Citation: Papakonstantinou, Stylianos (2012). Technological framework for ubiquitous interactions using context–aware mobile devices. (Unpublished Doctoral thesis, City University London)

CITY UNIVERSITY
LONDON

Technological Framework for Ubiquitous Interactions using Context-aware Mobile Devices

---

STYLIANOS K. PAPAKONSTANTINOU

Submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy (PhD) in Information Science

School of Informatics

Department of Information Science

Volume I

Supervisors: Prof. Vesna Brujic-Okretic & Prof. Jonathan Raper

London, August 2012
U.K.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Requirements

Acknowledgements

I would like to express my gratitude to everyone that has contributed or has been involved in the realisation of the research.

First, and most importantly, I would like to thank my *Family* and especially my *Parents* who have unconditionally supported me in every possible way. Both of them stood next to me and encouraged every decision that I made, regardless of how it affected them. I am truly obliged to them and I hope that they receive what they expect from their son. I owe them everything that I have accomplished so far and more.

Several people that I have met, discussed and collaborated with within *City University London*, the *Department of Information Science* and particularly in the *giCentre* have inspired me over the past few years. Dr. *Vesna Brujic-Okretic,* my primary supervisor, who sincerely trusted me from the beginning of this endeavour and invested in me as a researcher and as a person. Her dedication in helping me was remarkable. Prof. *Jason Dykes* provided valuable suggestions especially during the preliminary exploration of the subject. Furthermore, Dr. *Fotis Liarokapis* planted the seed in order for me to make the decision to pursue a PhD degree. His experience and guiding proved invaluable during the course of the project and for those reasons I need to thank him. Dr. *Andrew MacFarlane* was always there when I needed him, especially for clarifying any queries regarding the research process. I must also express my gratitude to Dr. *David Mountain* who advised and frequently spent time with me in order to produce high quality results in every project that we collaborated. The research guidance that I received from my second supervisor, Prof. *Jonathan Raper,* was invaluable for producing innovative findings and his constructive criticism added further value to the research. Likewise, I received significant assistance from Dr. *Jo Wood*, especially during the last part of the research, when he helped me validate the results of the statistical analysis. Conclusively, the comments that I obtained during both preliminary examinations by Dr. *Sebastian Hunt* and Dr. *Abdul Roudsari* proved vital for the continuation of the project.

Some aspects of work contained within this Thesis were inspired by the *LOCUS* project, which for three years offered an incredibly rewarding environment in which to work, and I would like to express my gratitude to every individual involved in it.

Furthermore, I am grateful to the administration of the *School of Informatics*, which has funded this study and invested in me as a scholar and a potential academic.

Following next, to the members of *City University Research & Enterprise Unit* and especially Dr. *Tim Meldrum* who gave me the opportunity to expose the developments of this research project to commercial investors and learn about the business trends that affect the applicable markets.

Two individuals that I consulted, in order to overcome certain issues that affected the development aspects of the work and the final polishing of the Thesis are *Nontas Chrysikos* and *Sofia Papanastassiou*. Furthermore, Prof. *Christos Douligeris* provided valuable comments regarding the final amendments. I really appreciate the effort that you put in helping me and I hope to be able to reciprocate you.

I must not forget to express my gratitude to my fellow *PhD Students*, who have shared their knowledge and expertise, as well as for the long creative hours that we pleasantly spent together in the office. Furthermore, I am in debt to every *Individual* that has trusted me and participated in the surveys that I organised for accomplishing the objectives of this project.

In addition, I am extremely grateful to my *Friends* that supported my decisions and were committed to our friendship, even though I had to let them down some times. They deserve my appreciation because they realised that I had to sacrifice a lot in order to accomplish my career goals and for their empathy. I must also be grateful to the loves of my life because they put a small stone in the development of my character and helped me reach this point in career and life, by making me more dedicated to my goals when needed most.

I would also like to thank those who posed complications in finishing the research, which resulted in a longer than planned time scale, but also made me accomplish certain achievements on my own. Pursuing a PhD degree was a great experience, which I will cherish and shall never forget. You were all, in one way or another, an important element of this significant endeavour!

Declaration

I grant powers of discretion to the University Librarian to allow the Thesis to be copied in whole or in part without further reference to the author. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

.

Stylianos Papakonstantinou, August 2012

Abstract

This report presents research and development of dedicated system architecture, designed to enable its users to interact with each other as well as to access information on Points of Interest that exist in their immediate environment. This is accomplished through managing personal preferences and contextual information in a distributed manner and in real-time. The advantage of this system architecture is that it uses mobile devices, heterogeneous sensors and a selection of user interface paradigms to produce a socio-technical framework to enhance the perception of the environment and promote intuitive interactions. The thrust of the work has been on software development and component integration. Iterative prototyping was adopted as a development method in order to effectively implement the users' feedback and establish a platform for collaboration that closely meets the requirements and aids their decision-making process. The requirement acquisition was followed by the system-modelling phase in order to produce a robust software prototype. The implementation includes component-based development and extensive use of design patterns over native programming. Conclusively, the software product has become the means to evaluate differences in the use of mixed reality technologies in a ubiquitous scenario.

The prototype can query a number of context sources such as sensors, or details of the personal profile, to acquire relevant data. The data (and metadata) is stored in open-source structures, so that they are accessible at every layer of the system architecture and at any time. By proactively processing the acquired context, the system can assist the users in their tasks (e.g. navigation) without explicit input – e.g. by simply creating a gesture with the device. However, advanced interaction with the application via the user interface is available for requests that are more complex.

Representations of the real world objects, their spatial relations and other captured features of interest are visualised on scalable interfaces, ranging from 2D to 3D models and from photorealism to stylised clues and symbols. Two principal modes of operation have been implemented; one, using geo-referenced virtual reality models of the environment, updated in real time, and second, using the overlay of descriptive annotations and graphics on the video images of the surroundings, captured by a video camera. The latter is referred to as augmented reality.

The continuous feed of the device position and orientation data, from the GPS receiver and the digital compass, into the application, makes the framework fit for use in unknown environments and therefore suitable for ubiquitous operation. This is one of the novelties of the proposed framework, because it enables a whole range of social, peer-to-peer interactions to take place. The scenarios of how the system could be employed to pursue these remote interactions and collaborative efforts on mobile devices are addressed in the context of urban navigation. The conceptual design and implementation of the novel location and orientation based algorithm for mobile AR are presented in detail. The system is, however, multifaceted and capable of supporting peer-to-peer exchange of information in a pervasive fashion, usable in various contexts. The modalities of these interactions are explored and laid out in several scenarios, but particularly in the context of user adoption. Two evaluation tasks took place. The preliminary evaluation examined certain aspects that influence user interaction while being immersed in a virtual environment, whereas the second summative evaluation compared the utility and certain usability aspects of the AR and VR interfaces.

List of Abbreviations

2	
2D	Two (2) Dimensions
3	
3D	Three (3) Dimensions
3G	Third (3rd) Generation Mobile Communication Technologies
4	
4G	Fourth (4th) Generation Mobile Communication Technologies
A	
A-GPS	Assisted GPS
AAL	ATM Adaptation Layer
AI	Artificial Intelligence
API	Application Programming Interface
AR	Augmented Reality
ATL	Active Template Library
ATM	Asynchronous Transfer Mode
AUML	Agent Unified Modelling Language
C	
CC/PP	Composite Capabilities / Preferences Profile
CDC	Connected Device Configuration
CDI	Content Distribution Internetworking
CE	Compact Edition
CD	Compact Disc
CG	Computer Graphics
CLDC	Connected, Limited Device Configuration
CMS	Context Management System
CODEC	COmpressor/DECompressor
COM	Component Object Model
COTS	Commercial Off-The-Shelf
CORBA	Common Object Request Broker
CPU	Central Processing Unit
CRM	Customer Relationship Management
CVE	Collaborative Virtual Environments
D	
D3DM	Direct3d Mobile
D-GPS	Differential GPS
DB	Database
DBMS	DataBase Management System
DCOM	Distributed Component Object Model
DEM	Digital Elevation Model

DM	DATA MINING
DOF	DEGREES OF FREEDOM
DSSS	DIRECT-SEQUENCE SPREAD SPECTRUM
DTI	U.K. DEPARTMENT OF TRADE AND INDUSTRY
DTD	DOCUMENT TYPE DEFINITIONS
E	
EDGE	ENHANCED DATA RATES FOR GSM EVOLUTION
EGL	EMBEDDED SYSTEMS GRAPHICS LIBRARY
EGNOS	EUROPEAN GEOSTATIONARY NAVIGATION OVERLAY SYSTEM
EOTD	ENHANCED OBSERVED TIME DIFFERENCE OF ARRIVAL
ER	ENTITY RELATIONSHIP
EU	EUROPEAN UNION
F	
FAI	FÉDÉRATION AÉRONAUTIQUE INTERNATIONALE
FIPA	FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS
FOR	FRAME OF REFERENCE
FOV	FIELD OF VIEW
FPS	FRAMES PER SECOND
G	
GB	GIGA BYTE
GDI	GRAPHICS DEVICE INTERFACE
GHz	GIGA HERTZ
GIS	GEOGRAPHICAL INFORMATION SYSTEM (OR SCIENCE)
GML	GEOGRAPHY MARK-UP LANGUAGE
GNSS	GLOBAL NAVIGATION SATELLITE SYSTEM
GPRS	GENERAL PACKET RADIO SERVICE
GPS	GLOBAL POSITIONING SYSTEM
GPU	GRAPHIC PROCESSOR UNIT
GPX	GPS EXCHANGE
GSM	GLOBAL SYSTEM FOR MOBILE COMMUNICATIONS
GUI	GRAPHIC USER INTERFACE
H	
H/W	HARDWARE
HCI	HUMAN COMPUTER INTERACTION
HMD	HEAD MOUNTED DISPLAY
HSDPA	HIGH-SPEED DOWNLINK PACKET ACCESS
HSUPA	HIGH-SPEED UPLINK PACKET ACCESS
HTML	HYPERTEXT MARK-UP LANGUAGE
HTTP	HYPERTEXT TRANSFER PROTOCOL
I	
ICT	INFORMATION COMMUNICATION TECHNOLOGY
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
IDL	INTERFACE DEFINITION LANGUAGE
IE	INTERNET EXPLORER

IETF	Internet Engineering Task Force
IPC	Inter-Process Communication
IPS	Information Presentation System
IR	Information Retrieval
ISP	Internet Service Provider
ISV	Independent Software Vendors
IT	Information Technology
ITU	International Telecommunication Union
J	
J2EE	Java Enterprise Edition
J2ME	Java Mobile Edition
J2SE	Java Standard Edition
JSR	Java Specification Request
K	
Kbps	Kilo Bits Per Second
KDD	Knowledge Discovery in Databases
L	
LBS	Location Based Service
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
LOD	Level Of Detail
M	
M3G	Mobile 3D Graphics
mAh	MilliAmpere Hour
MB	Mega Byte
Mbps	Mega bits per second
MDA	Mobile Data Association
MEMS	Micro-Electro Mechanical Systems
MFC	Microsoft Foundation Classes
MIDP	Mobile Information Device Profile
MMS	Multimedia Messaging Service
MP	Mega Pixel
MR	Mixed Reality
N	
NMEA	National Marine Electronic Association
NPC	Non-Player Character
NPR	Non-Photorealistic rendering
NTP	Network Type Protocol
NVML	NaVigation Mark-up Language
O	
OFDM	Orthogonal Frequency-Division Multiplexing
OEM	Original Equipment Manufacturer
OGC	Open Geospatial Consortium
OGLES	OpenGL for Embedded Systems

OLE	Object Linking and Embedding
OMA	Open Mobile Alliance
OO	Object-Oriented
OP	Optional Package
OPES	Open Pluggable Edge Services
OS	Operating System
OSI	Open Systems Interconnection
P	
P2P	Peer to Peer
PAN	Personal Access Network (Bluetooth)
PC	Personal Computer
PCB	Printed Circuit Board
PDA	Personal Digital Assistant
PDT	Position Determination Technology
PIE	Pocket Internet Explorer
PND	Personal Navigation Device
POI	Point of Interest
POIX	Point Of Interest eXchange
PSE	Personal Service Environment
Q	
QOS	Quality of Service
QR	Quick Response
R	
RAM	Random Access Memory
RE	Requirements Engineering
RFC	Request for Comments
RFID	Radio Frequency IDentification
RMI	Remote Method Invocation
ROM	Read Only Memory
RPC	Remote Procedure Call
S	
S/W	Software
SBSOD	Santa Barbara Sense of Direction Scale
SD	Strategic Dependency model
SDK	System Development Kit
SMS	Short Messaging Service
SR	Strategic rationale model
SVG	Scalable Vector Graphics
T	
TCP/IP	Transmission Control Protocol / Internet Protocol
TDOA	Time Difference of Arrival
TFT	Thin Film Transistor
TMC	Traffic Message Channel
TOA	Time of Arrival

U	
UAProf	User Agent Profile
UDP	User Datagram Protocol
UI	User Interface
UML	Unified Modelling Language
UMPC	Ultra Mobile Personal Computers
UMTS	Universal Mobile Telecommunications System
UTC	Coordinated Universal Time
UTM	Universal Transverse Mercator
UUID	Universally Unique IDentifier
URL	Universal Resource Locator
UX	User eXperience
V	
VE	Virtual Environment
VR	Virtual Reality
VRML	Virtual Reality Modelling Language
VSWM	Visuo-Spatial Working Memory
W	
W-CDMA	Wideband Code Division Multiple Access
W3C	World Wide Web Consortium
WDS	Wireless Distribution System
WEBI	Web Intermediaries
WGS84	World Geodetic System 1984
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
WP7	Windows Phone 7
WWW	World Wide Web
X	
X3D	eXtensible 3D
XML	eXtensible Mark-up Language

1 Introduction

This chapter constitutes an introduction to the theme of the research undertaken in this project. It introduces the motivation that has driven the research, followed by the principal aims and objectives of the project. Furthermore, the measurable contributions of the research and the contribution made to knowledge in the course of these investigations are presented. The chapter concludes with a brief presentation of the organisation of the material and an overview of the document structure.

1.1 Research Problem & Questions

In the last few years, the availability of sophisticated consumer mobile devices has increased rapidly. Modern devices are equipped with the latest technological features, such as fast processors, dedicated graphics acceleration and several types of communication enabling interfaces (e.g. 3G, Wi-Fi, Bluetooth). A recent trend followed by device manufacturers is to embed context-aware sensors that can fuse real-time information to the applications that run on the device. Therefore, these devices, and up to an extent the applications, are suitable for the ubiquitous provision of real-time information to the users of the system. This can prove extremely beneficial to the users in many ways. For instance, by processing spatial context (i.e. position and orientation information), the users can visualise and interact with certain entities that exist in their immediate environment – for instance, establishing a route towards the location of a point of interest (e.g. underground station). Currently, several stakeholders offer access to vast volume of data that can be employed by end-users at no cost. Furthermore, by processing personal user information (also regarded as context), such as age and gender, real-time social collaboration between users may be supported – for example, exchanging text messages or engaging in a social game. A simple working definition of context is that it is any information that can be used to characterise the situation of an entity (Dey et al., 1999). Therefore, several types of context are already available for processing in a mobile device, such as the user's calendar, which can be used to provide information about his or her activities.

However, the visualisation and interaction methods, which have been integrated to applications that try to satisfy such diverse user information needs through the use of mobile devices, have been fairly narrow, especially when compared to desktop alternatives. Early devices could represent context only in textual forms, without being technologically capable of providing interactive environmental representations. Newer devices started to make use of two-dimensional map interfaces in order to depict the user surroundings. Although this has been a positive leap, several issues surfaced regarding the visualisation and interaction features of such systems. Only recently, we have started observing mobile context-sensitive applications that utilise advanced user interfaces which not only try to achieve good utility, but also try to enhance the user experience. These interfaces are capable of representing real-time context in various forms, as well as for depicting the user's surroundings and any potentially relevant entities, in a virtual environment. These entities can be remote objects or other users which the local user may wish to interact with and, therefore, a goal of the interface is to support this task. There are several types of such environments (Milgram and Kishino, 1994) which are capable of combining artificial and real information in a single shared space, each one presenting distinct characteristics on the development and use of the system. In this project, we will explore virtual and augmented reality as the medium to blend diverse information types. It is against this background that a research problem surrounding the project presented in this thesis started to emerge. It could be summarised as follows:

"Does seamless integration of information spaces with real-time contextual information sensitivity contribute to the satisfaction of a mobile user's information needs?"

