to decide which files have been updated and thus must be downloaded including the mobile snapshot (.sdf format). After the completion of the synchronization process, a cleanup process starts in order to delete all unnecessary files (i.e. images, videos, etc). Additionally, the Mobile Multimedia Navigator informs the Synchronization System in order to update the field in the database which indicates that the mobile device is up-to-date (***Figure 41***).

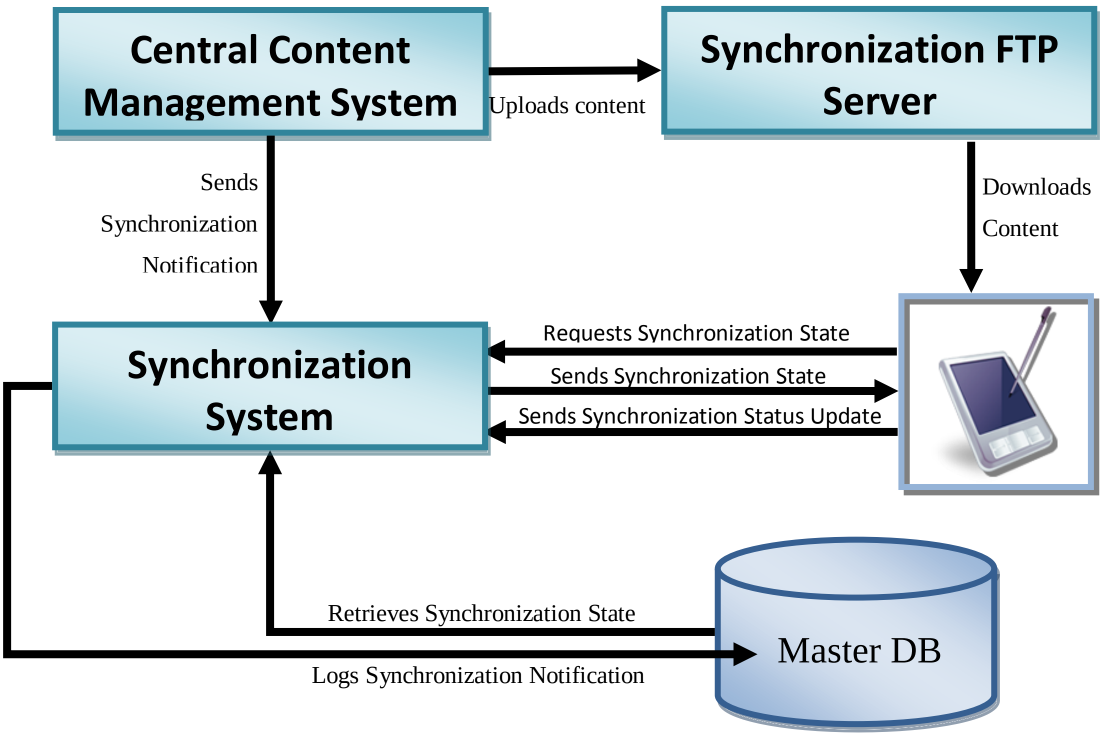


Figure 41: The Data Synchronization Mechanism

Storing the devices' MAC address along with their state (updated or outdated) helps the Synchronization System to distinguish the multiple mobile devices and update them.

4.8 System Update

Concerning the large number of mobile clients in space (e.g., the Acropolis scenario has 1000 PDAs), a semi-automatic mechanism was developed in order to automatically upgrade all instances of the Mobile Multimedia.

Specifically, if a new version of the Mobile Multimedia Navigator is released, it is added to the specific synchronization ftp folder. The Synchronization System adds a new entry to the index xml file so that mobile clients download it locally to their storage space. A daemon (eGuideLoader.exe) was developed to run on each mobile device in order to identify if a new version of the application was delivered after the synchronization process. In that case, the daemon replaces the old version with the newer one and runs it. This daemon is simply an application wrapper of the Mobile Multimedia Navigator. If something goes wrong (i.e. the Mobile Multimedia Navigator crashes) the daemon is responsible for restarting the application and providing feedback to the user/visitor about what went wrong. The process is described by the following transition diagram (***Figure 42***):

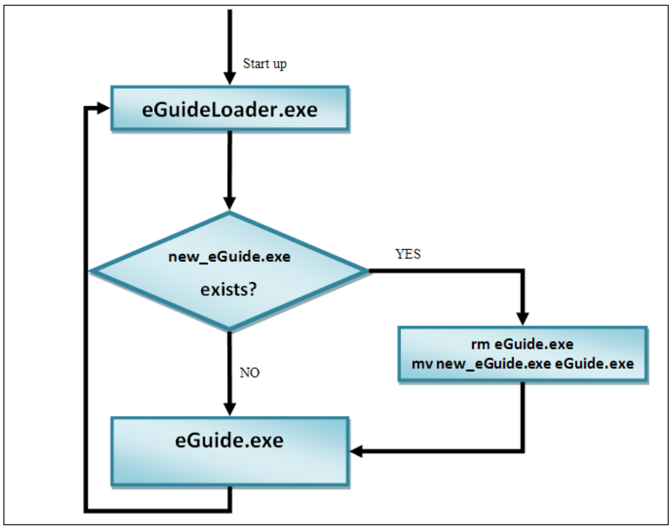


Figure 42: System update mechanism

4.9 User Interface Elements

In order to satisfy the design requirements, such as "single button interaction" and highlighting of each choice, a set of custom widgets was developed. The used software technology namely Microsoft .NET Compact Framework [29], with which the Mobile Navigator was developed, does not cover all the necessary programming needs, especially in the area of the user interface.

4.9.1 Custom Widgets

The widgets are related to the common buttons, making more convenient the interaction between the user and the interface of the Mobile Navigator. A hierarchy class diagram, provided in ***Figure 43***, presents the different types of widgets and the relation among them.

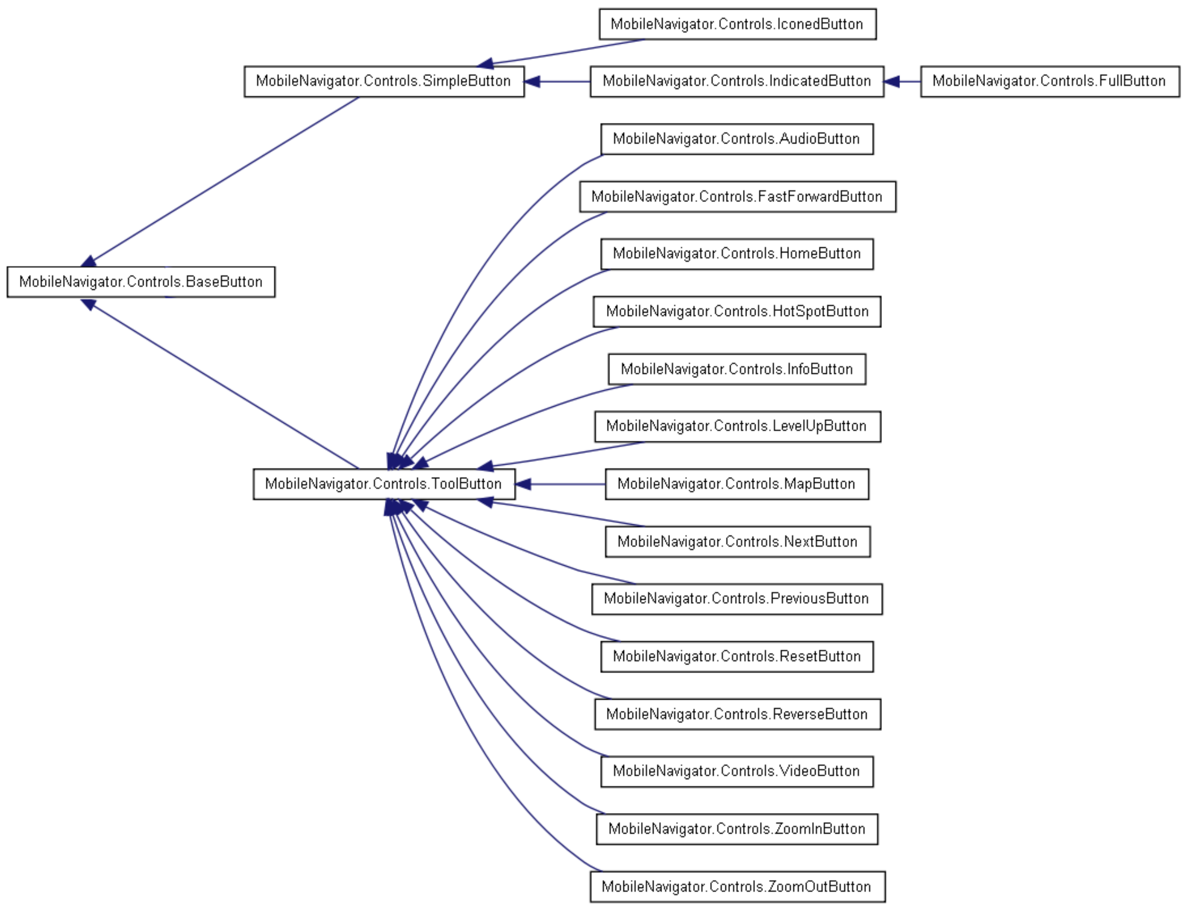


Figure 43: Widgets Controls Class Diagram

The user interface menu is constructed from widgets. In detail, buttons in the toolbar have a pre-defined specific size showing a single icon that represents the function triggered by the user. There are various states that a tool button can transit. According to the current state, the triggered action as well as the displayed icon can differ from the next one (***Figure 44***).

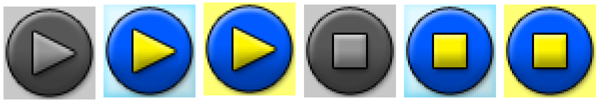


Figure 44: Widgets: Different states of a tool button

Additionally, in other user menu cases, such as the menu of "Accessing additional material for an information item" (see ***Figure 25),*** the button must provide a proportional icon as well as an indicator (except for the appropriated text label which describes the triggered function). The indicator is shown at the left of the button, meaning that this option has been visited before. For instance, when the visitor has seen all the available information of a specific category, then a green mark appears next to the text label (see ***Figure 26***).

4.9.2 Content Pages

Considering the variety and/or the similarity of content views, a framework of pages was developed to fill the appeared programming requirements. Each type of page inherits the base characteristics of the user interface master design layout as shown in ***Figure 22***. Each page consists of specific components (e.g., Custom Widgets), such as a Text Box in case of Text Page or Picture Box in case of Image Page. Furthermore, Video Page assigns the video playback to the embedded Microsoft Windows Media Player using runtime interop services (e.g., ActiveX). Thus, Media Player reproduces all sounds including these in auditory mode as well as the label or title of an information item.

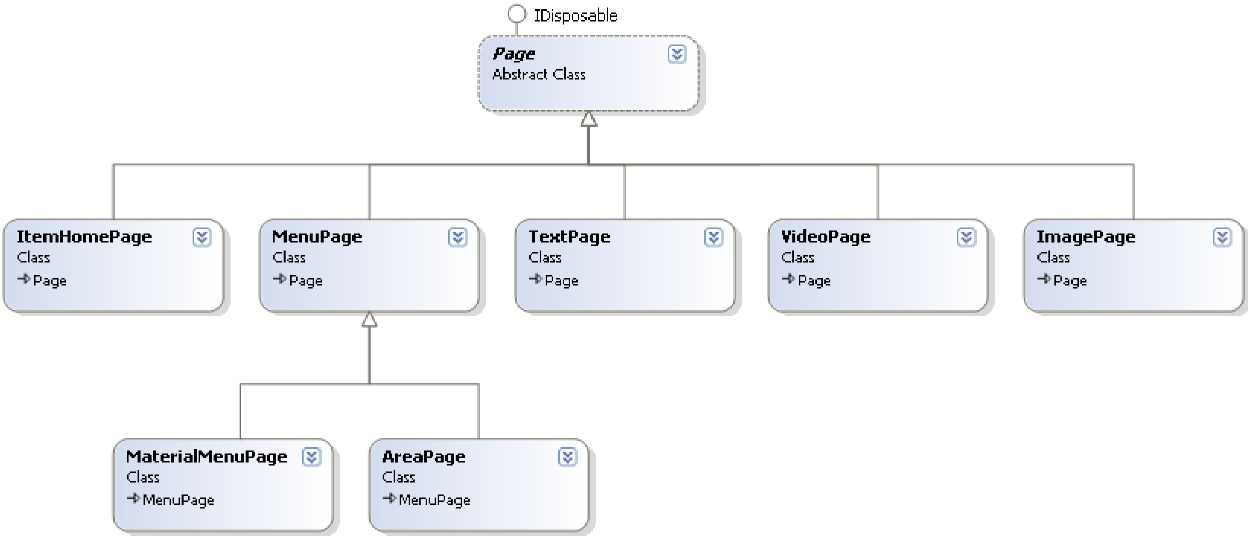


Figure 45: Content Pages Class Diagram

4.10 Virtual Navigation

During the requirements analysis phase it came out that whether using a personal guide or navigating with a leaflet, visitors prefer to deviate from predefined paths to visit alternative proximate or distant information items. Thus, the Mobile Navigator had to guide users across the scenario, allowing them to freely deviate from it, effectively enabling them to keep track of their overall path. Additionally, experienced guides reported that during tour sessions, many visitors use leaflets or books to review information details for previously visited items, and in some cases even for items to be met later during the tour, since in many situations the exposed items are directly related to each other. This had as a result an alternative insight on location-aware information delivery: besides automatic location-based information provision, users are enabled to freely review all information items, whether visited or not. In this case, the latter is possible either via previous/next navigation buttons (see ***Figure 25***), or by pointing directly with the pen over items on the displayed map or via the tool "Hot Spot Selection" offered by the toolbar of the displayed map (see 5.2.2).

4.11 Guidance Scenarios

The whole platform supports the authoring of arbitrary guidance scenarios, i.e. different types of tours that users may take, edited as sequences of selected information items associated with a short descriptive title. During use, the Mobile Navigator explicitly marks all the initially selected information items, displaying also their relative order within the guidance scenario.

4.11.1 Initial List of Items

In detail, when the user selects the desirable guidance scenario, a list of information areas are created by the corresponding Mobile Navigator's component, including one or more information items. These information areas may belong to different maps under the condition that the guidance scenario does not bring up an already existing map (this is a pre-condition that stands for Central Content Management System). A list expression might be as follows:

```
Guidance scenario list ⇔
                {
                     map_i <
                            area_j {item_1, item_2,…, item_k},
                            area_j+1 {item_1, item_2,…, item_k},
                            …,
                            area_j+m{item_1, item_2,…, item_k}
                        >,
                     map_i+1 <…>,
                     …,
                     map_i+n <…>
                }
```

4.11.2 Visiting Items from the Guidance Scenario

The Mobile Navigator keeps two lists while the navigation is on progress. The first one comes with the guidance scenario selection. The second keeps the history concerning the visited information items as in the first list. Consequently, when the user visits an information item that belongs to the selected guidance scenario, a new entry is being added to the history list. The visited information items are marked with a specific color on the displayed map.

4.11.3 Visiting Items out of the Guidance Scenario

The user can visit, through virtual navigation, any information item. This is achieved by pointing directly with the pen over items on the displayed map or physically visits them. When an information item does not belong to the guidance scenario, it is added to the visited list. As a result, the user can review all visited or remaining (according to the guidance scenario) information items via previous or next navigation buttons.

4.12 Power Consumption

The Mobile Navigator has various efficient methods for battery conservation including a) backlight adjustment and, b) the control of the Wi-Fi adapter.