By considering the research problem domain presented above, several interrelated Research Questions emerge, which are explored in this thesis. They are presented at this at this point, as they naturally occur in the sequence of stages of a research process:

1. *"Which are the users' expectations from a system that would answer the key requirements as described in the research problem: the context awareness, the ubiquity of access to information and the real-time response to the specific information needs? In particular, the requirements in terms of visualisation, interaction and collaboration features."*

Our intention was to produce an extensible framework that can easily support supplementary, application-relevant functionalities. Therefore, in order to satisfy several user information needs, the proposed framework should not serve a specific application domain.

2. *"Which are the users' main expectations of a mobile mixed-reality context-sensitive framework that may have a wide range of ubiquitous applications in areas such as entertainment, marketing, wayfinding and navigation?"*

Certain interface paradigms (i.e. virtual and augmented reality) were selected in this study, with distinct characteristics, to investigate how effectively do they influence the decision-making process of a user. The issues that facilitate this process include cognitive abilities, system design, applied algorithms, technological solutions and usability suggestions amongst other. This leads to the next Research Question which can be formulated as:

3. *"What technical specification (to be considered in the design and implementation process) would provide effective mobile context-aware services that answer the user requirements as gathered in the first stage of the research (related to Research Question 1)?"*

The Research Questions as formulated above suggest that it is necessary to adopt a scenario where ubiquitous access to information is ensured in order to evaluate whether the proposed features support the task accomplishment and how easy and intuitive it is for the user to operate the proposed system. Out of the several possible application domains (as mentioned in the Research Question 2) the selected domain for the implementation was urban navigation. It is a context with which the users are widely familiar and as such it could assist in the examination of the design issues and in evaluating various interface solutions by distinguishing effectively the features of different offered paradigms, as well as consumer devices and information metaphors. Furthermore, the adoption of this scenario can present the ability of the framework to successfully operate in unfamiliar environments. The next Research Question is presented below.

4. *"Is there a difference between AR and VR for urban navigation, primarily, in terms of user performance and, secondarily, in terms of user experience?"*

This project was conceived to design and deliver the system that would address the gaps in the current offerings. In more detail, these are (i) the richer geo-referenced graphical interfaces representing the user's surroundings and the dynamical connections between and (ii) the acquisition and distribution of contextual information ubiquitously and in real time. To achieve this enrichment and enhanced context this project has sought to conduct research into the representational and communicative aspects of context-aware services, and develop and evaluate alternative user-centred approaches that attempt to make information more relevant, appropriate, comprehensible and intuitive for mobile individuals. To accomplish this goal, we had to follow an agile development approach. The first phase of the project was the requirement acquisition phase which involved an analysis of the responses provided to a survey questionnaire by potential end-users. These responses contributed to the requirement list which was also influenced by the results of other relevant research projects. The development phase focused intensively in implementing the requirements to an innovative, context-sensitive, mixed reality framework. Nevertheless, invaluable suggestions about several associated concepts sprung out of an expert user evaluation which took place when the development of one of the two interfaces (i.e. VR) has completed and collected feedback from 8 individuals.. The collected user feedback informed the following development phase, which introduced the novel context-sensitive AR interface. The last part of the process introduced an Extensive Evaluation test that took place in the field with 23 regular specialist users. This summative evaluation task objectively examined user performance between two conditions (AR vs VR) in an urban navigation scenario and also produced subjective responses to questions about the framework's usability features.

This has required an interdisciplinary approach that has drawn upon Information Science, specifically Geographic Information Science, Information Society, Virtual and Augmented Reality, Mobile Computing, particularly Context-aware Services, and Information Communication Technologies. This section provided a brief description of the research background, the key questions, the overall character of the research method and development processes, the research outcomes, as well as the domains in which the research lies. The following sections of this chapter explain in full detail the genesis of the problem, as well as the aims and more specific objectives that have been investigated. The Contributions Chapter presents the measurable contributions of the research and the contribution made to knowledge.

1.2 Context Framework

The era we live in is often referred to as Information Age. With ever expanding mobile technologies, Location Based Services (LBS), which integrate the location information with the potentially useful and interesting content for the user, have already been successfully introduced and started to generate significant profit for service providers. A report (Fuente et al., 2005), which presents the findings of a mission organised by the former U.K. Government's Department of Trade and Industry (DTI), shows that mass-market consumer applications based on Global Navigational Satellite System (GNSS) products are seeing significant growth, especially in developed, in terms of the use of technology, countries.

Considering the European Union (EU) initiative to develop an independent satellite positioning system (i.e. Galileo), inter-operable with the existing Global Positioning System (GPS), we can deduce that, in the future, location-based services will see further development and will become part of everyday interactions. Consumers in continental Europe have not adopted this kind of services on a significant scale and still resort to the use of basic mobile communication methods for exchanging contextual information. Initially, the operational purpose of mobile phones was limited to verbal communication, but the progress in related technological fields has produced new services for other means of communication, such as instant text messaging and robust devices like GPS-enabled Personal Digital Assistants (PDA). Currently, consumers expect easily integrated solutions to evolve, with rich content, embedded privacy & security mechanisms, advanced functionalities and well-defined business models. Japan has already produced various successful applications based on explicit mobile platforms, mainly because of the wide availability of advanced hardware solutions and infrastructure. Unfortunately, mobile devices and services with built-in positional awareness features have only recently started to become widely available in the European market. This is contradictory to the use of technology that is made on other platforms, like the cumbersome desktop configurations.

Competition in the telecommunication market is increasing rapidly and network service providers are looking for novel types of services, which will enable them to differentiate from their competitors. Most commercial geo-referenced applications, which have evolved, focus on certain business domains, such as remote asset tracking and emergency management. Mike Short, Honorary President of the Mobile Data Association (MDA), argues that some of the reasons, which sustain this low rate of developmental progress, include the variable positional accuracy and the long interval that is required to obtain a position fix. Most important, however, is considered the fact that service providers have focused on advancing a wide variety of technologies and infrastructure, such as 3G/4G networks and videoconferencing solutions (Thomas, 2006), but not on the content. It seems imperative, now that the focus is placed on the area of ubiquitous operation, to design coherent mobile context-sensitive solutions, which would effectively provide advanced digital information services to their users.

According to the formal definition provided by the International Telecommunication Union (ITU), mobile applications are add-on software for handheld devices, such as smartphones and personal digital assistants (PDA) (ITU, 2011). This somewhat contradicts the concept of all LBS being truly mobile. Most LBS can hierarchically fit into two distinctive categories. Most common are those that reveal to the user the occupied position on earth, including additional information like orientation and interaction options. Less frequently and out of the scope of an average consumer, one can find applications that allow a 3rd party to examine positional and derived data of a remote subject, in real time. Out of these two types, only the second can always comply with the definition of a mobile application, because it is expected to exchange up-to-date context with a 3rd party, in real time.

1.3 Motivation & Background

One of the identified shortcomings of several context-processing applications is the inability to store the user's track logs and additional personal or location information, such as the description of selected Points of Interest (POI), in universally accessible space, for reproducing the supplied functionality at any time or for sharing it with other users. If this option becomes available, it will open up a completely new perspective, which can have intriguing consequences on the ways in which the service is being used. Technically, it is feasible to store contextual information and to manage and distribute it aptly - conforming to specific established rules and restrictions. An innovative concept can be thought of, in which the source-generated information to be recorded includes processing environmental variables as well as the user's details and other physical attributes, also taking into account time as a variable. The nature of such a system could be characterised as context-aware and to some extent, ubiquitous and pervasive (Chen and Kotz, 2000). An advantage of a mobile system like this is that it allows its users to retrieve real-time information from remote entities and offer its services according to their latest preferences or other up-to-date user context.

It is obligatory, though, for the system to respond to certain changes of the environment or user activity by augmenting the interface towards the user to reflect those changes. Discovering and taking advantage of changes, such as the proximity of another user, may trigger collaboration between the two parties and allow for other social interactions to take place. The probability of such interactions may increase, for example, when certain predefined social criteria are in place and both actors match them. If a certain scenario is integrated in the system, and both parties have the option to accept/reject it, the application could be used to initiate interaction and communication, on a peer-to-peer basis.

This communication structure requires an enabling middleware agent, which would operate between the users. Its operation does not have to be continuous, because the quantity of data to be processed would pose performance issues on non-hi-tech devices. The data that this agent should process includes the details of user profiles, the user spatial coordinates and temporal information, as well as meaningful representations of the surroundings. In order for a complex system like that to become useful for its users, the client interface should be able to promptly react and dynamically visualise the information. Mixed Reality (MR) modes of operation, which combine virtual and augmented reality techniques, in 2.5D and 3D, could be employed to accomplish this task. For the purposes of this project, we assume that advanced spatial information representation techniques are better for geo-visualisation, especially in unfamiliar environments when an external representation is required for effective wayfinding. One reason is that 2D representations, although efficient and popular, are fairly limiting regarding the volume of information that they can present, compared to 3D representations that carry much more information. This happens because in 3D models, there is a volumetric representation of space instead of a flat one and also because the modelled entities appear more realistic in comparison to 2D which are symbolically represented. Furthermore, in a virtual environment, users have the potential to select various directional observation perspectives that can complement their spatial knowledge about the surrounding area and they are not restricted only to a bird's eye view. Navigation in 3D environments appears more realistic and, particularly for the young user, invites for deeper exploration of the represented area. The simulation of the wayfinding tasks becomes more fulfilling, especially if we consider that movement takes place by accumulating additional Degrees of Freedom (DOF). Using controlled overlaying of objects (e.g. images, text or other) over the imagery of the real world, allows for the provision of more information in a given display area, without increasing the complexity of the interface (Papakonstantinou and Brujic-Okretic, 2006), which can have an impact when dynamically altering the details of a modelled entity.

This work is placed in the context of a specific application developed at the Department of Information Science of City University London, in 2005: *Mobile Virtual Reality Interaction with Global Positioning Systems* (Papakonstantinou, 2005), as the MSc dissertation of the author. The main aim of this study was to familiarise with the technologies and methods used in the field, with a view to creating a location-aware system, which works in real-time and in which the changes are dynamically reflected on the user interface. The preliminary work concentrated on exploring this concept and the limited technical developments took place on a laptop computer, so as not to be restricted by the early mobile device limitations. The advantage of enhancing and transforming the system to its mobile version is that it can produce an application, which is technically advanced and challenging, but also actively addresses a number of cognitive and social issues involved with its exploitation. This is accomplished by integrating, in a straightforward way, a smartphone, certain sensors and a selection of information visualisation environments. If this approach was implemented at a commercial level, it could significantly extend the traditional LBS approach in a way that it would provide truly ubiquitous operation and advanced user interactions leading to collaboration and advanced information retrieval capabilities. This can be demonstrated by suggesting a number of potential, commercially viable, location-based and in extent context-aware applications, which can be developed by utilising the framework.

In this project, the development phase produced a software tool, which is considered as a leading component to the main theme of the study towards a PhD degree. It is based on a client-to-client application model and was implemented using native programming languages. It is divided into 2 core parts and its main functionality is that the *Context*

Management entity polls external sensors and returns the obtained attributes to the *Information Presentation* layer, which in turn reproduces spatial information and interactions in a Virtual Environment (VE). The type of data that is collected and processed includes personal user, location and orientation information, which the system can communicate to a remote entity, in real time, that is capable of understanding and interpreting the custom networking protocol, which has been developed for exchanging real-time context. Although it can be associated with a server that offers advanced functionalities, the development has focused on client-side enhancements towards better user interaction and adoption of the system. The program has undergone considerable enhancements within its lifecycle, in response to the research findings.

In the core literature that examines research in the fields of *Context-awareness*, *Mobile Computing*, *Ubiquitous Service-composition* and *Human-Computer Interaction*, we observe similarities between the secondary goals and the perspective that has been adopted, in terms of accurate information provision towards the participating entities or users. Furthermore, we have noticed that in-context technologies and new research fields have emerged, which focus on the combination and satisfaction of issues found in all of these topics. For instance, the use of LBS is currently meeting increasing acceptance. Adoption by end-users and integration in their daily activities happens because these applications can provide accurate services, based on spatiotemporal constraints and in a beneficial way that can satisfy high-level user requirements. An important factor that contributes to the diversification and lack of evident interconnection between the research areas is the absence of unified hardware and software platforms, which could support further exploration and development.

Experience has shown that designing, implementing and deploying mobile-GIS systems that make use of real-time context requires knowledge from diverse research fields. The approach adopted in this project is similar to rapid prototyping in that it pursues continuous development and evaluation in a loop, which provides feedback to the development process. Resolving which elements and methods to embed in the framework and for which reason, is a product of the research undertaken so far, together with the technical experience that was gained through practical work. By utilising this approach we aim for component reusability and expandability, which means that additional features can be supported in forthcoming applications.

The development has advanced through small progressive steps. The process was closed and the loop included rapid prototyping, evaluating and improving, according to user and expert feedback. Furthermore, all iterations of the loop incorporated a distinct requirement acquisition phase that required user participation. As Houde and Hill mention in their article, there are some problems that affect this approach (Houde and Hill, 1997). Interactive systems are complex and users experience the combined effect of the interrelated features. Thus, every aspect of the system must be designed (or inherited from a previous version) and many features need to be evaluated in combination with others. Moreover, Houde et al. note that prototypes provide the means for examining design problems and evaluating solutions, by identifying the most important questions and requirements. This means that our series of prototypes should efficiently describe the implemented features, through transparent interfaces, relevant to a particular implementation. With such a rationale, the system should be able to present the role its functionality will play in the usability domain, including the social implications and the novelty of its operational abilities, through the new techniques that have been introduced to enable such functionalities. Ultimately, the results of this analysis will improve the requirements specification and influence the design of the following prototype version. The aim of the model presented in Figure 1.1 is to visualise the focus that is explored by the prototype developed during the course of this project. The position of the marker shows that the primary focus is on the implementation, in order to explore the distinct technologies that need to be utilised in order to bring our research product into life. The secondary objectives that our prototype intends to fulfil are the establishment of the context of the artefact's use (i.e. Role) and the simulation of the user experience (i.e. Look and Feel).

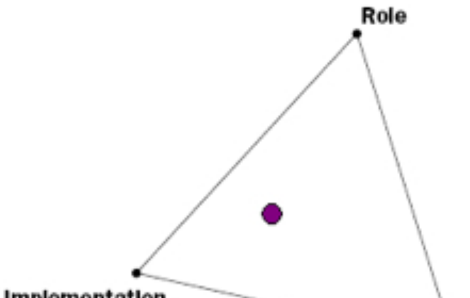


Figure 1-1: The model adopted during the prototyping process (Houde and Hill, 1997)

Fallman writes that *"(...) in design-oriented research, the knowledge that comes from studying the designed artefact in use or from the process of bringing the product into being is the contribution, while the resulting artefact is considered more a means than an end. It should include 'problem setting' as an important part, the possibility of exploring possibilities outside of the current paradigm*" (Fallman, 2003).

1.4 Assumptions & Hypotheses

This work is based on a series of assumptions that, during the course of research and development, will be tested for validity. The approach followed is described below.

Currently, mobile technologies have evolved to a point that renders the service composition and the service discovery experience easily accessible for most end-users. If we combine the features of these e-services with the current context-sensing technologies (e.g. GNSS) and the advanced features of current mobile devices, a basis for context provisioning services is obtained. The first hypothesis is that classic context-sensitive services, and more specifically LBS, can be transformed into a platform for pervasive systems through applications that integrate personalisation and user-profiling mechanisms, as well as precise application logic.