4.12.1 Backlight Management

When the Mobile Navigator is in the Suspended mode, the backlight is adjusted to 10% of full brightness. When a map of exterior space is loaded, the brightness increases up to 100% of full brightness. In other cases, it seems that the 60% is enough to cover the display requirements.

4.12.2 Wi-Fi

The Wi-Fi adapter consumes a lot of energy according to a series of experiments [30]. Thus, whenever it is not necessary, the Wi-Fi adapter should be disabled. This occurs in a unique application mode, namely Suspended. On the one hand, the goal is to reduce the power consumption in order to extend the battery life. On the other hand, reducing the power consumption has as a result faster recharge of the battery.

4.13 Implementation

This section provides an overview about the development information as concerning the programming language, the integrated development environment used and other software dependencies.

The Mobile Multimedia Navigator as well the platform, were developed in C#. The latter (pronounced C Sharp) is a multi-paradigm programming language that encompasses functional, imperative, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft as part of the .NET. C# is one of the 44 programming languages supported by the .NET Framework's Common Language Runtime and is used extensively with Microsoft Visual Studio .NET.

More specifically, Microsoft Visual Studio .NET 2005 with Microsoft .NET Framework 2.0 as well Microsoft SQL Server 2005 Mobile Edition were used to develop the Mobile Multimedia Navigator. Some of the statistics about .Net Project are the following:

(a) number of lines = 33.476,
(b) number of code files = 229
(c) number of code-generated lines = 3.738
(d) number of user-entered blank lines = 5.118
(e) number of user-entered comments = 1.942

5. Annotated Area Maps

The map depicts the relative to the user position according to the total space in which he is located every moment. As a result, the process of access in any location-relative information item is simplified and automated.

5.1 Map Display

A special hardware button on the hand-held device is dedicated to straightforward map access. In case of user map change, the map is automatically presented. In that case, the presentation of information item automatically stops and restarts after map presentation. Whenever the map is presented, the Mobile Navigator plays a characteristic sound.

The map comprises the following parts:

1) The yellow X represents the location of the user
2) The yellow arrow shows the direction to the next information item according to the selected guidance scenario. On the other hand, the user is free to follow or not the prompts of a conducted tour that Mobile Navigator provides. However, the latter gives the chance to the user to return at anytime to the predetermined guidance tour.
3) The red circles refer to information items that belong to the guidance scenario.
4) Green circles refer to information items from the guidance scenario that the user has already visited.
5) Blue circles refer to information items out of the guidance scenario.
6) Map provides info about navigation, such as the order of information items being part of the guidance scenario.
7) The section title contains the percentage of the zooming factor that follows the word “Map”.

Toolbar buttons (left to right):

8) Return to the presentation of previously visited information item.
9) Hot spot selection. Using the arrows (up/down/left/right), the user can select and visit among the information items that appear in the visible area.
10) Performing a sequential transition among the available display options in a circular manner.
11) Map movement using the arrows (up/down/left/right)
12) Zoom reset. The zooming level is adjusted to a pre-defined zooming factor, according to information relative with each map.
13) Zoom out. Progressively zooming out with automatically focus in the center of each map.
14) Zoom in. In case of user is in the visible area, the progressive zoom in, automatically focuses on him. Otherwise, the center of the map is focused.

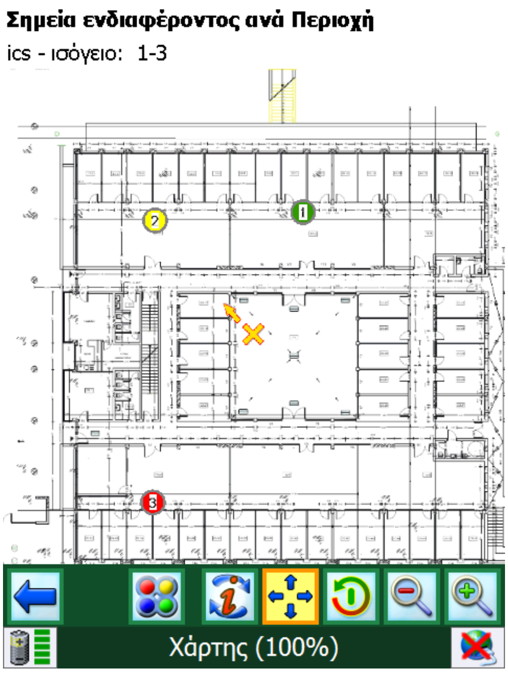


Figure 46: Map Presentation

5.2 Features

5.2.1 Real Time Smooth Zooming

Real-time smooth zooming is supported as shown in ***Figure 47***; the latter was accomplished by implementing the map rendering functionality using Direct3D.

Figure 47: Real-time smooth zooming. (zoom factors:110%, 130% and 150%).

When the user appears in the visible area, the progressive zoom-in automatically focuses on him. Otherwise, the center of the map is being focused. In detail, the Mobile Navigator performs general zoom-in with a pivot point (ux,uy) that is usually the user, assuming (cx,cy) as the camera intersection over the map plane (***Figure 48***). The pivot point (ox,oy), where the camera is initially placed, is distinguished as the midpoint of the plane. In all cases, zoom-out is performed in a similar way, assuming as the pivot point (ox,oy), i.e., the camera is tended to the origin (central) position. Regarding zoom-in, while the user is not in the view area, the camera is not moved on the XY plane, but only vertically.

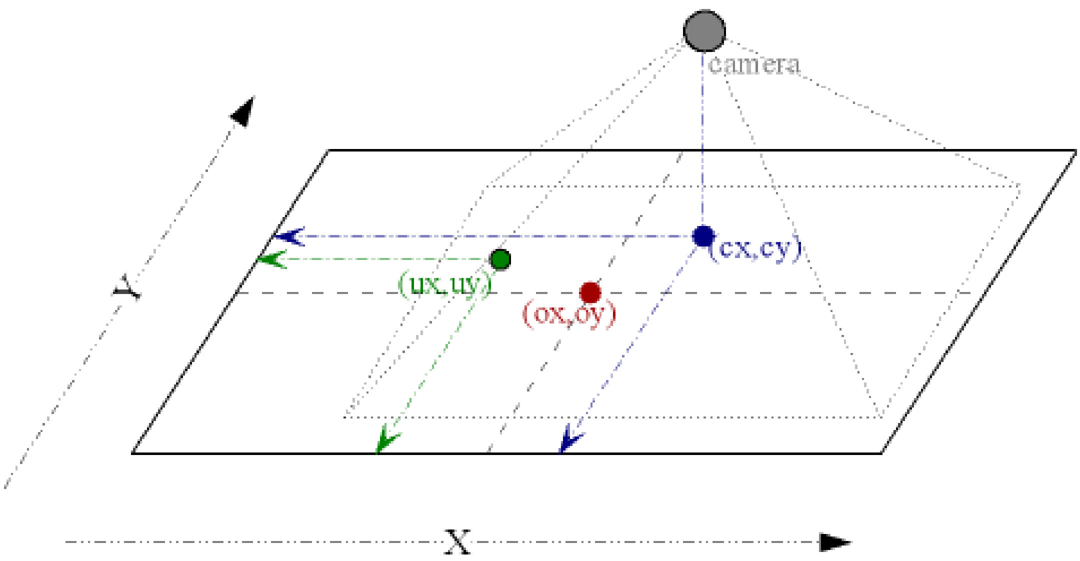


Figure 48: Smooth Zooming Map Plane

The smooth zooming model consists of the following:

- N: zooming levels number
- Current zooming level: $i \in [0, N-1]$
- Zoom$_0$ (no zoom): $cy = oy$, $cx = ox$
- Zoom$_{max}$: $cy = uy$, $cx = ux$
- Zoom$_i$: $dx = \frac{ux - cx}{N - i}$, $dy = \frac{uy - cy}{N - i}$, $cx = cx + dx$, $cy = cy + dy$

5.2.2 Hot Spot Selection

The Mobile navigator offers to the user the choice of virtual navigation. Except for directly pointing with the pen over the displayed map, the user can select information items with single button interaction. The tool "Hot Spot Selection" is an available function that the toolbar of the displayed map provides (***Figure 49***). After its activation, the user can move the focus using the hand-held arrow buttons sequentially (an especially color is used) to all visible information items. Visible information items belonging to the guidance scenario are in a higher focus priority than the rest. Using the "OK" button, the user is able to watch the selected information item presentation. Otherwise, the user can cancel the process by pressing the "map-access" button on the hand-held device.