Further processing of spatial, temporal and user data could provide application reactivity, which is an indispensable component of pervasive systems. In order to commercially attract wider audiences, certain rewarding extensions should also be implemented. Initially, privacy restrictions on location and personal information should be applied in order to render the system architecture trustworthy towards potential customers. Additionally, Mixed Reality (MR) representations of information and the surrounding environment should also be applied for more efficient information visualisation and for improving the interaction between the actors. We assume that the solution to this concept would be an underlying framework enabling collaboration and supporting advanced communication and even entertainment services for and between mobile users – this is the second hypothesis. Collaborative information visualisation (Frekon and Nou, 1998) is the method that makes this notion possible. It is expected that the adoption of such a system will be higher than the average demand of current off-the-shelf products. An analogous working m-Commerce architecture (Pennington,

2001) has been published, which could have applications in several areas and activities of everyday life.

As we have seen, a subset of mobile computing that has been the focus for much research and commercial endeavours over that last decade has been the field of context-aware services. These are mobile applications where information retrieved by a device user varies dynamically according to their spatial, temporal and personal behaviour. The most common approach to visualise significant real-time information associated with both the user and relevant information objects, like points of interest, has been two-dimensional representations of the environment (i.e. digital maps). However, users often find it difficult to associate the allocentric visualisation perspective of maps with their actual experiences that take place in the natural world because maps are designed for a detached overview (i.e. allocentric) rather than a self-referential personal view (i.e. egocentric), which poses new challenges for the representation of geographic information on mobile devices. The assumption is that mixed reality interfaces on mobile devices can provide supplementary representations (e.g. egocentric, oblique), which can augment an individual's view of the physical surroundings with virtual information more effectively – the third hypothesis.

This research project aims to bring together the research theories, methods and tools from two high-level research domains. These domains are Information Science, with particular aim on the societal and user’s perspective, and Information Communication Technologies (ICT). In more detail, the project tries to address the technical and the users’ challenges in the field of real-time context-aware information systems, with particular emphasis on mobile MR applications. In the information science domain, the goal is to identify the requirements, which influence the user expectations and the technical issues, which must be accommodated in order for the framework to become effective. On the ICT domain, the goal is to tackle issues relevant to: the discovery of remote real-world entities, ubiquitous information retrieval from these entities according to the user’s preferences and other contextual information, privacy of communicating sensitive information and the development of certain advanced User Interface (UI) paradigms, capable of sustaining and representing natural world behaviour.

1.5 Aims & Objectives

The research issues covered in this project cut across several research fields, ranging from mobile networking and information architecture to information retrieval and visualisation on mobile devices. This endeavour aims to embrace both the high-level system architecture issues involved with contextualising and managing information, and the low-level implementation methods and techniques.

Aims

1. To propose a technological framework – system architecture and functionality to support peer-to-peer interaction and context-sensitive information retrieval using mobile devices and networks;
2. To develop an application which will combine geo-referenced, 3D content with contextualised user information, in real-time, to promote collaboration and interaction between mobile clients;
3. To examine the framework against the technical as well as social aspects of what is required of an ubiquitous, context-aware application, using pervasive scenarios as a case study;
4. To evaluate the utility and certain usability aspects of this context-sensitive, integrated system with currently available location-based services, to identify possible prospects for commercialisation.

The goal of this project is to design, develop and operate a mobile system with minimum standardised configurations, easily customisable for commercial exploitation.

Objectives

1. To acquire high-level, user-related requirements through modelling user behaviour and to discover connectors with the technical aspects of the system;
2. To identify and include in the implementation low-level, technical requirements and specifications that can describe the full functionality;
3. To design and develop a mobile data communication protocol for cellular and wireless networks that will be able to transfer user and location context in real time;

4. To design and develop a flexible archive system, which will dynamically store the user's position, orientation and remote entity contextual variables. It should be concurrently accessible by several users;
5. To implement a robust context-aware, location model, which will be the fundamental element of the geographic component;
6. To develop user profiling and data management mechanisms, with emphasis on enabling peer-to-peer collaboration and interaction;
7. To apply privacy and security restrictions, which will govern user communications and exchange of information;
8. To develop and combine various level-of-detail interfaces for supporting ubiquitous interaction with the environment and selected elements;
9. To enhance the information visualisation framework of a mobile device, in order to support collaboration between actors and stakeholders;
10. To formulate knowledge-based scenarios, which can be integrated to the software environment, to test user interactions;
11. To evaluate certain usability aspects of the framework, especially in terms of information visualisation options, in order to enhance user decisions and their application in ubiquitous scenarios.

1.6 Contributions

This is the summary of the contributions made throughout the research conducted on the project aimed at designing and implementing a dedicated system architecture to enable the user of a mobile device/service to access and use relevant contextual information with respect to their previously formed personal profile.

The underlying concept, and a higher-level goal, has been to facilitate social communications and interactions between users *on the move*, using mobile devices and innovative integrated services. It is our belief that it could contribute to enabling mobility of information – an issue of integral interest for both social science aspects of the matter and for the underlying technological challenges.

The strength of the project lies in a holistic approach to the design and implementation of such a system – providing sound technical solutions, but never losing sight of the user's perspective and cognitive and social aspects of the network service, which it could ultimately result in.

The main contributions of this research comprise: (i) theoretical and methodological contributions to the knowledge in the area of context-aware systems, and (ii) contributions in the form of innovative technical solutions to facilitate ubiquity of access to relevant, multi-layered, context-sensitive information to the user – in real-time, in an automated fashion, on a mobile device. A detailed description of both aforementioned types of contribution is presented below.

Contribution to Knowledge

- The integration of (i) *a context-aware software system*, (ii) *two distinct interface paradigms*, (iii) *a user privacy and personalisation scheme* and (iv) *a dedicated context-sharing communication protocol* into a distinct mobile framework. The framework bears significance beyond the scope of the project and beyond the remits of the technology framework in that it presents a pre-requisite for the operations of mobile devices in unknown environments, which has not yet been achieved in mobile services provision;
- The fusion of several mathematical algorithms into the design of a distinct approach, and its software implementation to solve the calibration issue and association of remote information entities with virtual overlays on live camera feeds, in real-time. This is a key issue for any mobile system providing contextual information because ubiquity of access to information and real-time response to specific information needs is still a challenge. This approach comprises several stages: (i) *tracking*, (ii) *registering*, (iii) *camera modelling* and (iv) *scene rendering*. Certain transformation matrices are used to translate coordinates between the *world*, *camera* and the *image* coordinate systems in order to accurately identify the point of the user's current interest translated to its on-screen representation. The solution is novel in the context of the technology framework as presented in the thesis;
- A comprehensive system development methodology (i.e. *analysis*, *design*, *implementation* and *evaluation*) that enhances the utility and the mobile users'

experience of a mixed reality system which processes context in real time and dynamically offers information services to its users in situ. This contribution is novel from the Information and Computer Science perspective in the context that the resulting product is customisable to support a variety of application domains and that it operates ubiquitously in environments without any prior training.

The deployment of the novel architecture in a prototype application provided another significant contribution, since researchers and practitioners can be equipped with a tool to make comparisons and evaluate their ideas.

- An intuitive mobile navigation application. The application is novel not only because it facilitates two distinct visualisation and interaction interfaces, each one providing distinct functionalities, but because it is also cost-effective and can work at any part on the world by processing real-time contextual information.

Technical Contributions

In terms of the design and the development of the underlying technological framework to answer the Research Questions, the contributions could be summarised as follows:

- Design and implementation of a fully functional application to proactively aid the user in acquiring context-relevant, geo-referenced information on a mobile device, in real time.
    - Acquiring the functional requirements and proposing a working system design contributes intellectually to the methodology of relevant projects and future research. Potential system developers will have a starting point to cope with the issues examined in this project. This can be reflected by the iterative engineering techniques that have been adopted.
- The developed application is multi-faceted and features the following key functionalities:
    - Acquires position and orientation data from the sensors on-board a device (i.e. GPS and digital compass) and couples it automatically with a representation of the immediate environment, giving the user greater sense of the surrounding space.

    - Provides to the users the option to explicitly input their preferences, through the provided interfaces, in order to form their profile, which will subsequently be used for informing system's intelligent choices, where appropriate.
    - Provides the user with the choice to select various Level of Detail (LOD) interfaces, depending on their current needs – from 2D/3D photorealistic models to highly stylised environment representations.
    - Allows for on-demand interactions with other users on a one-to-one basis to take place, in order to facilitate social communication by using collaborative information visualisation through the adopted interfaces.
- An Augmented Reality (AR) mode of operation is fully developed on a mobile device providing the following features:
    - Overlaying additional information on selected Points of Interest (POI) captured by the on-board camera, in real time.
    - The operation is independent on any markers or predefined settings in the environment, thus rendering the application ubiquitous. This approach is novel and advantageous over the majority of other similar applications that do require calibration and registration with the real environment prior to proper functioning of the service.
- The developed system architecture allows for two modes of operation:
    - As a standalone application (*Aura*) that supports full functionality, or
    - As a set of reusable components that could be plugged-in to another host application - most notably, mobile web browsers, which makes the system platform-independent when running in that mode.

1.7 Collaboration Statement

This project would not have progressed if collaboration with other academics and scientists had not materialised. This section lists the publications, which have been produced by collaborating with other researchers involved in this research project.

[1] S. Papakonstantinou, "Mobile Virtual Reality Interaction with Global Positioning Systems," in *Department of Information Science, School of Informatics*. vol. MSc in Information Systems & Technology. London, U.K.: CITY University London, 2005, p. 118.

[2] S. Papakonstantinou and V. Brujic-Okretic, "Design and implementation of an advanced generic location-aware engine," in *Proceedings of the 14th Annual Conference in GIS Research UK (GISRUK'06).* Nottingham, U.K.: The University of Nottingham, 2006, pp. 380-386.

[3] F. Liarokapis, D. Mountain, S. Papakonstantinou, V. Brujic-Okretic, and J. Raper, "Mixed Reality For Exploring Urban Environments". In *Proceedings of the 1st International Conference on Computer Graphics Theory and Applications (GRAPP'06).* Setubal, Portugal: Insticc Press, 2006, pp. 208-215.

[4] F. Liarokapis, V. Brujic-Okretic, and S. Papakonstantinou, "Exploring Urban Environments Using Virtual and Augmented Reality". *Journal of Virtual Reality and Broadcasting (JVRB).* vol. 3, February 6th, 2007, pp. 1-13.

[5] C. Gatzidis, S. Papakonstantinou, V. Brujic-Okretic, and S. Baker, "Recent Advances In The User Evaluation Methods And Studies Of Non-Photorealistic Visualisation And Rendering Techniques". In *Proceedings of the 12th International Conference on Information Visualisation (IV'08).* London, U.K.: South Bank University. IEEE Computer Society, 2008, pp. 475-480.

[6] S. Papakonstantinou, and V. Brujic-Okretic, "Prototyping a Context-Aware Framework for Pervasive Entertainment Applications". In *Proceedings of the 1st International IEEE Conference in Serious Games and Virtual Worlds (VS-Games'09).* Coventry, UK: Coventry University. IEEE, 2009, pp. 8-15.

[7] S. Papakonstantinou, and V. Brujic-Okretic, "Framework for Context-Aware Smartphone Applications". In the *International Journal of Computer Graphics* (*The Visual Computer).* vol. 25(11). Springer, 2009, pp. 1121-1132.

[8] D. Mountain, F. Liarokapis, S. Papakonstantinou and V. Brujic-Okretic, "Architectures for location-aware, mobile mixed reality systems: a case study from the LOCUS project". In the *Computer Graphics and Applications (CG&A).* vol. 25 (Special Issue on Mobile Graphics). IEEE, 2007. [In preparation for re-submission]

[9] S. Papakonstantinou and V. Brujic-Okretic, "Novel Interface Paradigms on Context-Sensitive Mobile Devices". In the *International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI).* ACM Press, 2012. [In preparation]

[10] S. Papakonstantinou and V. Brujic-Okretic, "Prototyping Mobile Context-Aware Interfaces for Pervasive Services". In the *International Conference on Pervasive Services (ICPS).* IEEE Computer Society Press, 2012. [In preparation]

[11] S. Papakonstantinou and V. Brujic-Okretic, "Augmented Reality for Social Interaction Using Intelligent Mobile Devices". In the *Journal of Location Based Services (JLBS),* vol. 5, Taylor & Francis, 2012. [In preparation]

[12] S. Papakonstantinou and V. Brujic-Okretic, "Evaluation of context-aware user interfaces for ubiquitous scenarios". In the *Journal of Location Based Services (JLBS),* vol. 5, Taylor & Francis, 2012. [In preparation]

1.8 Organisational Structure

Chapter 1: Introduction

This chapter constitutes an introduction to the theme of the research undertaken in this project. It introduces the motivation that has driven the research, followed by the principal aims and objectives of the project. Furthermore, the measurable contributions of the research and the contribution made to knowledge in the course of these investigations are presented. The chapter concludes with a brief presentation of the organisation of the material and an overview of the document structure.

Chapter 2: Literature Review

The second chapter concentrates on the discussion of the fundamental topics, which underpin the theoretical background of the research. It covers a broad range of subject fields like mobile computing technologies that enable LBS, Mixed Reality for pervasive applications, as well as personalisation and privacy mechanisms that support collaboration. While every effort has been made to make sure that this study is as comprehensive as possible, some background work may have been omitted or placed in other chapters of this report.

Chapter 3: User Modelling & Requirements Engineering

This chapter presents the process, which has been adopted in order to realise the design of the framework architecture, and the milestones that were the driving force towards that direction. The issues discussed in this Chapter, and specifically in Chapter 3.3 and its subchapters, present several issues which can directly influence the outcome of the 1st and 2nd Research Questions of this project. Forming models of potential users based on their behaviour and familiarities, exploring relevant cognitive issues, modelling the immediate user environment and finding out how pervasive computing can mix with advanced visualisation techniques to promote collaboration are the user-related issues examined in this chapter. Furthermore, producing high-level framework models, which can be used to relate the previously acquired user challenges to the achievable system functionalities, is the last point described in this chapter.

Chapter 4: Prototype Development Methodology

In this chapter, the reader can discover specific issues that affected the analysis and design of the developed system. The framework architecture, which includes the hardware and software components that were put in use, is described here. Examining the available technological products, standards and protocols has produced the hardware specifications required for the operation of the proposed system. The selection of the software components and platforms that sustained the development efforts is also presented in this chapter. Furthermore, the desired system functionality, which is expressed in functional and non-functional requirements, is laid out. The chapter concludes with the delivery of the system design that illustrates certain core aspects inherited and enabled by the realisation of the system architecture.

Chapter 5: Applying Context-Awareness on Mobile VR & AR Interfaces

In this chapter, we introduce a context-aware application, *Aura*, designed and implemented on a mobile device platform. *Aura* can adapt its functionality according to context changes related to the user and the environment in real time. The ability to visualise contextual information through a variety of interfaces is the main feature that promotes interactivity. The implemented solution includes a scalable 2D map-based environment, a detailed virtual 3D engine and a photorealistic image-based augmented reality interface. The application queries the coupled sensors to identify modifications in context, integrates the output and adjusts the mode of interface to be employed, as requested by the user. The sequence of operating modes can vary, depending on the context and/or user's preferences. Use cases describing navigation models have been applied and more complex pervasive scenarios have been explored. The proposed framework aims for truly ubiquitous operation that will enable novel collaboration patterns to evolve, which in sequence may trigger social interaction based on proximity and user preferences. By implementing the requirements that were presented in the previous chapter, we can influence the development of the technical specification which is required for achieving the necessary results for the 4th Research Question, illustrated in Chapter 1.1. The approach that was selected to implement the requirements in the proposed framework is presented in this Chapter.

Chapter 6: Potential Context-Sensitive AR Applications

This chapter presents a detailed analysis of the potential applications that may evolve by customising the developed framework. Several applications have been identified during the course of the project, but this section describes those, which can reflect certain capabilities of *Aura*. The chapter also presents possible solutions for commercialisation, which have been triggered either by interrogating potential users, by research on this field, or by actively getting involved in the promotion of the framework features to potential investors (i.e. stakeholders). A potential commercialisation model can be found in Appendix XIV of the report.