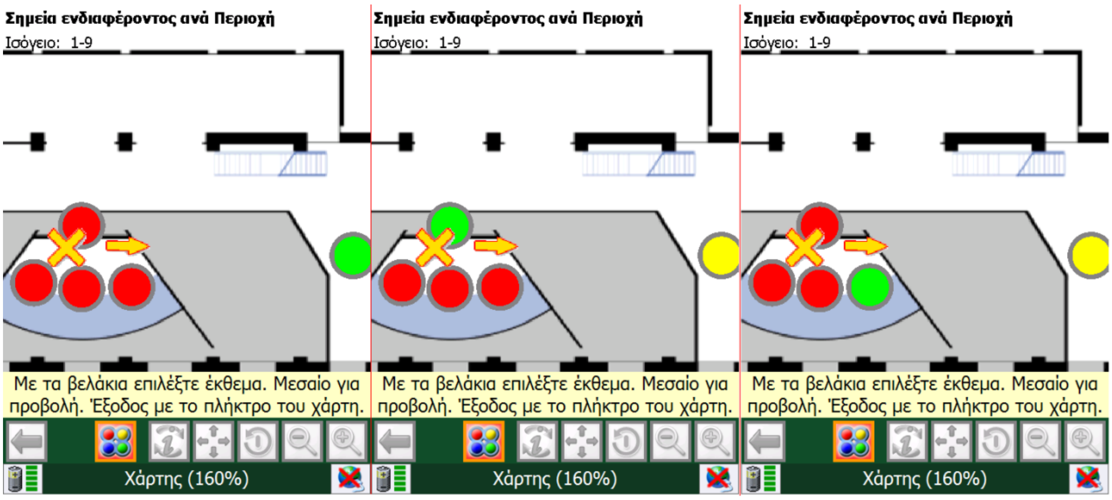


Figure 49: Hot Spot Selection (user selects the item with the arrow buttons).

5.2.3 Navigation Info

A snapshot of the navigator's map view is provided in ***Figure 50***, showing how users may interactively control the display of extra details for information items, performing a sequential transition among the following display options in a circular manner:

- Display no extra detail (default)
- Display the order of non-visited items that are part of the guidance scenario (see labels indicated with thick circles in ***Figure 50***).
- Display only the titles for all items

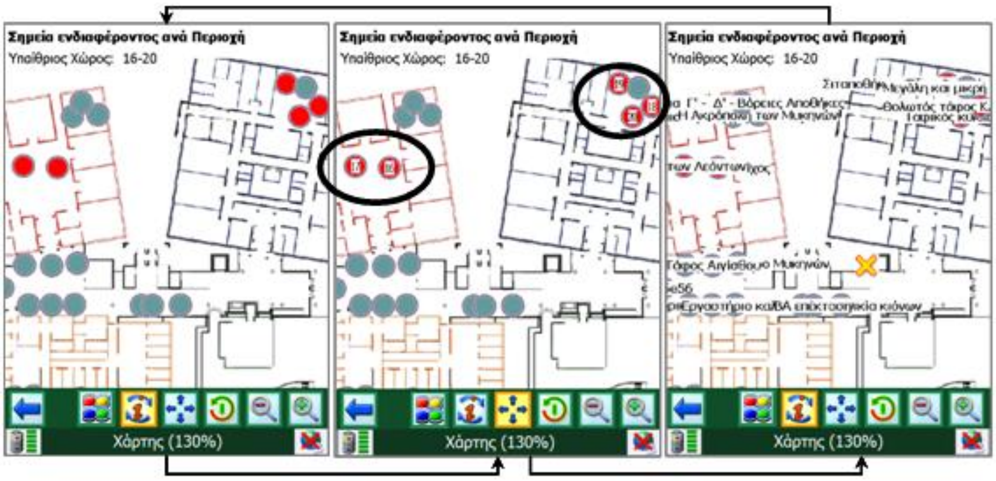


Figure 50: Controlling level of overview detail for information items.

6. Location-Sensing Management

Generally, targeting for a unified solution across indoor and outdoor settings, no single technology suffices. The idea is to enable integration of solutions with acceptable estimations, rather than to introduce a new method for location-sensing. Currently, outdoor information systems lack genericity (no indoors), practicality (satellite signal can be lost) and precision at the level of information points (coarse-grained positioning). On the other side, most indoor navigation systems employ tag technologies, like infrared beacons or radio tags, with known issues as previously mentioned. The latter technologies implement context-sensing rather than position tracking, so user monitoring and trajectory recording is minimal. Naturally, better positioning may be gained by placing a large number of tags around; however, this is hardly a globally acceptable option.

6.1 Location Sensors

Available technologies for location-sensing are considered to be the following: (a) WLan through the Ekahau Positioning Engine or CLS, (b) GPS, (c) IrFM through infrared beacons, (d) $RFID_{short}$ (e) $RFID_{long}$, and (f) Bluetooth

The most of the above location-sensing technologies were adopted in the Mobile Navigator. The use of RFID tags was rejected for usability reasons: (a) in typically crowded museum rooms tags are hardly locatable; (b) since PDA-powered readers require touch-like proximity, a social protocol is needed in case multiple visitors intend for the same exhibit, something that is less preferred due to the significant cultural diversity among visitors; and (c) they require technical equipment proximate to exhibits, which may not be allowed (e.g., the new Acropolis museum was designed with a glass-rooms, allowing extra equipment to be mounted only on the ceiling corners.

6.1.1 WLan

6.1.1.1 Ekahau Positioning Engine

The Ekahau Positioning Engine (EPE) is an enterprise-grade real time location software that leverages any standard 802.11 wireless network of any brand or generation. The EPE provides accurate location, presence, and status information in real time for Ekahau Wi-Fi tags and other supported Wi-Fi compatible devices such as mobile devices. Attaching Ekahau Wi-Fi location tags to people and objects allows the EPE to collect data on the tag whereabouts and status, establishing asset visibility within the enterprise. Location is made possible by Ekahau's innovative algorithms that calculate accurate tracking based on signal strength measurements.

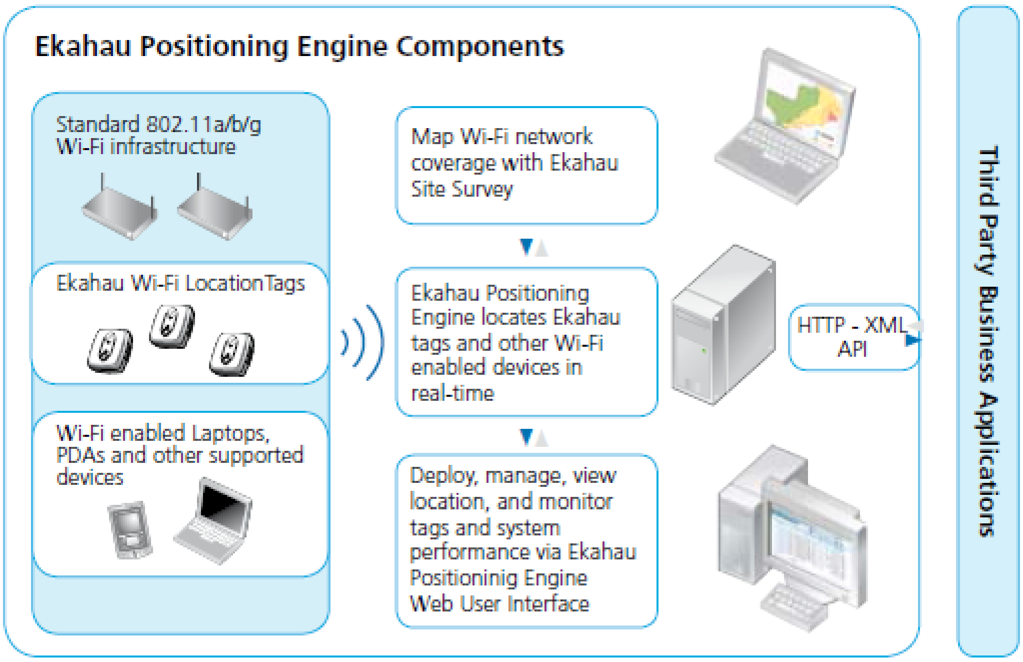


Figure 51: Ekahau Positioning Engine Components

Ekahau specification reports that the average location accuracy of a wireless device tends to be 1-2 meters (indoors, three or more overlapping access point signals). However, the results of the experiments lead to the conclusion that the accuracy of Ekahau is much lower than reported. For instance, if there are less of four installed access points, the estimated average accuracy of a wireless device is of 4-5 meters. On the other hand, if there are more than seven access points, the accuracy of the Ekahau is considerably enhanced.

6.1.1.2 CLS - Cooperative location-sensing

Cooperative Location-sensing system [11] (CLS) is a novel positioning system that employs the peer-to-peer paradigm and a probabilistic framework to estimate the position of wireless-enabled devices in an iterative manner without the need for an extensive infrastructure or time-strenuous training. CLS can incorporate signal-strength maps of the environment to improve the position estimates. Such maps have been built using measurements that were acquired from Access Points (APs) and peers during a training phase.

According to CLS, hosts cooperate and share positioning information [32]. CLS uses a grid representation that allows an easy incorporation of external information to improve the accuracy of the position estimation [13]. The evaluation of the performance of CLS via simulation according to the impact of the density of landmarks, degree of connectivity, range error, and grid resolution on the accuracy had as result that the average error is less than 2% of the transmission range, when used in a terrain with 20% of the hosts to be landmarks, average network connectivity above 7, and distance estimation error equal to 5% of the transmission range.

6.1.2 GPS

The Global Positioning System (GPS) is a satellite-based navigation system made up of a network of 24. GPS works in any weather conditions, anywhere in the world, 24 hours a day. There are no subscription fees or setup charges to use GPS.

GPS satellites circle the earth twice a day in a very precise orbit and transmit signal information to earth. GPS receivers take this information and use triangulation to calculate the user's exact location. Essentially, the GPS receiver compares the time a signal was transmitted by a satellite with the time it was received. The time difference tells the GPS receiver how far away the satellite is. Now, with distance measurements from a few more satellites, the receiver can determine the user's position.

A GPS receiver must be locked on to the signal of at least three satellites to calculate a 2D position (latitude and longitude) and track movement. With four or more satellites in view, the receiver can determine the user's 3D position (latitude, longitude and altitude). GPS's considerably precision is about 2-5 meters. The major disadvantage is that it cannot work in internal spaces, such as buildings. Other disadvantages are (a) low degree of renewal, (b) quite expensive, (c) quite energy consuming and (d) takes one or two minutes in order to lock on to the signal of at least three satellites.

6.1.3 IrFM

It concerns a light source removably attachable to a lighting fixture that supplies electrical power at a determined voltage. The light source has an attached voltage converter to provide a reduced supplied voltage. An infrared beacon is powered at the reduced supplied voltage through an electrical connection to the voltage converter. The infrared beacon broadcasts a data signal representative of a physical location. The main advantage of infrared beacons is that they provide high levels of precision. On the other hand, the major disadvantage is the average purchasing and installation cost.

6.2 Implementation

To support location-sensing, an infrastructure was implemented supporting multiple concurrent sensing channels as dynamically installed and managed API instances, each given a specific static priority relating to practical precision characteristics (i.e. trust) of its respective location-sensing approach. The infrastructure allows the Mobile Navigator to accommodate dynamically alternative location-sensing APIs (in the form of runtime loaded components with an XML configuration file listing the API components to be loaded). Such scalability and extensibility was accomplished by making APIs pertain to the same generic interface, while loaded dynamically. In particular, the statically assigned trust-based priority is taken by invoking a specific method (GetPriority – see *Figure 52*). This software architecture approach allows modularly incorporate alternative sensing technologies without modifying the original administration and navigation systems. The super-class is provided under
Figure 52, defined in C# as in the original specification.

```
using mapid_t = System.UInt32;
using areaid_t = System.UInt32;
using priority_t = System.UInt32;
using time_t = System.UInt32;
using point2d_t = Utils.UintPair; //pair<mapid_t, point2d_t>
using location_t = Utils.LocationPair; //pair<mapid_t, point2d_t>
typedef pair<mapid_t, point2d_t> location_t;

public class LocationSensingAPI {

public virtual string         GetId ();
public virtual priority_t     GetPriority ();
public virtual time_t         GetValidPeriod ();
public virtual void           SetPollingInterval (uint t);
public virtual bool           IsConnected ();
public virtual void           StartUp ();
public virtual void           CloseDown ();
};

class PointBasedAPI : public LocationSensingAPI
{ public virtual location_t GetLocation ()};
class AreaBasedAPI : public LocationSensingAPI
{ public: virtual List<areaid_t> GetAreas ()};
```

Figure 52: The generic interface for dynamically loaded location-sensing APIs.

There are two derived super-classes for location-sensing components, reflecting the classification of alternative technologies in two families: (i) those relying on user-position tracking and estimation (like Ekahau, GPS, radio tags/badges, and vision modules); and (ii) those relying on context marking (land-marking), like infrared beacons and passive/active radio tags. In this thesis technologies estimating directly user's position are referred as point-based methods, while those relying on land-marks as area-based methods.

Following this approach, the three specific technologies for location-sensing that Mobile Navigator uses, are categorized as follows with increasing priority value:

- **WLan**: point-based location-sensing technology using Ekahau.
- **GPS**: point-based location-sensing technology.
- **Infrared beacons (TAGS)**: area-based location-sensing technology.

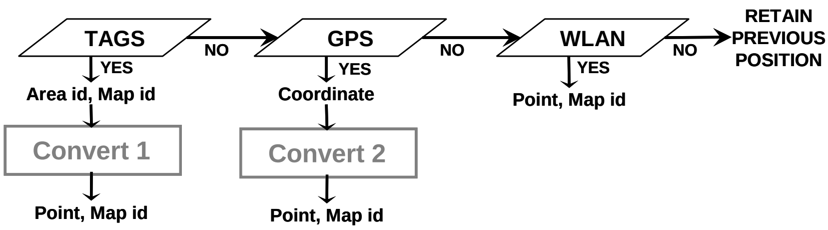


Figure 53: The location-sensing technologies ordered by trust-based priority.

The location-sensing pipeline is illustrated in ***Figure 53***, showing the way the APIs are actually deployed with decreasing priority. One should consider that this sort of processing is generalized for the case of N distinct prioritized location-sensing API components. Following ***Figure 53*** the infrared receiver API is used to check if a signal is detected. The ray actually emits a single number in which an area and a map identifier are encoded. From the area id its respective polygon and centre are extracted (indicted as Convert 1), the latter is returned together with the original map id as an approximation of the current users' position. It should be noted that the conversion of area identifiers to planar coordinates is not part of the area-based API implementation itself, but is a standard component of the overall location-sensing API management logic.

In case no signal is detected and the GPS is on, the current geographical coordinate is taken, used to identify the current enclosing map from the list of maps – maps are stored in the database with their geographical boundaries, and turn the geographical location to a raster coordinate inside this map (indicted as Convert 2). This conversion is included in the GPS point-based API implementation, returning the point and the map id as the current user's position. Finally, and alternatively, if the WLAN is on, the local EPE client API is used to inquire the current PDA position and the respective map id. As expected, if none of the previous technologies is active, the old user position is retained. The adoption of this particular implementation style to cope with varying location-sensing technologies has been proved to be a key complexity-reduction factor in this architecture.