Chapter 7: End-User Evaluation and Results

This chapter presents how the evaluation phase of the developed framework was set up in order to measure the effects of the acquired requirements. Several evaluation cycles of variable extent were accomplished during the course of the project. This section will present the scenarios that have been selected in order to evaluate the framework and describe in more detail two evaluation tasks. The first, Preliminary Evaluation took place during the first half of the research span while the second one is an Extensive

Evaluation of the system performance and usability aspects that occurred at the end of the project.

Chapter 8: Discussion of Evaluation Results

This chapter provides a discussion of the topics, which have been explored during this research project. The chapter commences with a presentation of the results obtained by the preliminary evaluation cycles. Following next, the reader will find a discussion of the extensive assessment's results. The discussion of the results that have been produced by both evaluation processes is presented in the context of the framework's usability and functionality features.

Chapter 9: Conclusion

This chapter provides a summary of the topics that have been explored in this research project. The initial aims and objectives are examined, to verify how they have been satisfied for the purpose of the research. The sections that follow present the overall contributions made through this research as well as a critical analysis of the results, including the identified limitations and the recommendations for future work. These recommendations will allow overcoming the identified limitations and could assist in the production of a tangible commercial solution out of this research project.

Chapter 10: References

The core information sources that have been injected in the conceptual progress and the main body of this document are presented in this chapter. Their style conforms to the *Harvard* referencing system and they are sorted in alphabetical order.

Chapter 11: Bibliography

This chapter presents information about the sources of documentation, mostly found online, that were considered relevant and useful during the course of the research. They are presented in an *Annotated* style and they are sorted in alphabetical order.

Chapter 12: Appendices

This chapter contains relevant and explanatory information about several aspects applicable to this project.

2 Literature Review

The second chapter concentrates on the discussion of the fundamental topics, which underpin the theoretical background of the research. It covers a broad range of subject fields like mobile computing technologies that enable LBS, Mixed Reality for pervasive applications, as well as personalisation and privacy mechanisms that support collaboration. While every effort has been made to make sure that this study is as comprehensive as possible, some background work may have been omitted or placed in other chapters of this report.

2.1 Mobile Context-Aware Computing

One of the goals of modern Human Computer Interaction (HCI) research is to make user interaction with computing devices easier and to provide means to trigger meaningful actions with them. A system architect has to conceive the feasibility of a system by taking into consideration the process of sensing implicit contextual information from the environment and successfully fusing it to the system. This way, the users can decide which information is relevant to them, in order to manage and take advantage of it more efficiently. The need for precise contextual information increases when we envisage systems that work in mobile environments. Users need to dynamically interact with other people and objects, which makes the importance of real-time context more evident. Currently, a set of technologies has formed the infrastructure, which enables ubiquitous access to desired information services. Portable interconnected computers have made the distribution of contextual information possible to interested parties. Furthermore, while the size of mobile devices is shrinking, their processing power and embedded features are reverse-proportionally increasing. Additionally, the operational effectiveness and enhanced bandwidth of wireless data communications protocols (e.g. GSM/UMTS, WLAN and Bluetooth) can provide interaction between devices, from anywhere and at any time. "*Such mobile-aware applications will be more effective and adaptive to users' information needs without consuming too much of a users' attention, if they can take advantage of the dynamic environmental characteristics (...)*" (Chen and Kotz, 2000).

2.1.1 Context Definition

Many researchers have tried to provide a clear definition of what context is. Some definitions provide examples of which the elements that constitute context are while others supply synonyms for it. In order to select which types of contextual information to include in our framework and to create abstraction mechanisms for effective use, we need to consider every definition. Initially, Schilit et al. split the important elements of context into three categories and later Dey et al. refined them in a clear structure. Descriptive sets of measurable elements of the user environment are presented in the following list (Schilit et al., 1994) (Dey et al., 1999):

1. *User context* consists of user profile, location, orientation, nearby people, social situation and activity;
2. *Computing context* consists of processor power, input devices, visualisation resources, network connectivity, bandwidth and computing costs;
3. *Physical context* consists of light, noise, weather conditions, temperature and traffic;

The list can also include the following categories:

4. *Time context* consists of date, time and season;
5. *History context* can hold values from all previous categories, which have been documented across a time span.

More precisely, Dey defines context as "*any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*" (Dey et al., 1999). Furthermore, Schmidt et al. provide the following definition of what context-awareness is "*(...) knowledge about the user's and IT device's state, including surroundings, situation, and, to a lesser extent, location*" (Schmidt et al., 1999) and models it. The three-dimensional space that is used by Schmidt to describe contexts is presented in Figure 2-1.

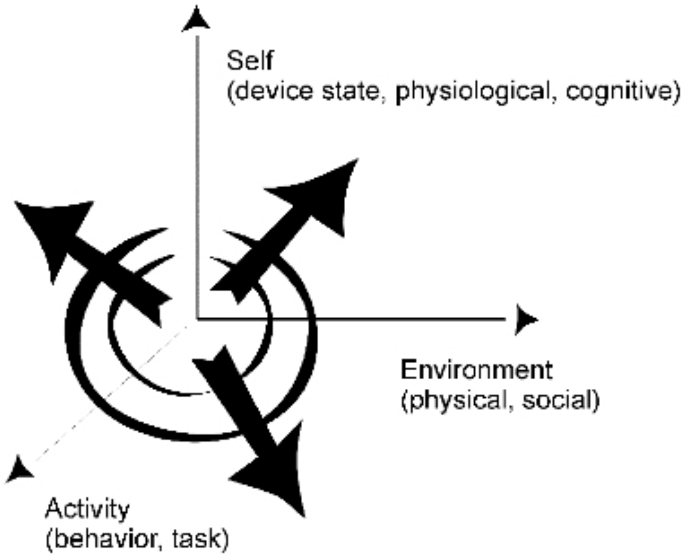


Figure 2-1: 3D Context Model (Schmidt et al., 1999)

In the late 1990's, Salber et al. provided a supplementary definition of what can be considered context (or environmental information), in the context of an application's operating environment. It is the information that can be sensed by the application, which may include the *location*, *identity*, *activity* and *state of people*, *groups* and *objects*. They supported that context can also be related to *places*, such as buildings and rooms, or the *computing environment* variables, and can be used to describe the capabilities of nearby *resources* (Salber et al., 1999). Finally, in the same publication, they describe the major issues, which make the use of context difficult in most applications. Namely, these are: a) the need to utilise *uncovential sensors*, b) the need for *context abstraction*, c) the *multiple distributed and heterogeneous sources of context* and d) its *dynamical changes* (Salber et al., 1999).

In the case of mobile computing, the most descriptive definition of context comes from Chen et al., who additionally divide it into active and passive. "*Context is the set of environmental states and settings that either determines an application's behaviour or in which an application event occurs and is interesting to the user*" (Chen and Kotz, 2000). *Active* context influences the behaviour of the system and *passive* context is relevant but not crucial for the operation of the application.

A very interesting perspective on the available types of context comes from Göker et al. They support that "*the type of context depends on the entity or actor the context is intended for*" (Göker et al., 2009) and provide some examples on how should context be characterised according to the situation that is being described. One of the examples that they provide is that contexts that describe a *user's situation* should be named *user context*.

2.1.2 Context-Aware Computing

Apart from sensing and retrieving contextual information, managing and presenting it to the user is an important objective for system designers (Dey, 2000). Context-aware computing demands applications that are capable of operating in highly dynamic environments, by placing minimal demands on user attention (Henricksen and Indulska, 2004). According to Schilit et al. applications that fulfil these requirements are based either on a single or a combination of the following processes: *proximate selection* (i.e. entering locus and selection of objects), *automatic contextual reconfiguration* (i.e. manipulating components and connections between them), *contextual information and commands* (i.e. predicting actions based on situation) and *context triggered actions* (i.e. to specifying how systems adapt) (Schilit et al., 1994). Pascoe extends this categorisation with a set of core capabilities that can be used as a vocabulary to identify and describe context-awareness independently of application, function, or interface. The taxonomy that is proposed includes *contextual sensing*, *contextual adaptation*, *contextual resource discovery* and *contextual augmentation,* which is used to associate digital data with the particular context that it is related to (Pascoe, 1998). Dey took under consideration both taxonomies described earlier and produced his own requirements, which context-aware computing application designers have to cope with (Dey et al., 1999).

i. *Presenting* information and services to a user;

ii. *Executing* a service automatically;

iii. *Tagging* of context to information for later retrieval.

Finally, based on his definition, Chen grouped context-aware applications into two classes (Chen and Kotz, 2000).

- *Active*: an application automatically adapts to discovered context, by changing its behaviour;
- *Passive*: an application presents the new or updated context to an interested user or saves it for later retrieval.

Literature of Existing Applications

A large number of context-sensitive applications have been developed for research or commercial purposes. It would be extremely difficult to review and compare the vast majority of them, as they do not utilise the same types of contextual information and because their functionality is not particularly relevant to this project. The most frequently sensed type of context is *location*, which limits the value of other types of information in these applications. The use of context *history* and *activity* was very interesting to observe, in the few applications that were embedded. Some applications that treated the user as a source of context were particularly inefficient. One such example was *Campus Aware* (Burrell et al., 2002), which was found to be distracting and confusing for the user. Its main functionality was to behave as a campus tour guide, but also to collect and annotate user experiences about previously visited places. On the other hand, five publications that discussed and evaluated a number of existing context-sensitive systems were found to be extremely useful for the research at hand. The research method in each publication investigated the associated applications from a particular perspective. Dey et al. reviewed 14 applications (Dey et al., 1999). Thirteen of them were using location and only 4 were considering activity context. Moreover, 10 applications just presented information to the user, but did not trigger any distinct service, nor did they attach metadata to the information for indexed recording. Furthermore, a valuable reference was published by Chen and Kotz, which includes a review of 15 context-aware applications (Chen and Kotz, 2000). They particularly focus on the operation of mobile frameworks and on the type of context that they utilise (i.e. active or passive). In another survey that was published in 2000, Korkea-aho reviewed several context-aware applications and classified them in distinct categories. This classification scheme included applications that were conceived as *office and meeting tools*, *tourist guides*, *fieldwork tools*, such as archaeological assistant tools, *memory aids* and, finally, *frameworks* that can support context-aware applications (Korkea-aho, 2000). An important outcome of this survey was that the development cost of such systems was high due to their complexity because "*they tend to be resource hungry, since they usually need to do continuous monitoring or complex calculations*" (Korkea-aho, 2000).

Newer evaluations, published between 2005 and 2007, characteristically present the attempts that have been made to improve the development of these solutions and introduce additional types of context in better-formulated procedures. We are going to examine pervasive games, a distinct kind of context-aware applications, because playing desktop computer games decreases the users' physical activities and social interaction. Furthermore, it was found that they trigger only minimal interaction between users, by visualising artificial information. A fast growing trend in today's games industry is to enable complex physical and social interactions, while still utilising the benefits of mobile computer and graphic systems. An interesting publication, which examines a context-aware application that monitors the user's heart rate and promotes healthy physical activity, was published by Boyd Davis et al. (Boyd Davis et al., 2007). *'Ere be Dragons* also works in multiplayer scenarios making social interactions possible in the boundaries of a game. In the paper "*Pervasive Games: Bringing computer entertainment back to the real world*" (Magerkurth et al., 2005), the authors separate pervasive gaming into 5 categories and review 19 applications. *Smart toys* are traditional physical toys equipped with simple sensing technology linked to computer logic. *Affective games* capture how a player feels at any given moment and integrate this personal representation of context into a game. *Augmented tabletop games* do not serve as input to the virtual game logic but add the richness of social situation to the virtual domain. Their success can be attributed to the direct interaction and communication between the players. Following next, come *Location-aware* games, which are presented in the following paragraph and *AR games*. These are the most interesting for our research. Augmented Reality games are a variation of Virtual Reality games, which draw virtual objects into a real-world environment. Currently, research on AR is focused on tracking, registering objects in the scene, error filtering and the development of effective interaction metaphors.

Coulton et al. maintain that location-based context allows users to play games that incorporate knowledge of their physical location and landscape. This kind of entertainment provides users with the ability to interact with both real and virtual objects within that space (Coulton et al., 2008). Rashid et al. examine 16 location-aware games based on mobile phones or PDAs (Rashid et al., 2006). It has been noted that location information has significant influence on user behaviour. They consider only applications that are truly mobile by conforming to the official expression of ITU; "*the term mobile can be distinguished as applying to those systems designed to support terminals that are in motion when being used*". The publication describes current Position Determination Technologies (PDT), which namely include Cell ID, Time of

Arrival (TOA), Time Difference of Arrival (TDOA), Enhanced Observed Time Difference of Arrival (EOTD), GPS, assisted GPS (A-GPS) and implied location solutions such as WLAN, RFID tags and Quick Response (QR) codes. Moreover, there is a discussion (Rashid et al., 2006) of how sophisticated services need to interpret raw positional data and proximity to other cellular users and introduce the concepts of spatial databases and XML-derived languages to assist the exchange of geo-referenced data. The described XML derivatives include the Geography Mark-up Language (GML), Point of Interest eXchange (POIX) and NaVigation Mark-up Language (NVML). To this set, it will be useful to add GPX (TopoGrafix, 2011), which is the official GPS eXchange format (GPX) for transferring trajectory logs and describing interesting landmarks. Ultimately, the 5 most crucial concerns that have been raised by traditional game developers and executives have been pointed out (Rashid et al., 2006).

i. Within games, movement is mostly designed as a necessity;
ii. Games cannot simply be consumed; to entertain, they constantly need the players to act;
iii. Games often require other users to function;
iv. Location-based games are not capable of introducing a story;
v. The ability of location-based games to handle player network latency in fast moving games.

From the previous assessments (except the fourth one, which has not been fully valid due to the development of location-based games such as *The Journey II* and *Songs of the North* since the publication of the article), we can conclude that context-aware applications are currently custom-developed, in order to support specific predefined functionalities. Most of them target certain devices and need subscription to specific service providers. Another context-aware system that is used to provide LBS for tourism, by retrieving information from remote content providers, is *AmbieSense*. Its goal "*(...) is to help achieve the digital, ambient environments that make user's information-related tasks easier by adapting to user's context and personal requirements*" (Myrhaug et al., 2004). Undoubtedly, though, these systems can capture the imagination of the intended audience and if they mature, they may achieve an important goal, which is to enhance computer use by making computers available throughout the physical environment, while making them invisible to the user (Weiser, 1993).

2.1.3 Context Models

As we have seen in the previous paragraphs, more than a few attempts to develop context-aware systems have been made by several researchers and developers. In order to increase the effectiveness and usability of such applications, some researchers developed *Contexts Models*, which can assist the creation of formalised technical architectures, as well as satisfy the user needs when applied on context-sensitive systems. Generally, a *model* is an abstract representation of a situation or an object and is used to represent a state of the actual concept or entity that is being investigated.

The majority of the developed context models were designed to fulfil only the information requirements of the targeted application. In a paper, which compares previously defined context models (Kaenampornpan and O'Neill, 2004), the authors observed that there is a need for the design of a context model that provides an understanding of the key elements needed to comprehend users' intentions. Furthermore, they point out that the relationships between various elements of context need to be explored because they can affect the efficiency of context-aware applications and assist the system to understand the user's activities and intentions, as well as to represent the user's world more accurately (Kaenampornpan and O'Neill, 2004). For these reasons they propose their own context model based on the *Activity Theory* (Rogers and Scaife, 1997).

In a very informative paper (Strang and Linnhoff-Popien, 2004), the authors provided a survey of the most relevant approaches to modelling context for ubiquitous computing. Ubiquitous computing is a core concept in this research project and context-awareness is considered an invaluable element of it, which we are going to investigate in the next section of the report. The researchers classified the context-modelling approaches into six categories. Namely, these are *Key-Value* models, *Markup Scheme* models, *Graphical* models, *Object-Oriented* models, *Logic-based* models and *Ontology-based* models (Strang and Linnhoff-Popien, 2004). After classifying them, they evaluated each category against 6 ubiquitous computing requirements. Namely, these requirements are *Distributed Composition*, *Partial Validation*, *Richness and Quality of Information*, *Incompleteness and Ambiguity*, *Level of Formality* and *Applicability to Existing Environments* (Strang and Linnhoff-Popien, 2004). The results of the evaluation demonstrated that *Ontology-based* models are the most expressive models that satisfy most of the requirements. The following table presents the conclusive results of the evaluation.