6.2.1 Basic Infrastructure of the GPS Location API

A class library was implemented supporting the functionality and requirements to support the GPS location-sensing system (GPSLocator.dll). The GPS location API encapsulates the interface of the class library by offering only the necessary functions to the Mobile Navigator's Positioning System (see *Figure 52*).

The class library supports the following facilities.

- Setup a serial port
- Open a serial port
- Read from a serial port
- Close a serial port
- Determine the status of the serial port
- Parse NMEA sentences
- Provide geographical coordinates (Latitude and Longitude)
- Convert between geographical and pixel coordinates

All actions are carried out asynchronously using a combination of methods and events. Typically for a single action, a method is called to start the action. The method returns immediately. Sometime later, one or more events (DataReceived and/or ErrorOccured) are fired.

The constructor method of the class accepts one string parameter. This parameter defines the port name that it will be used to accept GPS data.

The properties of the serial port and also the geographic coordinates and the initial screen coordinates have to be defined in order to start reading GPS data from the serial port and to convert Lon/Lat to X/Y screen coordinates.

Figure 54: GPS Infrastructure, Lon/Lat to X/Y screen coordinates.

The class library is used with a configuration file namely App.config as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="HDOP" value="10" />
    <add key="CellID" value="1" />
    <add key="InitX" value="0" />
    <add key="InitY" value="0" />
  </appSettings>