Model/Requirement	DC	PV	RQI	IA	LF	AEE
Key-Value	-	-	--	--	--	+
Markup Scheme	+	++	-	-	+	++
Graphical	--	-	+	-	+	+
Object-Oriented	++	+	+	+	+	+
Logic-based	++	-	-	-	++	-
Ontology-based	++	++	+	+	++	+

Table 2-1: Appropriateness Indication (Strang and Linnhoff-Popien, 2004)

In 2007, Baldauf et al. analysed the design approaches of several context-aware systems and frameworks. Besides the numerous differences found on such systems, they proposed a layered conceptual architecture, which "*(...) augments layers for detecting and using context by adding interpreting and reasoning functionality*" (Baldauf et al., 2007). The separation between the acquisition and the use of context (Dey, 2000) received particular focus, while developing their framework. The following table presents the proposed layered structure. Furthermore, they compared 8 context-aware systems and frameworks, which support the aforementioned separation of concerns, and presented their similarities concerning the layered structure. The criteria on which the comparison was made were thoroughly analysed and are composed of the *Architecture*, *Resource Discovery*, *Sensing*, *Context Model*, *Context Processing*, *Historical Context Data* and *Security & Privacy* for each approach.

Application
Storage Management
Pre-Processing
Raw Data Retrieval
Sensors

Table 2-2: Layered conceptual framework for context-aware systems (Baldauf et al., 2007)

In 2003, Myrhaug and Göker proposed a distinct user context model. One of the advantages of this model is that it can explore user contexts across several application domains. The model focuses on the users because it examines several aspects of a situation from their own perspective. The structure of the proposed model is composed out of 5 context categories, which can describe the user context. These categories are: *Environment* context, *Personal* context, *Task* context, *Social* context and *Spatiotemporal* context (Myrhaug and Göker, 2003). Each category can be considered as a container of potential application-specific contextual attributes. In a subsequent publication, the authors review aspects of several context models designed by other researchers, spanning across various applications domains, and explain how they relate to the context categories of their user context model (Göker et al., 2009).

User Models

As we have seen in the previous paragraphs, there can be several context types, which have been classified into several categories according to the perspective of the researcher and the application requirements. A certain classification of context into two broad categories was found particularly useful in terms of modelling it (Göker et al., 2009). The authors separated low-level, technical context that is relevant to ubiquitous computing from high-level context that is relevant to the user, both physiologically (Boyd Davis et al., 2006) and cognitively (Reichenbacher, 2007). Context modelling can be used to describe low-level context that is external to the user, whereas user modelling describes high-level context relevant to the personal and task context of the user (Göker et al., 2009). A user model is an explicit representation of the attributes of a particular user or a group of users. User modelling is concerned with building and updating the user model in adaptive systems (Brusilovsky, 1996). A system that is based on a user model and makes use of it can adapt its performance to personalise information and services to individual users. Therefore user modelling is related to personalisation, which is discussed in a following section of this report. Rich was one of the first to suggest that user models can assist the personalisation process of information systems (Rich, 1979). Kobsa has, also, accomplished significant work on user modelling. He reviewed several generic user modelling systems and described their purpose, their services within user-adaptive systems, as well as the required design requirements (Kobsa, 2001). In a latter publication, the authors identified different kinds of user data that user-adaptive systems may need to consider when designing a user model (Kobsa et al., 2001).

2.2 Mobile Ubiquitous & Pervasive Computing

One of the first people that used and explored the term Ubiquitous Computing was Mark Weiser. "*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it*" (Weiser, 2002). His observation is met very frequently in relevant bibliography and is considered fundamental in this research field. Weiser realised that most computing devices, between 1988 and 1993, needed explicit user attention because they were conceived as objects isolated from the environment. His research tried to embed computing technologies in common activities of everyday life. This is how he came across the description of what ubiquitous computing stands for. Interestingly, he tried to match his idea with an evolving, at that time, technology, Virtual Reality. His goal was to have VR make the computer disappear by taking over the human sensory and affector systems (Weiser, 1993). He specifically foretold that VR will be extensively applied on scientific visualisations and entertainment, in future, and identified 2 issues that would restrict it from being the ultimate interface between an actor and a computing device. The first problem was that VR could not produce an environmental simulation of significant verisimilitude at reasonable cost. The second problem was that its primary goal was to deceive the user – to leave the physical world and immerse in the virtual environment. From the previous results, it is obvious that the underlying technologies, mainly geo-visualisation and computer graphics, had not evolved to a level that could be considered adequately effective for generic scenarios. Furthermore, regarding the second problem that has been mentioned, research has shown that users interact with mobile devices several times per day but for short periods, in contrast to bulky computer systems where interaction is less frequent and more time-consuming. In essence, in mobile scenarios the user immerses in the virtual environment very frequently, while being involved in the process of information retrieval. Technological evolution has provided the resources to overcome these limitations and, therefore, signify VR as one of the most suitable user interfaces for ubiquitous computing. Every piece of potentially useful information, which has been produced in the environment, can be visualised in a remotely identical virtual space by the user.

The early concept of pervasive computing was formed based on ubiquitous computing research, which took place in XEROX PARC labs. Several relevant research projects have been triggered around the world since then. Most remarkable are those from Carnegie Mellon University (Aura) (Carnegie Mellon, 2002) and the University of California, Berkeley (Endeavour). A brief review of such operational pervasive environments can be found in a following paragraph. Satyanarayanan indicates that some of the technical problems in pervasive computing correspond to problems already identified and studied in the fields of distributed systems, mobile computing, HCI, Artificial Intelligence (AI) and proactive software agents (Satyanarayanan, 2001). Furthermore, he identifies four driving forces that should be, either partially or fully incorporated in any pervasive application.

i. *Effective use of smart spaces*: bridging the virtual world with objects in the real world;
ii. *Invisibility*: minimal user distraction;
iii. *Localised scalability*: altering the density of interaction according to the proximity of surrounding objects;
iv. *Masking uneven conditioning*: balancing and filtering pro-activity of the system.

After analysing the operation of Aura (Sousa and Garlan, 2002), Satyanarayanan found that it is very important to track the intention of a user, in order for *pro-activity* to be more effective, and that intelligent *service discovery* mechanisms need to be in place, for better *adaptation* between the source and the recipient of a resource. He directly related the *processing capabilities* of the client device to the amount of energy that it consumes and claimed that apart from battery technology, higher application layers must operate by actively considering *energy management* techniques. Finally, *privacy* and *trust* issues, concerning the exchanged information such as user profiles and position, were imperative. This demands a layered approach, which is also an elegant programming practice - to separate abstraction from implementation. Banavar and Bernstein observe the topic from a different perspective and name some of the research challenges in pervasive computing. Their assessment informs us that it was very premature for a computing system to enter the social environment. *Semantic modelling* is one of the challenges because certain ontologies can be introduced to describe users' tasks, as well as their goals, in order to enable reasoning about a user's needs, therefore, dynamically adapt to changes. *Designing* such a complex system may prove difficult because there are issues of service discovery, adaptability and provision. Moreover, *developing* and *porting* an application provides interesting results because of the integration issues related to the assembly of the software components. The last challenge is related to *validating the user experience* because the complexity of the system may produce arbitrary results (Banavar and Bernstein, 2002). Want and Pering believe that the hardware infrastructure required for successful operation of pervasive applications has reached an acceptable level, but there are still issues with the software platform capabilities, which have not advanced at a pace that can take full advantage of this infrastructure. Furthermore, they mention some challenges and expand on them. These challenges are *power management*, *limitations of wireless discovery*, *UI adaptation* and *location awareness*. We detect that power management is still an important issue even nowadays and that location-awareness has developed into one of the core elements of pervasive computing (Want and Pering, 2005).

Review of Existing Pervasive Applications

Pervasive computing requires systems and devices that perceive context. Saha and Mukherje describe a number of state-of-the-art applications, which can be characterised as context-aware and pervasive (Saha and Mukherjee, 2003). We are not going to extensively analyse their infrastructure and functionality, but only assess which of the previously expressed challenges have been addressed. The first application is Aura (Carnegie Mellon, 2002), which has absolutely no relation with the mobile application that was developed by the author of this report during the course of this project. Aura is one of the first pervasive computing environments designed in the early 1990’s in Carnegie Mellon University. It is composed of smaller components that create a distinct whole. It tries to tackle the reduced effectiveness of users due to explicit and implicit distraction inflicted by computers. The application aims to address most challenges in every branch of this research field, as it is an umbrella project. Specific privacy issues and performance measurements for context-aware computing have been identified in the following publications, after extensively researching and observing Aura’s platform (Smailagic and Kogan, 2002) (Sousa and Garlan, 2002). Another pervasive application, *Oxygen* (MIT, 2004), focuses on 8 environment-enabled technologies and tries to evaluate and enhance the user experience. Following next, comes *Portolano* from the University of Washington. This project emphasises invisible intent-based computing and tries to cope with all challenges described earlier. The last remarkable application is

Cooltown (HP, 2001), which focuses on extending web technology, wireless networks, and portable devices to create a virtual bridge between mobile users, physical entities and electronic services. Service composition and user experience validation are the goals, which have been successfully met by the research (Saha and Mukherjee, 2003). Finally, some other intriguing, fully pervasive projects are (i) *Sentient Computing* by Cambridge University which communicates with the environment via the sensory-represented world model and also forms an interface to various actuators (Ward et al., 1998) (ii) *EasyLiving* by Microsoft which is a prototype system for building intelligent environments that facilitate the interaction between people as well as between people and devices (Brumitt et al., 1998) (iii) *WebSphere Everyplace* which provides the functionality for data access and e-business applications to mobile devices (IBM, 2007) and (iv) *Aware Home* which aims to address the fundamental technical, design and social challenges for people in a home setting (Georgia Institute of Technology, 2011). We can assign *Semantic Web* (W3C, 2011) to the context of pervasive computing because information is given a well-defined meaning and enables people and computers to work in cooperation. In that case, it focuses on semantic modelling. Sun/Oracle and later Apache have also produced an open architecture that enables developers to create network-centric services, which are highly adaptive to change. The framework has been implemented on hardware and software platforms (Apache, 2011a). It is called *Jini* and it is a Java derivative that focuses on context-based adaptation. Further information about these projects has been provided in the Bibliography section.

Architecture of Pervasive Systems

Since the introduction of the first pervasive systems, research on the core infrastructure has considerably evolved. Nevertheless, some elements have remained unaffected because they contribute to the structural integrity of each application. By targeting ubiquitous information access, pervasive applications significantly affect how computing devices are deployed and how people interact with the resulting interface paradigms, because one of the main goals is considered to be ubiquitous information access (Papakonstantinou and Brujic-Okretic, 2009a). It is challenging for software developers to create real time applications that continuously adapt to dynamic changes in the environment even if people are frequently relocating and even if there is limited connectivity with remote networking resources. Currently, the essential elements of the global design of pervasive systems have been identified and modelled (Saha and Mukherjee, 2003) into four categories.

i. Mobile devices;
ii. Pervasive networking;
iii. Pervasive middleware;
iv. Pervasive applications.

Most existing applications tend to utilise this architecture. Primary design requirements for pervasive systems have been analysed earlier in this chapter. It is vital to append a more formal approach that was introduced in 2004 into that set. These additional design needs will render the development of the system more robust and would smoothly integrate the context-awareness concept (Grimm et al., 2004). The first requirement is to *embrace contextual change*, because it is impractical to repeatedly ask for explicit user input. The second is to *encourage ad hoc composition* because interposing on an application's interactions with other applications and network services must be a simple and invisible process, as far as the user is concerned. The last requirement is to *recognise sharing as the default process* because actors need to exchange contextual information (e.g. user profiles) and it would become too intrusive to manually act every time that is requested. This would also oppose the definition of ubiquitous computing.

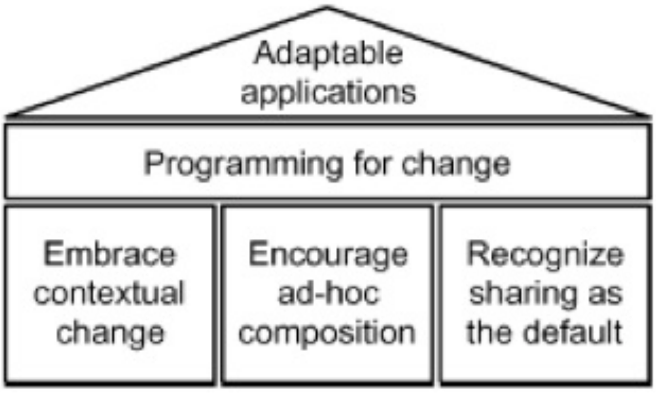


Figure 2-2: Three global requirements that guide the design process of pervasive applications (Grimm et al., 2004)

Mobile Devices

Due to the fact that the four ingredients of pervasive systems are very broad in context, there is specific demand for requirements for each individual one, in order to incorporate them in the global design. The underlying hardware, even though it is continuously evolving, it poses challenges, which have to be effectively overridden. The mobility of the devices causes unpredictable variation in network bandwidth, low trust between mobile elements, weight and size constraints and energy issues. In the extensive literature that exists about mobile computing, concerns about various factors, like mobile information access, support for adaptive applications and mobile networking, have been discussed. It is important to comprehend that the term *mobile devices* also describes their functionalities - for example, the input mechanisms, visualisation methods and operating system support. It is interesting to note that mobile devices are the only element of the proposed pervasive architecture, which is beyond the design remit of this research project. We have been in the position to test newer models but we could never influence their advances. The other three elements have sustained considerable developments.

Pervasive Networking

The next layer in the taxonomy describes the capabilities of pervasive networking, which must be extended, in order to meet the anticipated demand. Quality of Service (QOS) and dynamic discovery of other peers has to be automatically supported. We have mentioned that distributed network computing introduces seamless access to remote information sources and communication mechanisms with fault tolerance, high availability and security features (Satyanarayanan, 2001). However Grim et al. disagree with the approach of some implementations because these try to hide distribution and rely on technologies like Remote Procedure Calls (RPC), which extend single-node programming methodologies to distributed systems. This occurs because "*(...) these technologies hide remote interactions, favour static composition through programmatic interfaces, and often encapsulate data and functionality in the form of objects, they make it hard to anticipate failures, to extend applications, and to share and search data*" (Grimm et al., 2004). They implemented a new framework called *one.world* and evaluated it against a specific laboratory scenario. There are two interesting parts in this architecture, which may complement the design of our system. The first element is that communication is taking place in the form of events, which make changes explicit to the higher layers of the application. The second element is that client environments host applications, store persistent data and, through nesting, facilitate composition of services.

Distributed Computing and Networking Systems

There are several definitions about distributed systems in the literature, but we are going to employ the description provided by Emmerich because he approaches the issue from an Object-Oriented (OO) perspective. He defines distributed systems "*(...) as a* *collection of autonomous hosts that are connected through a computer network. Each host executes components and operates a distribution middleware, which enables the components to coordinate their activities in such a way that users perceive the system as a single, integrated computing facility*" (Emmerich, 2000). The terms *components*, *middleware* and *facility* show that object-based technology is utilised and that after various alterations, it can perfectly suit the pervasive computing context. During the last decade, we witnessed a large growth in communication mechanisms between applications. Technologies such as Inter-Process Communication (IPC), RPC, distributed Database (DB) transactions and message-oriented middleware were found crucial for the evolution of mobile telecommunication software. Distributed OO systems need additional functions to support polymorphism, object migration and other specific technicalities of the proposed system. There are many advantages in distributed systems, though, which can apply in our case. Most importantly, the application can share resources, which are located in remote machines. There is additional flexibility in locating these resources. Furthermore, the increased computational performance, which takes place on a remote client, provides the means to upgrade the functionality. That is because the execution of the algorithms takes place on the remote machine, rather than the local client. The only functionality that mobile clients should accommodate is to act as a receiver for the stream of data and as a medium for visualising content. Furthermore, communication between users should be transparent and must concurrently support a large number of participating entities. Such a system is considered reliable and will not freeze or act unpredictably if a client unexpectedly loses network access. The key elements, which need to be managed during the development process, are the middleware, including the interfaces, and the Interface Definition Language (IDL). Middleware, in our case, is the layer between the network and the sensors that is used to exchange data with the application. Moreover, naming and locating remote objects instances is an additional challenge. Locating remote users and providing communication options is an issue that must be accommodated by introducing specialised methodology. By using programming interfaces to define remote services, we can replace current functionality of the system with an updated one on the fly. The IDL that may be used, apart from defining interfaces, will have to define stubs that are going to act as a server proxy for the client and vice versa. A centralised naming and locating mechanism could also be developed, to provide specific services for the mobile devices. This mechanism is useful because changing the service provider would not have to present difficulties for a client device, while searching for a new one.

There are certain issues which must be taken into consideration when designing distributed applications (McCarty and Cassady-Dorion, 1998). Based on the contextual information that is stored in the mobile device, specific functionalities should be enabled, executed and terminated in a transparent way. Moreover, the *lifecycle issues* of these software objects are responsible for advising the upper layers of the application, when new client devices are nearby, so they can trigger communication with them. *Referencing remote objects* is difficult, because specific types of information such as location, security and metadata must be encoded which is a task that requires additional storage space. While the processing speed is generally increased due to the existence of a remote entity, the *latency* of the underlying communication channel grows proportionally. This happens because more data must be exchanged and, instead of nanoseconds that would be required if it was stored locally, now it may need several hundred of milliseconds to complete. As we cannot assume that all devices will always be online and accessible, the distributed infrastructure must be able to *activate* and *deactivate* specific resources. The state of the resources must be identical to the state they conformed to before the change of condition. This means that persistent storage is required at both ends. Furthermore, synchronisation mechanisms need to be implemented because information will constantly alter form and value. Thus, *parallelism* management of executing tasks is instructed. Additionally, the networking infrastructure should be able to *accommodate any failures* that may occur, because distributed systems have more components to manage, than centralised ones. The processes, which must be implemented to accommodate such events, include storage and retransmission of previous requests, and undo/redo mechanisms for specific transactions. After all, in distributed environments, participants may communicate with each other and with a server node over insecure links. That is why certain security procedures should be introduced; to control parallel requests by various clients and check if they are authorised to access specific information sources. This demands the development of certain facilities that will encrypt, authenticate and authorise specific activities based on predetermined privacy and access control policies.

Pervasive Middleware

Middleware is the proposed solution for interoperable communication between services and applications, generated from various frameworks and platforms. There is not a single definition for middleware, but instead the Internet Engineering Task Force

(IETF) has produced a Request for Comments (RFC 2768) that summarises the concept (Aiken et al., 2000). The goal of the report was to identify existing middleware that could be leveraged for new capabilities as well as for identifying additional middleware services requiring research and development. Typical middleware functionalities include *brokerage services for discovering transactions*, *persistent repositories*, and various levels of *transparency* with the lower and higher layers of the system. Very common middleware tools that are being used nowadays include the Common Object Request Broker Architecture (CORBA), Sun/Oracle's Java Enterprise Edition (J2EE), Microsoft's Component Object Model (COM) and Distributed COM (DCOM). For wireless mobile environments, Java Micro Edition (J2ME) is used most frequently. Raatikainen et al. mentioned that the Open Mobile Alliance (OMA) standardised open interfaces for generic services supporting mobile computing. Moreover, they observed that various competing middleware specifications provide similar services but with slight twists (Raatikainen et al., 2002). In order to overcome problems due to diverse specifications, the Parlay Group (ETSI, 2011) has specified a set of Unifying Modelling Language (UML) models and corresponding Application Programming Interfaces (API) that can be implemented in CORBA, Java and DCOM environments. It is obvious that pervasive middleware must mediate interactions with the networking kernel on behalf of the user, and it must also keep users connected with the rest of the pervasive computing environment (Saha and Mukherjee, 2003).

2.3 User-Centred Mobile Computing

Adequate acceptance and familiarisation with a mobile environment depends to a large extent on the applications and services that are provided to the actors. Adjusting the services to support the actual user needs is a crucial factor for the information exchange process. A communication system that is based on the provision of real-time context would be ideal for successful immersion of a person in a virtual environment. The system could represent each user in their own customised environment and could also adapt the supported functionality according to different circumstances, physical conditions and even behaviour. Arbanowski et al. concluded that a system like that must support context-awareness, adaptability and certain personalisation mechanisms (Arbanowski et al., 2004). Context-awareness has been discussed in previous sections and in this one; we are only going to describe potential issues that can affect the user needs. Furthermore, personalisation will be introduced as it is applied in many fields of the research. Adaptability of a service to specific user needs implies changes to the behaviour of the application when certain attributes and features of the natural environment have been altered. Functionalities that should be supported by the adaptive layer include environmental monitoring, event notification, information distribution and context matching. As we have seen in the previous chapter, adaptive applications must access components from several nodes of the system. This process needs to be accomplished automatically and transparently by the user. Efficient processing of user preferences, device capabilities (incl. sensor technology) and system requirements must be carefully handled in adaptive applications. Arbanowski believes that adaptive applications are based on multiple models such as QOS, user preference and behaviour. These models are implications of extensive use of AI (e.g. Bayesian, probabilistic, regressive, predicative, propositional, logical). For this reason, a *model combiner* should be designed in order to formulate goals out of distinct sub-goals. It is possible for an individual user to have multiple roles in a scenario. The aforementioned flexibility has to be supported by the framework.

2.3.1 Personalisation & User Profiling

One of the reasons that personalisation has evolved rapidly is due to information overload that people experience in their daily interactions with computing devices. The basic element that is utilised for personalisation is user context because it can characterise the current state. The functionality that is provided after processing user variables should run transparently. "*Personalization is defined as a process that changes the functionality, interface, information content or distinctiveness of a system to increase its personal relevance to an individual. The effect of the changes should persist across sessions*" (Blom, 2000). The disadvantage of this user-centred approach is that it does not mention who or what initialised the process. *Personalisation* should optimally be application-initiated, whereas *customisation* should be user-initiated. In this context, however, personalisation means that something is individualised to fit a person's needs and does not include a definition of the process of how to achieve it (Jørstad et al., 2004). Therefore, customisation is regarded as one way to accomplish personalisation. Blom divides the reasons for personalisation into two categories and a number of subcategories. The first reason comes from *work*-related motivations, while the second from *social* motivations. Furthermore, *work* incentive personalisation may be triggered in order to enable access to information content, accommodate goals and individual differences. On the other hand, social motivations for personalisation include expressing the unique user identity and elicit emotional responses. The model that is described by Blom can be applied in many different cases including mobile software platforms. Information in a user profile must change dynamically, which means that the context has to evolve with the lapse of time as a subsequence of user experiences. The concept of the user profile can create a highly cohesive link with the user, but can also raise concerns about personal privacy. Searby gives another definition of personalisation and names the categories that it can be applied to. "*Whenever something is modified in its configuration or behaviour by information about the user, this is personalisation*" (Searby, 2004). Effective personalisation applications can be found in targeted marketing and advertising, but most interestingly, in collaborative filtering, which exploits similar behaviours by monitoring the patterns of groups of users and then trying to match a user with an appropriate group.

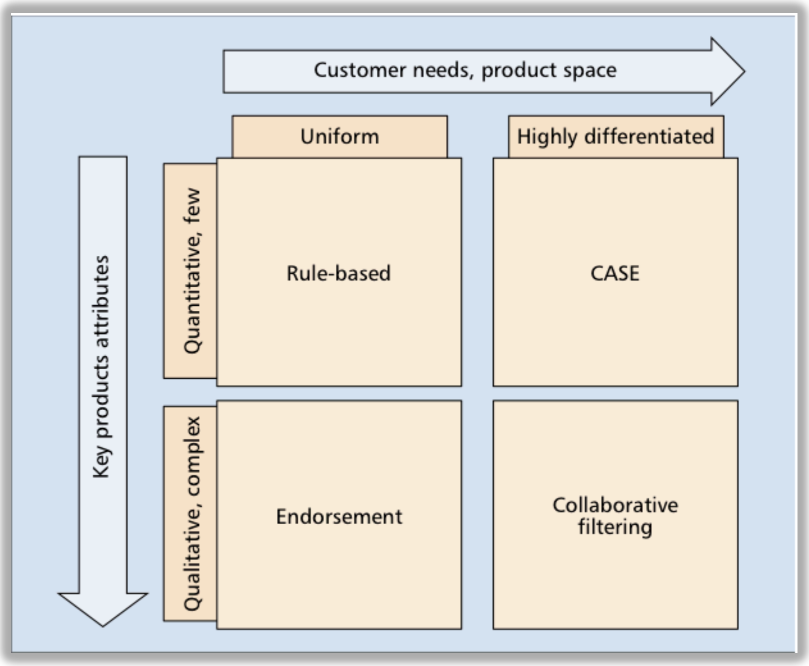


Figure 2-3: Technologies to achieve personalisation (Arbanowski et al., 2004)

In Customer Relationship Management (CRM), the departments of an organisation need personalisation mechanisms in order to build their business model based on the user, rather than their products or services. Easy *integration* with the required services and *knowledge management* are the last 2 categories that have been defined by Searby. The latter includes features such as responding to user context, matching information to interest and preferences and sharing information with other people that conform to specific restrictions. Similarities in interests and context can benefit the Information

Retrieval (IR) process and promote collaboration. *Identity* is another central concept of personalisation because specific implementation methods need to determine if a user is actually the one that is trying to prove. *Authorisation* and *authentication* refer to the association of a person with an identity and the access permission controls over information, respectively (Searby, 2004). Successful personalisation depends on the establishment of trust between the involved parties, which implies an identity management subsystem in the system architecture. If the user profiles are connected to the available services, the combination could offer many opportunities to integrate with other applications and devices. To develop a personalised mobile service, the issue of service continuity has to be taken into consideration as well. Every element of the service has to be available at any time and when the context is updated, it should remain in that state for the whole period that the service is running, even if the user does not act. Jørstad et al. define personalisation in that context. *"Personalization of a service means that mechanisms exist to allow a user U to adapt, or produce, a service A to fit user U's particular needs, and that after such personalization, all subsequent service rendering by service A towards user U is changed accordingly"* (Jørstad et al., 2004). Furthermore, they name two mechanisms to apply personalisation to a service, the *explicit*, which is achieved by the user or service provider and the *implicit*, which is adopted by the service after specific user behaviour is recorded and analysed. The following table shows the features of personalisation, in relevance to how they are achieved and the motivation that is required for the users.

Features to personalise	Explicit or Implicit	Motivation
Look & Feel	Explicit	Social
Personalisation of service portfolio	Both	Work
Individual service personalisation (behaviour)	Both	Social & Work
Personalisation by content	Explicit	Work
Personalised service composition	Explicit	Work

Table 2-3: Classifications of Personalisation (Jørstad et al., 2004)

Technologies and Methodology

The most common incentive to personalise an application or service is related to the social aspects of life and particularly between younger people. The potential to facilitate collaborative work is very high and should not be ignored. Personalisation technologies especially for service composition are too difficult to put to work, mainly due to their high complexity while integrating them with the other system components. To support personalisation Jørstad provides a distinct methodology composed of six rules. *Gathering user context information* through implicit and explicit means, ubiquitous *availability of the user profile* and a profile for the *mobile device* are the most important. In addition, the continuous *availability of user content* and the specific *user composed services* that have been planned for execution should be part of the process (Jørstad et al., 2004). *Privacy issues* will be discussed later in this chapter. In 2002, Wagner et al. described the standardisation efforts that took place in order to assist the development of interoperable personal mobile services (Wagner et al., 2002). *Federated online identity* aims to enable a single logging-in procedure for a number of different services. The *Liberty Alliance Project* (The Liberty Alliance, 2010), *Live ID* (Microsoft, 2011) and *OAuth* (Hammer-Lahav, 2010) are the best-known efforts that direct standardisation activities in this area. The reader can find further information about these technologies by following the web links found in the Bibliography section. The variation of hardware and software infrastructure of mobile devices requires a personalisation scheme that could provide optimised content to *different client devices*. The Composite Capabilities / Preferences Profile (CC/PP) (W3C, 2007) and the User Agent Profile (UAProf) (WAG, 2001) have this goal exactly; to address the needs for device independence and provide an interoperable basis for metadata descriptions of profile information (Wagner et al., 2002). Lankhorst et al. proposed a custom Personal Service Environment (PSE), which could prove a valuable reference in the composition of our system. It offers generic discovery of services as well as adaptation services to accommodate changing conditions and situations. The PSE is a holistic approach to the personalisation of mobile data services in a scalable and efficient way. Furthermore, it employs the centralised as well as the distributed functionality and intelligence (Lankhorst et al., 2002). The approach that is examined classifies profile architectures in distinct categories and describes its contents.

i. *User Profiles*, including preference, interests and behaviour;

ii. *Service Profiles*, including characteristics of services;

iii. *Mobility Profiles*, including position, orientation and speed;

iv. *Network & Device Profiles*, including hardware characteristics;

v. The remaining contextual personalised information.

Several frameworks and standards for profile management have been introduced, based on the capabilities of a system. Some of them are the Open Pluggable Edge Services (OPES) (Barbir et al., 2004), the Content Distribution Internetworking (CDI) (Day et al., 2003) and the Web Intermediaries (WEBI) (Cooper and Nottingham, 2002), which are working groups of IETF. The part of the architecture that focuses on the automatic management of the declared profiles is important for the design of our system. Collaboration through service discovery is going to be supported after developing a brokerage subsystem based on agent technologies. Moreover, evolution of the brokerage agents to adapt to changes in user preferences and context is required (Lankhorst et al., 2002). In that publication, the authors separate the tactics and strategies that may be adopted by intelligent software agents into 4 processes; the *rule-based* approach, the *case-based reasoning* approach, the *game-theoretic* approach or the *adaptive-learning and evolutionary* approach. They advocate evolutionary agent negotiations for two reasons. The first reason is that time, resource and behaviour tactics in agent negotiations occur in mobile service delivery negotiations, as well. The second reason is that the agent's mental state captures knowledge about other service delivery issues, which in our case are the profiles. Additionally, in order to implement suitable brokerage mechanisms, an agent system architecture is required because it enables the creation of negotiation strategies and evaluates the QOS. For mobile service delivery, a case-based architecture and a fuzzy architecture are the most suitable. These architectures are appropriate because they can support the user, service and context aspects. (Lankhorst et al., 2002). Ultimately, they describe the implementation issues of their application and name the technologies that they put to work. For the design of such a system the Agent Unified Modelling Language (AUML) (FIPA, 2007) can be used, which is a UML derivative and supports the syntax of data flows and the interaction between agents. Two software platforms have been identified for the implementation of the intelligent agents; Grasshopper (IKV++, 1999) and Tryllian's agent development kit (Tryllian, 2007). Although, during the course of the research at hand these platforms have been outdated, an extensive evaluation of further agent development products can be found in the following publications (Nguyen et al., 2002) (Leszczyna, 2004) (Sudeikat et al., 2005).

Terziyan considers a different approach that allows operations based on different levels of profiles to reach consensus about general and specific features of the relationship between two actors or between an actor and a stakeholder. This multilevel profiling framework is based on XML documents that become interfaces in distributed environments. The advantage over CORBA or other distribution technologies is that they are easier to develop and be used by a wider range of users and that they can naturally represent the selected context (Terziyan, 2001). The top level copes with international and local security, as well as with privacy and transaction management issues. The level beneath it includes Document Type Definitions (DTD). Profiles is the primary level that the user context is stored in, while the last one holds concrete XML documents, which describe different features of obtained consensus. The following figure describes the suggested framework and the connection between its elements.