</configuration>
```

The initial X, Y coordinates on the device screen, the Cell ID as well as the maximum allowed horizontal dilution of precision is also defined in the configuration file. Signals with HDOP value lower than the defined maximum HDOP, will raise an ErrorOccured event.

The App.config XML file has to be copied in the same directory as the class library (GPSLocator.dll). If the App.config file is not found, the values of the will be defined with the default hard coded values, which are:

- HDOP = 10
- CellID = 1
- InitX = 0
- InitY = 0

6.2.1.1 Methods

The methods of the GPSLocator class library are the following:

- **StartGPS:** This procedure opens the serial port and starts reading data from the GPS device. In order to start reading from the port, the port has to be initialized using the properties described below.

- **StopGPS:** This procedure closes the serial port

6.2.1.2 Events

(a) **DataReceived**: This event is fired after the StartGPS() method has been called. The data returned from the chip are passed as an event String array parameter.

Prototype

DataReceived(data() as String)

Parameters

Data()*:* A string array containing the following values

- **Index 0:** The cell number (Defined in the App.Config XML file)
- **Index 1:** The X coordinate of the current geographic location, geo-referenced on the images grid
- **Index 2:** The Y coordinate of the current geographic location, geo-referenced on the images grid
- **Index 3:** The current Latitude
- **Index 4:** The current Longitude
- **Index 5:** The current Horizontal Dilution of Precision (HDOP)
- **Index 6:** The current no of satellites fixed on the GPS signal

(b) **ErrorOccurred:** This event is fired after the StartGPS() or StopGPS() methods have been called. The data returned from the chip are passed as an event String parameter.

Prototype

ErrorOccurred (errorMessage as String)

Parameters

errorMessage:A string containing an error message.

6.2.1.3 Properties

- **BottomLeftLon (type = double) :** The bottom left Longitude value of the image grid
- **BottomLeftLat (type = double) :** The bottom left Latitude value of the image grid
- **BottomRightLon (type = double) :** The bottom right Longitude value of the image grid
- **BottomRIghtLat (type = double) :** The bottom right Latitude value of the image grid

- **TopRightLon (type = double) :** The top right Longitude value of the image grid
- **TopRightLat (type = double):** The top right Latitude value of the image grid
- **InitX (type = double):** The initial X coordinate on the device screen. Default value 0.
- **InitY (type = double):** The initial Y coordinate on the device screen. Default value 0.
- **Baudrate (type = integer):** Baud rate of the GPS serial port. Default value 57600.
- **Parity (type = IO.Ports.Parity):** The parity value of the GPS serial port. Default value IO.Ports.Parity.None
- **StopBits (type = IO.Ports.StopBits):** The stop bits value of the GPS serial port. Default value IO.Ports.StopBits.One.
- **DataBits (type = integer):** The data bits value of the GPS serial port. Default value 8.
- **HandShake (type = IO.Ports.Handshake):** The handshake value of the GPS serial port. Default value IO.Ports.Handshake.None.
- **ReadTimeout (type = integer):** The read timeout value of the GPS serial port. Default value 5000 milliseconds.
- **DtrEnable (type = boolean):** The value of the Data Terminal Ready pin of the GPS serial port. Default value True.
- **RtsEnable (type = boolean):** The value of the Request To Send pin of the GPS serial port. Default value True.
- **MaximumAllowedHDOP (type = double):** The maximum value of the Horizontal Dilution of Precision accepted by an external application. Default value 6.

7. Application in a Museum

The platform was installed, deployed and evaluated in the Natural History Museum of Crete, encompassing a total of thirty hand-held devices.

7.1 Natural History Museum of Crete

The Natural History Museum of Crete has been functioning and operates under the framework of the University of Crete since 1980, being a pioneer institute at national and European level in the following activities:

- Study and Management of the Natural Environment of Crete
- Public awareness, education and sensitization of local people as well as the visitors of the area
- Link University activities with Cretan society
- Set up a network of Ecological Museums in Greece and throughout the Eastern Mediterranean area.

7.2 Equipment Used

7.2.1 Mobile Device

The hand-held device used was the Fujitsu Siemens Pocket LOOX N520. It measures 71 x 116 x 14mm and much of the surface area is taken up with a 3.5-inch TFT colour screen which has a conventional resolution of 640 x 480pixels.

The controls at the bottom of the screen are illuminated by a blue backlight, which makes them incredibly easy to find in the dark (i.e., useful for a dark exhibition center like an Aquarium) while it weights only 146g.

Figure 55: Fujitsu Siemens Pocket Loox N520

The rest of the specification consists of a 312MHz Intel XScale PXA270 processor with 64MB RAM and 128MB ROM to give a total of 192MB. There's also 802.11b/g wireless, SiRFStar III GPS, an SD card slot and Windows Mobile 5.

7.2.2 Servers

Two DELL PowerEdge™ R300 servers were installed at the NHMC in order to fullfill the platform’s requirements. The main services running at the first server was the Ekahau Server as well as the server-side of the Real Time Monitoring System. In the second server, the main running service was the SQL Server, parts of the Device and Renting Management System as well as the Central Content Management System. Other applications (i.e., Renting Application or Monitoring Viewer) have been already installed in a personal computer found at the reception.

7.2.3 Network Devices

Considering the network architecture, more than 20 wireless access points were installed to cover the museum building (approximately 6 on each floor). Firstly, some main characteristics of the wireless access points used are the following (a) IEEE 802.11g compliant, (b) WPA2 enabled, (c) POE capable and (d) support for

VLANs. Secondly, as regarding the switches they are: (a) manageable, (b) VLAN capable, (c) POE capable, (d) 2 optical interfaces and (e) 2 gigabit interfaces. Thirdly, concerning the routers used the characteristics are: (a) ADSL interface, (b) gigabit interfaces and (c) VPN capable. Finally, the minimum satisfied requirements for the servers used are: (a) a At least Dual-core Intel Xeon 3000 series and (b) 2GB ram.

7.2.4 Infrared Beacons

More than 30 infrared beacons were selectively installed to the NHMC, using a model with up to six LEDs per device, placed at the top corners with a 45 degrees angle downwards (Figure 56).

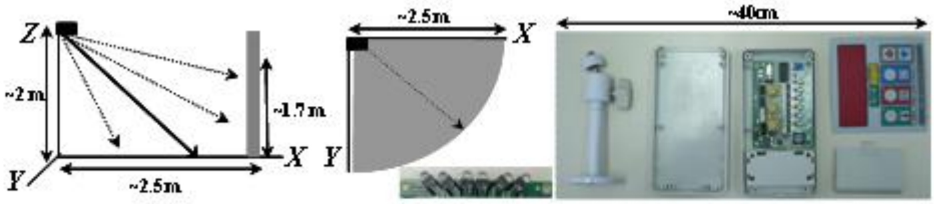


Figure 56: Typical placement and boxing of infrared beacons in room corners.

7.3 Deployment Architecture

The following figure provides an overview of the overall system that is installed for supporting the electronic augmentation of the museum, illustrating its distinct functional components, as well as how and where these were deployed.

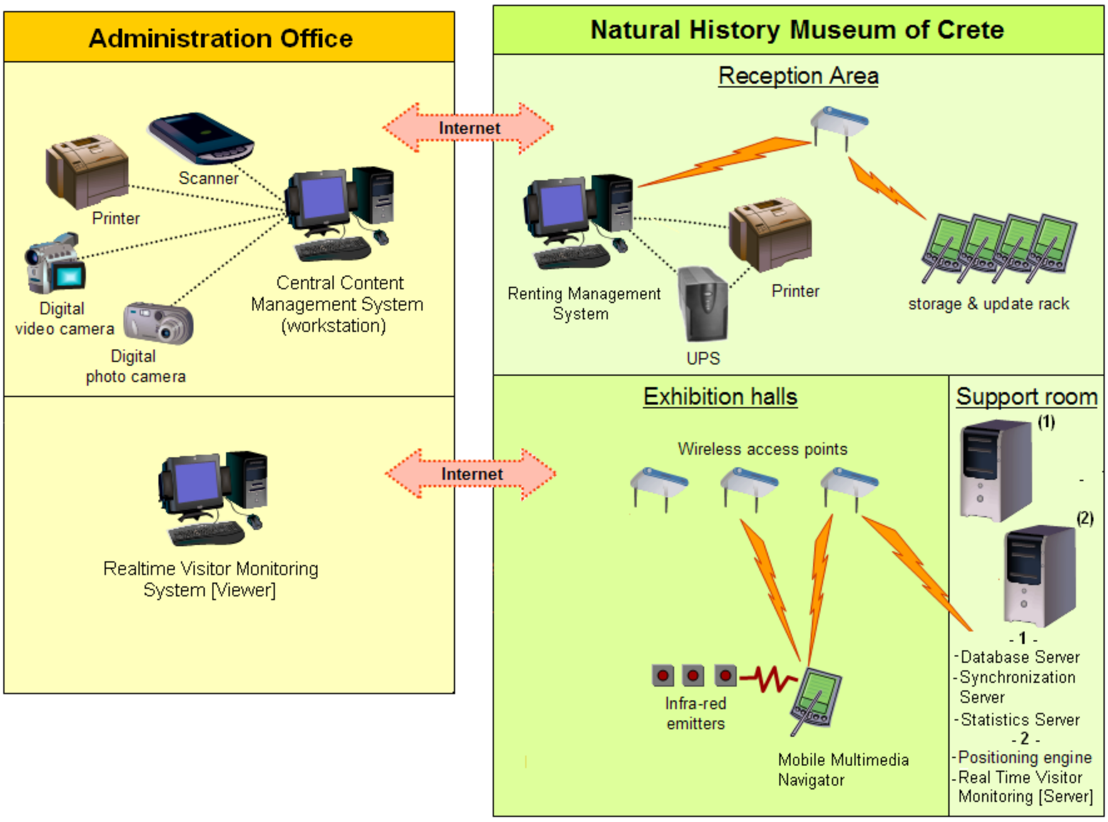


Figure 57: Deployment Architecture at NHMC

7.4 Training Process

In order the Ekahau Positioning System to provide as more as considerably location precision, a training process must to be done. The process, also known as calibration, aims at creating a distinction among physical cells using signal strength signatures from wireless access points. The calibration process consists of the following steps:

- With the calibration software mark a square grid out on the floor of the area at every possible point of interest (i.e., points located exhibits or points where the user stands for a few seconds during his navigation session). Each point should be at least 5 feet from the other points.
- Running the calibration software on a laptop, the following procedure must be initiated by somebody:
    (a) standing on grid point i.

(b) 360 degree smooth turning while standing on grid point i for exactly 20 seconds time period.

(c) moving to grid point i+1 and repeating from the first step (a) until the end of all the grid points.

This procedure allows the location-sensing engine to record the signal-strength signature pattern of each cell. The goal between the 20-second calibration time of each point and 360-degree turn is to retrieve a valid signal strength sample from the available wireless access points, one that is not affected by temporal changes [31].

7.5 Renting Process

In this context, it should be noticed that existing location-aware mobile information systems do not deal with the issue of explicit device renting and returning processes. The overall platform supports very fast device management and renting processes (charging of visitors is optional) as shown in ***Figure 58***.

To achieve this, any mobile device has a pre-defined unique barcode number at the back surface. In order to start the navigation session, the museum employee uses the barcode scanner to rent the mobile device according to the desirable navigation type and language (***Figure 58***). After navigation completion, the same process is followed to un-rent the mobile device.

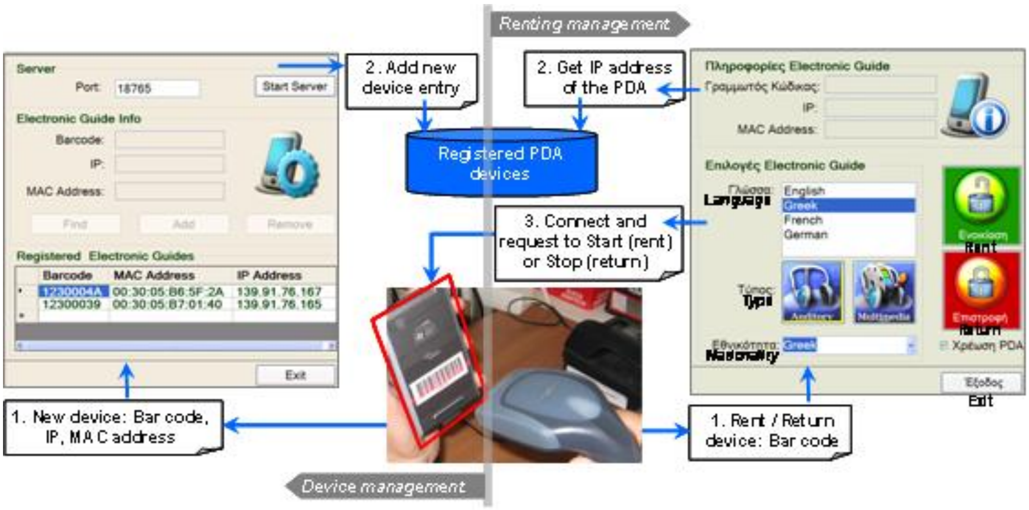


Figure 58: Device and renting management systems and processes.