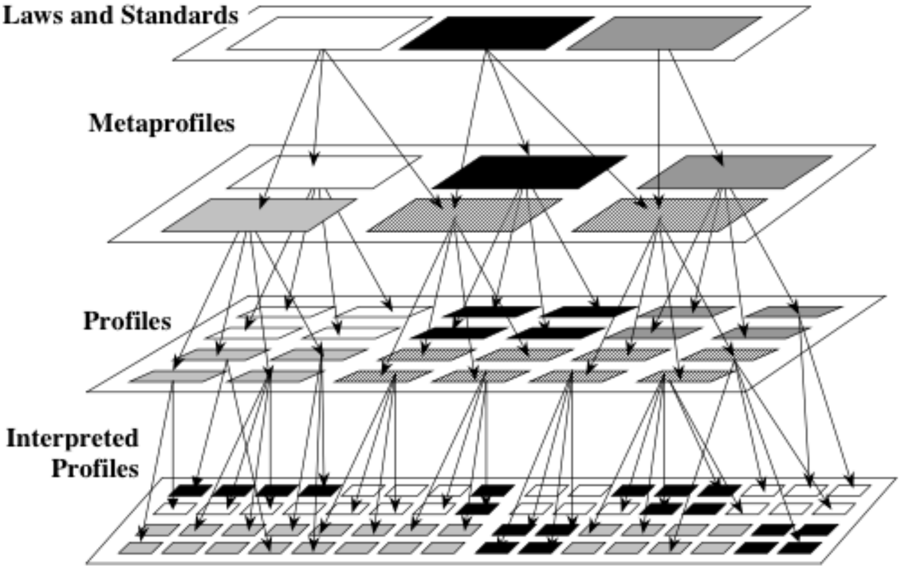


Figure 2-4: Multilevel structure of profiles (Terziyan, 2001)

In a *semantic metanetwork* (Terziyan and Puuronen, 2000) every level controls the structure of the lower level. A multilevel network can be used in an adaptive control system and the structure is automatically changed following alteration of the environmental context. "*Multilevel representation of a context allows reasoning with contexts*" and provides solutions to the following problems (Terziyan, 2001).

- Deriving knowledge interpreted using all known levels of its context;
- Deriving unknown knowledge when interpretation of it in some context and the context itself are known;
- Deriving unknown knowledge about a context when it is known how the knowledge is interpreted in this context;

- Transforming knowledge from one context to another;
- Deriving trends within any problem, considering it in several contexts and using such trends to derive more precise solutions to the problem.

Weiβenberg et al. present a third comprehensive approach to personalisation for mobile applications, which we are going to describe at this point. The FLAME2008 prototype (Weiβenberg et al., 2004) is capable of supporting mobile users by providing personalised situation-aware services in push and pull modes. Structural and Entity Relationship (ER) diagrams are provided, which depict the implemented general user model. The core entities that are associated with the user are a *unique identifier*, an *address book*, a *calendar*, a *profile*, a *situation* and a *history*. The first four are self-explanatory and will not be analysed in more detail. History holds value pairs of location and time in addition to information or services that a user requested at a specific location and time. Situation was found to be more complicated to model, so they have made the assumption that it will not change during a certain time interval and one can describe a demand that occurs in this situation (Weiβenberg et al., 2004). Furthermore, they separate the system attributes into three levels; *sensor values*, *context* and *situation*. They developed an ontology, which is formed out of sub-ontologies and is clearly depicted in the following figure.

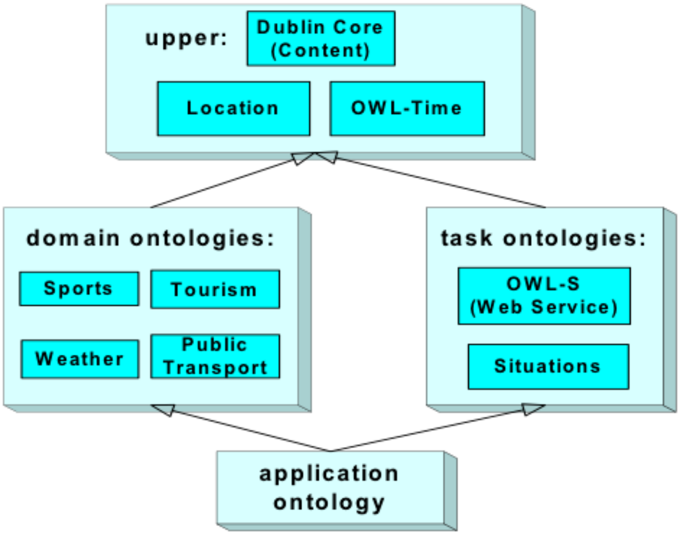


Figure 2-5: FLAME2008 modular ontology architecture (Weiβenberg et al., 2004)

As we can see in the previous figure, situation is incorporated into *task ontologies*. Situations are described by semantic situation profiles. They are determined by a semantic classification of an aggregation between abstract user context, user profile and related information. "*Situation detection requires a modular ontology with sub-*

ontologies for all the different context dimensions (time, location, ...), all combined in the situation ontology. (...) Whenever a significant change in sensor data is detected for a user, his service space is re-evaluated, using the situation detection functionalities of the semantic registry" (Weiβenberg et al., 2004). Profile and context data can be imported to the inference engine on demand. In FLAME2008, the logic predicates are implemented in Java and the selected engine was *Ontobroker* by Ontoprise. *Ontobroker* is a deductive, object oriented database system that was originally developed as a research prototype and later became a commercial product (Ontoprise, 2011).

Personalisation Applications

There are several uses of personalisation in mobile context-aware applications. We are going to examine a few of them, which have functionality similar to the one our framework intends to realise. The first application tries to cope with mobility in urban environments, which directs the development of innovative tools and novel techniques for individual route planning (Balke et al., 2003). It attempts to combine information from various online sources, in order to integrate and use it for recommending specific routes to mobile users based on their preferences. The finest route is calculated by using the SR-Combine algorithm. The required information is integrated by agent-based systems that collect information from distributed sources within complex CORBA architectures. The system incorporates a centralised and a distributed architecture for use with ministry and other user data, respectively. It is built on an IBM DB2 server (IBM, 2011), which stores many types of data (e.g. GIS, routes, user preferences) and on a J2EE application server (Oracle, 2011), with Apache (Apache, 2011b) to provide the communication interface. The authors state that the users can express preferences on various features of the route, which leads to advanced personalisation.

The second system that we are going to examine tries to prevent interruptions in the mobile map reading process by applying personalisation techniques (Nivala and Sarjakoski, 2004). Interruptions may occur when the conceptual models of the cartographer and the user do not match. This problem can be solved by bringing context-awareness into the maps of mobile devices and by providing each user with symbols, which are adapted to current conditions and preferences. The main challenge in the map reading process is how well the map is being perceived by the user. Moreover, repetitive disturbance can have frustrating effects. However, a few interruptions may be beneficial, because they may maintain the level of interest of a user during a specific task (Nivala and Sarjakoski, 2004). The authors concluded that apart from the user's position, which is the obvious context, several other context elements are required. Even a small change in the map content can significantly improve the usability of the map and satisfy the user. The context elements that have been applied are: a specific use case (e.g. outdoor, cycling, emergency), user identity (e.g. age, language) and time. The utilised technologies include geo-databases with national spatial data and XML for information delivery, and the presentation aspects conform to the specifications of the Open Geospatial Consortium (OGC) and Scalable Vector Graphics (SVG). The results of this personalised application are obvious. By taking into consideration the contextual information that is being utilised, the POIs on the map can be visualised from several different perspectives. This can actively benefit different age groups, as well as people intending to perform a specific activity in the actual environment.

Kortuem et al. analyse another very intriguing application that focuses on enhancing mobile collaboration by introducing personalisation and profiling mechanisms. They define "*Profile-based cooperation as a way to support awareness and informal communication between mobile users during chance encounters*" (…) "*and are interested in how mobile technology can be used during encounters of people who have never met before and who don't know each other*" (Kortuem et al., 1999). The interesting part, though, is the architecture of the system. It is separated into 4 conceptual elements. *User Profile*, *Encounter*, *Profile Exchange* and *Rules of Encounter*. Apart from the typical user context, profiles record an additional attribute (i.e. interest) that users must agree with, in order to exchange user context. The notion of an encounter is that it is a situation (e.g. proximity) that may occur between individuals, within a certain span of time. Profile exchange takes place only if the owner agrees to submit any volume of his context. Between devices, the rules of encounter contain the notion of software agents, which can scan profiles for user-defined patterns on behalf of users. An extra feature of each agent is to support initialisation of relevant activity after a successful match has been found. The potential applications that Kortuem et al. conceived for this infrastructure were an awareness tool, a reminder, a diary and a matchmaking service. Proximity of other devices was found from custom-based radio transmitters, but for our research, GPS could easily replace the proposed position determination technology. During an encounter, the devices communicate on the service layer, which is based on *Jini* by Sun/Oracle. The rule-based system, which is manually defined by each user, is very simple and includes only three conditions. The examined (i.e. 1999) implementation did not include a particular privacy policy and the developers did not pursue that option because their main target was to promote awareness and informal communication between people that had never met before.

Under the guidance of a European IST project called CRUMPET, which aims at the creation of user-friendly mobile services customised for tourism, Schmidt-Belz et al. focused on the usability challenges and how they can be improved after the application of certain personalisation schemes. CRUMPET employs personal user interests and current location. Additionally, a mediation service is introduced that is based on software agents such as middleware. They handle context awareness of the system in a manner that should be able to solve the usability issues associated with mobile devices and improve the interaction with the user. The process to support personalisation is divided into 3 elements (Schmidt-Belz et al., 2002).

i. Adapting the user’s tourism interests to other preferences;

ii. Updating the user’s model according to interaction history;

iii. Being aware of the user’s current spatial context.

Modelling user interests is based on domain taxonomy, while employing the user model could provide proactive execution of specific functions. An important observation is that users must be able to override the model assumptions. The model can adapt to current context explicitly and implicitly by using movement to infer the interest of the actor. They also note that when location-awareness is combined with user modelling, new possibilities in location-based services with added values for the user can open up (Schmidt-Belz et al., 2002). Some applications sense active objects or users in the vicinity of the actor, identify their location and offer meaningful directions. For offering location-based services, the logical or topological position is required, in addition to the orientation and the applied privacy constraints. CRUMPET can provide means to draw the attention of the user to nearby objects by utilising the middleware agent and a separate geo-coding service. Moreover, for personalising map visualisations, technical as well as cognitive elements are exploited. Technically, the novel part of this system is the operation of intelligent collaborative mediation agents. Three distinct levels of agency are described, with the most crucial being the second one. It is responsible for matchmaking, offering anonymity, service provision and further secondary functionalities. The ontology that exists between the mediator agents and the user models covers the tourist domain, the service features and the information request habits. Further addition of metadata on services and content is required. Agent implementations are based on FIPA standards (IEEE, 2011).

2.3.2 Privacy & Trust

Several times across the Literature Review we mentioned that privacy issues are important for the balanced provision of services to users. *Balanced* means that the value of the provided quality of service towards the compromised information will favour the user. In context-aware computing, certain types of context may be characterised as private and confidential by the users. Furthermore, in pervasive computing, the distributed infrastructure states that certain low-level functions must exist, which cope with the privacy concerns between mobile clients. Personalisation implies an invisible connection between the actor and a device or application/service. This way, examining a device and the personal data that it contains could possibly reveal the identify of a specific user. The concept of privacy has been with the human race since ancient times but especially in our times – the Information Age, almost everyone who works with digital technologies tries to minimise the amount of personal information that is passed to a 3rd party. Fraud is the most obvious reason for which people try to cope with privacy challenges. Technology has become a tool at the hands of criminals, as well. Users understand that the span of possible harmful channels of communication has increased in order to deceive them. There is a big discussion about the quality, volume and type of information that service providers keep stored in their records. On the other hand, there is some information that must always remain confidential (e.g. government data and information from organisations that can provide a competitive advantage). Traces of personal information are logged every day for millions of people in the technologically developed world, some times legally while others illegally. Network operators and ISPs are obliged to record mobile phone operation and computer activity in databases for several years. Individual countries and continents apply legislations on service providers, about how to manage sensitive information (e.g. Regulation of Investigatory Powers Act) (Home Office, 2000). Definitions of how to protect user private data (i.e. Data Protection Act, Freedom of Information Act) (Department of Constitutional Affairs, 1998) (Department of Constitutional Affairs, 2000) can also be found with variable strictness in every European country. EU has applied the standards and each country is responsible for conforming by introducing relevant legislations. Service providers need to take under consideration that personal data may grow fast and that it is difficult to keep it private. Moreover, in order to allow each person to manage which types of data must be kept secret and which need not, certain privacy schemes must be applied. It is possible to gather information about a person from different sources, but, if combined, it can produce a complete user profile. People tend to protect their personal information by any means possible, but there are situations in which they can release information more easily than usual. These factors include the level of appreciation towards the service and, also, how established the provider is. Generally, each person has its own limits. This is why personalised communication should take place directly between the source of the information and the recipient. A service provider needs to build an amount of trust with the user (Viega et al., 2001). Macgregor classifies personal information in five categories; *public*, *private*, *situational*, *historical* and *transitional* (Macgregor, 2003). It is noted that information is passed to the service providers in a two-stage process - initially during registration and subsequently while using the service. He also maintains that while some data is rendered anonymous by using data aggregation techniques from diverse sources, other elements of a profile can be revealed rapidly. Several software tools are already available that employ matching and geo-demographic data to classify and segregate users.

Raper claims that LBS can only exist within a privacy framework, because they generate real-time personal data (Raper, 2002). For efficient operation of a location-based service the users must trust the provider because the latest position coordinates are transmitted in exchange for advanced GIS functionality. The EU Data Protection Directive (EU, 1995), which defines sensitive data, does not include positional traces in this category. LBS providers need to appear as trustworthy parties, otherwise the consumers would not trust them with handling their spatial information. Furthermore, Raper investigated the fundamental concepts of location privacy and described the most critical. Initially, a set of specific questions needs to be examined; whether the identity of a person can be defined by location data and which other sources of information must be aggregated in order to achieve it. The next issue discussed is concerned with how metrics can be applied on positional accuracy and at which resolution location data would be rendered sensitive. A specific pattern has to evolve about the way that location context is accessed. Does it have to be available only while a person is using the service or does it need to stay in persistent storage for more than that? For LBS offered by a non-network provider, this can cause many difficulties. Furthermore, some specific information needs (e.g. proximity search) of a user and, most importantly, the way that users try to access it, may reveal geographic patterns that may be exploited to expose the socioeconomic profile of the actor.

In order to understand what is important for users when choosing whether and what to disclose about their current location, Consolvo et al. conducted a specific study. The experiment included 16 participants and was separated into three phases (Consolvo et al., 2005). In the initial phase, the participants completed a demographic questionnaire that included a *Westin/Harris Privacy Segmentation Model*. This is a privacy classification survey, which classified the participants into *fundamentalists*, *pragmatists* and *unconcerned* based on their privacy awareness. For the second phase, they used the *Experience Sampling* technique, which involves access and accurate reporting of information available to conscious awareness. The third phase was an interview during which general concerns about location technology were expressed. One of the most notable results was that participants either, wanted to disclose what they thought would be useful to the requester, or completely rejected the request. There was no evidence found of participants intentionally blurring their location. Furthermore, they observed that the privacy classification, by the Westin/Harris model, was not a good predictor of how they would respond to requests for their location. The most crucial conclusion about the factors that would allow users to release sensitive data is hierarchically based on who is requesting information, then the reason for which it is requested and lastly, which level of detail is acceptable by the requesting party. For the development of our privacy guidelines, it is possible to include the emerged pattern of the previous research, which in essence, describes the decision process for location disclosure. Similarly, Iachello et al. developed a prototype application called BOISE, and fused the results from the previous research (Iachello et al., 2005). They offered specific instructions to developers about design issues affecting location-aware services. The main requirement was that the system had to support various levels of location granularity. From initial investigation, the results illustrated that location is used as proxy for other activities and automatic initialisation of services has to be done very carefully, if at all. Furthermore, in person-to-person communication flexible replies, such as denial, deception and simple evasion have to be supported. The last rule reflects a user who has to be in the appropriate mood in order to communicate. Thus, notification about his state must be available prior to communication.