8. Conclusion

The Mobile Navigator was designed and developed for deploying mobile location-aware information systems, such as hand-held museum guides. It is the main component of an overall integrated platform. The latter comprises a set of management and authoring tools and supports the creation of large-scale systems with very crowded use sessions (i.e., hundreds of simultaneous museum visitors).

The Mobile Navigator can be used in both indoor (e.g., museums) and outdoor environments (e.g., archaeological sites), as well as in environments that comprise both indoor and outdoor spaces. The delivered information can be provided in any number of alternative languages. Custom guidance scenarios are supported as mentioned in section 4.11. Beyond multimedia information, users can also access a map (see section 5.1), where their current position is depicted along with information concerning surrounding exhibits and points of interest. In order to achieve the maximum possible accuracy in a flexible and efficient way, it combines multiple location-sensing technologies, including WLAN, GPS, and infrared beacons (see chapter 6).

In this context, the conclusions of what is discussed in the thesis are: (a) the unified location-sensing infrastructure for indoors and outdoors simplifies maintenance and extensibility; (b) trust-based statically-prioritized multi-channel location-sensing solves many precision problems increasing significantly the robustness of the position tracking infrastructure, even though it puts an extra authoring overhead; and (c) when tags are introduced to augment the location-sensing approach, they should be programmed to identify locations or areas, i.e. return geometrical information, rather than transmitting internal database keys or indices of particular information items.

9. Future Work

A very important issue that must be studied is the positioning system. Besides the fact that multiple location-sensing technologies are in use, position identification seriously lacks in terms of accuracy.

Filtering of misleading values (i.e., “wall jumping”) can be achieved by adding additional information to the maps that the Mobile Multimedia Navigator uses. As a result, structured maps must be included in the content in order to replace the images. Structured maps will include information about the walls or physical obstacles. Besides the filtering of misleading values, the Mobile Multimedia Navigator will be able to provide information to support path planning.

Regarding the graphical user interface, future work can be done in order to support interactive configuration that will allow the user to control or customize the interface. e.g., selecting to hide some controls or viewing less detailed information.

Finally, taking advantage of statistics analysis, Mobile Multimedia Navigator could offer auto-generated guidance scenarios including for example the most popular information points.

10. References

[1] Sumi, Y., T. Etani, S. Fels, N. Simone, K. Kobayashi and K. Mase. "C-MAP: Building a Context-Aware Mobile Assistant for Exhibition Tours". Social Interaction and Communityware, Japan, June 1998.

[2] Want, R., A. Hopper, V. Falcao, and J. Gibbons. "The active badge location system". ACM Transactions on Information Systems, 10(1): pp. 91-102, January 1992.

[3] Long, S., D. Aust, G. Abowd, C. Atkeson. "Cyberguide: Prototyping Context-Aware Mobile Applications". Proceedings of the conference on Human Factors in Computing Systems, CHI'96.

[4] Smailagic, A. and R. Martin. "Metronaut: A Wearable Computer with Sensing and Global Communication Capabilities". Proceedings of the International Symposium on Wearable Computing, IEEE, 1997.

[5] Kreller, B., D. Carrega, J. Shankar, P. Salmon, S. Bottger, and T. Kassing. "A Mobile-Aware City Guide Application". ACTS Mobile Communication Summit, Rhodos, Greece, 1998.

[6] Rekimoto, J, Y. Ayatsuka and K. Hayashi. "Augmenta-able Reality: Situated Communication through Physical and Digitial Spaces". Proceedings of the International Symposium on Wearable Computing, IEEE, 1998.

[7] Mynatt E., M. Back, R. Want. "Designing Audio Aura". Proceedings of the conference on Human Factors in Computing Systems, CHI'99.

[8] Lamming, Mik and Mike Flynn. "Forget-me-not: Intimate computing in support of human memory". FRIEND21: International Symposium on Next Generation Human Interface, pp. 125-128, 1994

[9] Rhodes, B. “The Wearable Remembrance Agent: a system for augmented memory”. Proceedings of the International Symposium on Wearable Computing, IEEE, 1997.

[10] Theodore Patkos, Antonis Mpikakis, Grigoris Antoniou, Maria Papadopouli, Dimitris Plexousakis. "A Semantic-based Framework for Context-Aware Pedestrian Guiding Services", Second International Workshop on Semantic Web Technology For Ubiquitous and Mobile Applications 2006, Riva del Garda, Trentino, Italy August 28, 2006

[11] Charalampos Fretzagias and Maria Papadopouli, "Cooperative location-sensing for wireless networks", Second IEEE International conference on Pervasive Computing and Communications 2004, Orlando, Florida, March 14-17, 2004

[12] M. van Setten, S. Pokraev, and J. Koolwaaij, ‘Context-aware recommendations in the mobile tourist application COMPASS’, in Proc. 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2004), Eindhoven, The Netherlands, (2004).

[13] Wasp, http://wasp.freeband.nl (2006).

[14] OWL Web Ontology Language, http://www.w3.org/TR/owl-features/

[15] Woodruff, A., Aoki, P.M., Hurst A., and Szymanskim M. H. (2001). Electronic Guidebooks and Visitor Attention. Proc. 6th Int’l Cultural Heritage Informatics Meeting, Milan, Italy, Oct. 2001, 623-637.

[16] Butz A., Baus J., Kruger A., (2000) Augmenting buildings with infrared information. In: Proceedings of the International Symposium on Augmented Reality (ISAR 2000), IEEE Computer Society Press, New York.

[17] Oppermann R., Specht M. (2000) A context-sensitive nomadic exhibition guide. In: The proceedings of symposium on handheld and ubiquitous computing, LNCS 1927, Springer, Berlin Heidelberg New York, pp 127–142

[18] Ciavarella, C. and Paterno, F. (2004). The Design of a Handheld, Location-Aware Guide for Indoor Environments. Personal and Ubiquitous Computing 8, 82–91.

[19] Chou, L.-D., Wu, C.-H., Ho, S.-P., Lee, C.-C., and Chen J.M., (2004b). Requirement Analysis and Implementation of Palm-based Multimedia Museum Guide Systems. AINA 2004, 18th International Conference on Advanced Information Networking and Applications, vol.1, pp. 352 - 357.

[20] Chan L.-W., Hsu Y.-Y, Hung Y.P., Hsu J. Y.-j, (2005). Orientation-Aware Handhelds for Panorama-Based Museum Guiding System. UbiComp 2005, Workshop: Smart Environments and Their Applications to Cultural Heritage, Tokyo, September 2005.

[21] Hsi, S., (2003). The Electronic Guidebook: A Study of User Experiences Mediated by Nomadic Web Content in a Museum Setting. Journal of Computer-Assisted Learning, vol. 19, no 3 (September): 308–319.

[22] Hsi, S. (2004) I-Guides in progress: Two prototype applications for museum educators and visitors using wireless technologies to support informal science learning. Proceedings of the 2nd IEEE International Workshop on Wireless and Mobile.

[23] Bay, H., Fasel, B., Van Gool, L., (2005). Interactive Museum Guide. The Seventh International Conference on Ubiquitous Computing UBICOMP, Workshop on Smart Environments and Their Applications to Cultural Heritage, September 2005

[24] Albertini, A., Brunelli, R., Stock, O., and Zancanaro, M. (2005). Communicating user's focus of attention by image processing as input for a mobile museum guide. In Proceedings of the 10th international Conference on intelligent User interfaces (San Diego, California, USA, January 10 - 13, 2005). IUI '05. ACM Press, New York, NY, 299-301.

[25] Yatani, K., Sugimoto, M., Kusunoki, F., (2004). Musex: A System for supporting children's collaborative learning in a museum with PDAs, International Workshop on Wireless and Mobile Technology in Education(WMTE 2004) 109-113

[26] Kindberg, T. and O'Hara, K. (2006). "You Are Here" - Experiences with Located Symbols. HP Technical report HPL-2006-120

[27] Nikolaos Kazepis, "Integrated platform for location-aware information systems: Spatial Content Administrator", M. Sc. Thesis, Computer Science Department, University of Crete, October 2008

[28] Stefanos Doumboulakis, "Visitor's monitoring in a mobile, location-aware, information system"", M. Sc. Thesis, Computer Science Department, University of Crete, November 2008

[29] Microsoft. .NET Compact Framework. http://msdn2.microsoft.com/engb/netframework/Aa497273.aspx.

[30] Feeney, L.M. Nilsson, M. "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment",

INFOCOM 2001, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Anchorage, AK, USA.

[31] The limits of localization using signal strength: a comparative study. Elnahrawy, E., Li, Xiaoyan and Martin, R. P. 2004. Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on. pp. 406--414.

[32] Konstantinos Vandikas, Lito Kriara, Tonia Papakonstantinou, Anastasia Katranidou, Haris Baltzakis and Maria Papadopouli. Empirical-based analysis of a cooperative location-sensing system. First International Conference on Autonomic Computing and Communication Systems (ACM Autonomics'07), Rome, Italy, 28-30 October 2007.