According to Spreitzer et al. there are two major issues in context-aware system security (Spreitzer and Theimer, 1993); ensuring the accuracy of location information and identities, and establishing secret communications. In addition to protecting the content of the communication, the address of the content should also be protected for preventing disclosure of location information. Spreitzer et al. also mention that perfect privacy guarantees are generally hard and expensive to develop. Ideally, the user should be able to control the contextual information and who may gain access to it. "*The system architecture needs to provide user-controllable tradeoffs between privacy guarantees and both functionality and efficiency. But it is difficult to be specific about what context information should be visible to who, and when*" (Chen and Kotz, 2000).

As we have seen earlier, the examination of user profiles and context is a requirement in order to implement personalised context-aware services. Most existing solutions for privacy control are very complex and provide privacy only for selected context. An application that addresses the technical issues related to privacy and contextual information exploitation in order to provide different levels of service is examined by Mitseva et al. (Mitseva et al., 2006). The novelty of this work is that it suggests an approach that offers context-aware privacy for flexibility of sensitive information according to various levels of granularity but without conforming to the standardisation initiatives that define inflexible architectures for user context offering some security measures but do not guard privacy (Mitseva et al., 2006). Clearly, this implementation includes some functionality that has not been adopted by any other application, which has been examined until now. Some of its novel features include filtering of sensitive data prior to handling, integration of low-level context to higher-level profiles, restricting interactions based on scenarios and rule-based context-aware privacy mechanisms that make decisions on behalf of the user. Their concept for privacy must maintain information, context and location privacy. Identification of a number of issues, for the server-side of the system, has proven robust and the correct usage of the application has been laid out. The policies that govern communications and exchange of data incorporate techniques that request only the minimum necessary context for the service to operate. Likewise, it is imperative to keep it for the time needed and not pass it on to any other party without informing the owner. The method that differentiates the level of services between users is twofold. Different users access the same service with the same device type or the same service is accessed by employing different device types. This method can prove beneficial for our framework because it may support two kinds of devices, one offering minimal functionalities (e.g. mobile phone) and one with more advanced features embedded (e.g. smartphone). The system developed by Mitseva et al., supports various profile types. Namely, these are *users*, *devices*, *context*, *scenarios* and *services*.

2.4 Mobile Mixed Reality Interfaces

The complexity of most context-aware systems is reflected in the provided user interface. The main cause for this phenomenon is the existence of numerous components that are integrated in such solutions, but without conceptualising the design of dedicated interaction and visualisation means. With the advance and increased availability of technologies such as computer graphics and mobile computing, in software and hardware terms, design and implementation of intuitive user interfaces has become a consideration for any project that aims to satisfy complex user requirements as well as to fulfil the operational goals and improve usability.

2.4.1 Virtual Reality

Geo-Visualisation

Geographic Information Systems that were introduced in the early nineties could manage and visualise only two-dimensional (2D) geographic information composed out of spatial data. One of the capabilities of geographic visualisation or geo-visualisation is to present phenomena and representations of the real world, one overlaid on top of the other, in a 2D medium. For geographers the use of this kind of information is invaluable for the composition and exploration of coherent datasets and for the collection of data related to their work. In contrast, non-specialists are familiar only with some limited features of geo-visualisation, such as the representation of digital maps, which are mainly employed for navigational and wayfinding purposes. MacEachren defines geographic visualisation as "*a form of information visualization that emphasizes development and assessment of visual methods designed to facilitate the exploration, analysis, synthesis, and presentation of geo-referenced information*" (MacEachren, 1998). The introduction of 3D technologies has added interactivity and dynamism in the visualisation of spatial information. Jo Wood argues that "*more than ever before, we* *have the ability to create more information-rich, interactive, realistic and dynamic visualization processes using 3D*" (Wood, 2005). The realism, which 3D representations of the natural world provide, is higher and complementary to what 2D maps visualisations have been offering.

GeoVR Construction	GeoVR Use
Selection	Interactivity
Immersion	Augmented Reality
Information Intensity	Object Behaviour
Autonomous Agents	Autonomous Agents

Table 2-4: The GeoVR factors (Wachowicz et al., 2002)

Virtual Environments

Virtual Reality (VR) is conceptualised as the set of underlying 3D technologies, which can compose a synthetic environment. The resulting environment is a Virtual Environment (VE) and it can provide interaction patterns for its users. Fundamental LBS applications that embed VEs assist the end-users to navigate by providing representations of the real world and its elements that enhance user presence in the virtual environment. This is also supported by Verbree, who writes that "*Virtual reality offers new and exciting opportunities to visualize 3D GIS data. Users can walk through 3D environments, see newly planned buildings and appreciate changes in the landscape. In most cases, however, interaction with the data is limited to viewing. At the most there is some restricted form of navigation and interrogation, e.g. the user walks around in the virtual environment and can point to objects in the scene and ask for information from a GIS database*" (Verbree et al., 1999). Based on the taxonomy of Mixed Reality (MR) displays, which is represented by the *virtuality continuum* (Milgram and Kishino, 1994) found in the following figure, MacEachren et al. identify four important factors for virtuality. Namely these are: i*mmersion*, *interactivity*, *information intensity* and *intelligence of objects* (MacEachren et al., 1999). These are a group of meta-factors that contribute to the virtuality of a geo-referenced VE and can be shared with the real environment as well.

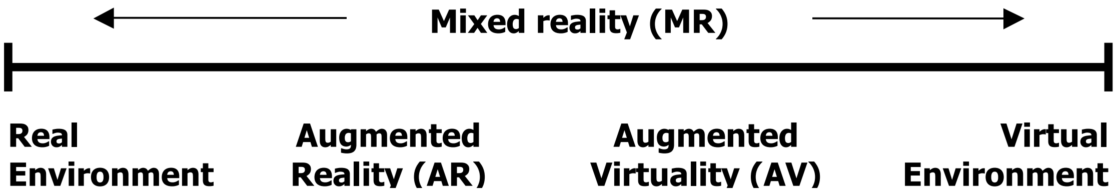


Figure 2-6: The virtuality continuum (Milgram and Kishino, 1994)

Interactivity of a geo-referenced virtual environment does not only include the ability to move on all degrees of freedom (i.e. 6-DOF) but also allows the manipulation of certain characteristics of the environmental entities. *Information intensity* refers to the detail with which entities and its features are represented in the virtual environment. *Intelligence of objects* describes the ability of virtual entities to present context-sensitive behaviour. MacEachren, also, writes that "*immersion describes the sensation of "being in" the environment. (...) There will be degrees of immersion in a virtual environment that, in part, are a function of which senses are stimulated in ways similar to that experienced in the real world and, in part, are a function of the fidelity of that stimulation*" (MacEachren et al., 1999). Every factor that has been described contributes to the realism of a geo-referenced VE. An *interactive* virtual environment can be considered one that allows the users to interact with virtual entities the same way as they would with real objects. *Information intensity* can be affected by how real do the virtual objects look. Realism in a virtual environment can be enhanced if the *intelligent virtual entities* exhibit behaviours that correspond to those of real objects. The following paragraph continues the discussion about *Immersion* and relates it with the user's sense of *presence* in a virtual world. In this project, though, we consider immersion as the result which the utilised technologies and low-level functionalities of our framework produce in order to enhance the user's feeling of *being in* the virtual environment.

Presence

According to Slater et al., immersion is an objective description of what any particular system does provide, or else, a quantifiable description of a technology; and describes the level to which the visualisation output is *extensive*, *surrounding*, *inclusive*, *vivid* and *matching*. Following next, the authors define *presence* as "*a state of consciousness, the (psychological) sense of being in the virtual environment, and corresponding modes of* *behaviour*" (Slater et al., 1996). Furthermore, they suggest that presence depends on two variables.

- *The first variable is the extent of the match between the displayed sensory data and the internal representation systems and subjective world models typically employed by the participant;*
- *The second variable is the extent of the match between proprioception and sensory data.*

The authors also suggest that the realism of a virtual environment (i.e. virtual scene and 3D content) is a contributing factor to the feeling of presence (Slater et al., 1996). In a later publication, Slater et al. relate immersion and presence by writing that "*(...) presence is a "response" to a system of a certain level of immersion*" (Slater et al., 2009b). The authors maintain that there are two ways to achieve presence; either by developing a high fidelity system that it becomes indistinguishable from reality or by embedding in a specific representation of reality only what is important for the task in hand (i.e. deliver presence even when the level of immersion is not high). The aforementioned literature suggests that while immersion has measurable criteria - such as field of view, image and colour resolution, textures, and scene content - the sense of presence is a subjective experience, quantifiable only by the user experiencing it.

Another definition of presence has been proposed after taking into consideration the Heideggerian and Gibsonian views of the ontology of being. The authors support that "*presence is tantamount to successfully supported action in the environment*" (Zahorik and Jenison, 1998). The authors maintain that reality is formed through actions and that presence in a virtual environment is based on the ability to act in it. Other researchers have also supported this concept, arguing that matching kinaesthetic proprioception and sensory data is crucial (Schubert et al., 2001). In a journal paper, Sanchez-Vives and Slater explore the factors that influence presence in virtual environments (Sanchez-Vives and Slater, 2005). Namely, these factors are (i) display parameters, (ii) visual realism, (iii) sound, (iv) haptics, (v) virtual body representation, (vi) body engagement. In our project, we are going to take into consideration every factor that was described by Sanchez-Vives and Slater, except sound and haptics. Earlier, Witmer and Singer have also defined presence as "*(...) the subjective experience of being in one place or environment, even when one is physically situated in another*" (Witmer and Singer,

1998) and classified the factors that contribute to the sense of presence into four categories. The following table presents this classification.

Control Factors	**Sensory Factors**	**Distraction Factors**	**Realism Factors**
Degree of control	Sensory modality	Isolation	Scene realism
Immediacy of control	Environmental richness	Selective attention	Information consistent with objective world
Anticipation of events	Multimodal presentation	Interface awareness	Meaningfulness of experience
Mode of control	Consistency of multimodal information		Separation anxiety/ disorientation
Physical environment modifiability	Degree of movement perception		
	Active search		

Table 2-5: Factors Hypothesized to Contribute to a Sense of Presence (Witmer and Singer, 1998)

In an extensive review paper that examines the concept of presence, Lombard and Ditton identify six different conceptualisations of presence and present a detailed explanation of each one (Lombard and Ditton, 1997). Namely, the authors consider presence as (i) social richness, (ii) realism, (iii) transportation, (iv) immersion, (v) social actor within medium, (vi) medium as social actor; and outline the causes that they believe to encourage or discourage a sense of presence in media users, as well as the physiological and psychological effects of presence. Furthermore, the concept of presence has been extensively defined by ISPR (International Society for Presence Research (ISPR), 2000). The previous literature on presence is not exhaustive but it shows that many authors consider and define presence in a different way. For the purpose of this research, though, we consider presence as a subjective mental experience of which the feeling of being physically located in a mediated environment is one of its core elements.

Realism

The concept of *realism* is very important for this project because one of its objectives is to develop spatially referenced interfaces and content, which will be used to represent the real environment and selected elements of it in our application. The virtual environment should be as accurate as possible, corresponding to the real environment, and also aesthetically pleasant. Di Bias et al. support that in immersive environments in which users can move around, realism is a fundamental requirement. Realism is also relevant to the ability of reproducing the physical behaviours of the real world in the virtual environment (Blas et al., 2005). An important conclusion in the aforementioned publication is that the key factor in virtual environments is not realism but virtual presence, to which realism or high quality graphics are not absolutely relevant issues. Ferwerda argues that there are three varieties of realism and defines the criteria that need to met for achieving each variety. The varieties are presented in the following list.

i. Physical realism – in which the image provides the same visual stimulation as the scene;
ii. Photo-realism - in which the image produces the same visual response as the scene
iii. Functional realism - in which the image provides the same visual information as the scene

Although that the technology has evolved it is still difficult to develop believable virtual environments. Generally speaking, building a visually realistic environment influences positively the development of believable user experiences (Fraser et al., 2000). But the problem in developing a model that approaches total visual realism is the complexity of the real world. In a review that we accomplished during the LOCUS project we found out that there are several types of applications that can present the most important information necessary to give a believable experience even with lower levels of realism (Gatzidis et al., 2008). Another study suggested that the realism (e.g. texture quality) between the world and its elements should be consistent throughout the represented environment (Vinayagamoorthy et al., 2004). The experiment studied the impact of realism on the presence in an immersive virtual environment. Khanna et al. support that in visual realism, another important factor, just like geometric realism for example, is illumination realism and conducted a subjective experiment, which suggests that more realistic illuminations that include dynamic shadows and reflections are associated with higher reported presence (Khanna et al., 2006). This study was extended in order to measure the physiological responses (i.e. skin conductance and electrocardiogram) of the participants and validated the results of the initial one (Slater et al., 2009a).

In another study that focused on presenting route instructions on mobile devices, the participants suggested that the 3D model should be more detailed and realistic and the points of interest should be highlighted, even though that the 3D world quality was quite high (Kray et al., 2003). A very detailed 3D model, in terms of texture image quality and polygon count, would pose performance issues on the mobile device (i.e. affect immersion) due to the vast resources required. Therefore, another user interface for geographic representations must evolve (i.e. AR) that can offer photorealistic depictions, without utilising a lot of the system resources so that it will not affect the user's sense of presence. Furthermore, another issue with the use of 3D maps with a photorealistic depiction is that they provide low-abstraction levels because they try to simulate reality. The focal point of the majority of navigation maps is on the functionality rather than on geographical accuracy in order to help the user focus on certain elements of the represented environment. By examining photorealistic 3D worlds, an observer's cognitive workload can be higher when he or she tries to analyse the presented information, especially in a mobile device with a small panel (Kray et al., 2003). Consequently, the usability of the system is affected and it may also affect the decision-making process of the user (e.g. wayfinding task). In a very interesting paper about computer depiction, Durand maintains that "*non-photorealistic pictures can be more effective at conveying information, more expressive or more beautiful*" (Durand, 2002) and discusses the concept of non-photorealism, as well as the difference between *image*, *picture*, and *visualisation*, all in the context of computer depiction. Davis argues that photorealism and other highly mimetic forms are more suitable for affective expressivity rather than informational expressivity and that "*photo-mimesis may be suppressed, distorted or subverted in order to convey information more effectively*" (Boyd Davis, 2007). Furthermore, the author maintains that, in most cases, a photorealist depiction does not offer many informational benefits in contrast to a non-photorealist depiction and that "*depiction should be subject to the same decisive, goal-oriented considerations as any visualisation: it is all a matter of design*" (Boyd Davis, 2007).

The aforementioned research suggests that a balance must be achieved between the realism of the virtual environment, the immersion potential that the selected technologies offer and the overall sense of presence that the user experiences. In this research project, presence is considered as a contributing factor to User eXperience (UX) which is described in more detail in Chapter 3.3.6. Furthermore, we should develop abstract 3D models (i.e. generalised representations) for maximising performance but also for making selected features of interest more evident to the user. The techniques that have been used to generalise the developed 3D models are presented in Chapter 5.4.1. Furthermore, in the Extensive Evaluation we reduced the texture quality of the VE's buildings for comparing non-photorealistic (i.e. VR) and photorealistic (i.e. AR) interfaces on urban navigation scenarios. According to Davis, it is the responsibility of the overall system designer to find the balance between the previous concepts so that the framework offers, primarily, good utility and, secondarily, good usability to its users.

2.4.2 Augmented Reality

Research on Augmented Reality (AR) focuses on promoting highly interactive interfaces that are able to inform the users and dynamically cope with their experiences in the real world. This is accomplished by combining real world features with contextual information that describes the natural features and complements them in a way that is found useful for the user's decision-making process, dependent on the targeted application. The presentation of information no longer take place on the static display of a computing device but is overlaid on digital feedback that represents the surroundings. This makes the process of blending real-world characteristics with artificial information more natural and the resulting user interface becomes more efficient, descriptive and user-friendly.

Originally, a formal definition and identification of Augmented Reality as an independent field of study was given by Azuma (Azuma, 1995) (Azuma, 1997). These publications quote that augmented reality is a variation of virtual environments. The evident difference between VR and AR can be found while immersed in the synthetic environment. In VR the user cannot observe features from the real world whereas in AR digital information is superimposed over real objects of the world. Thus, the latter case does not replace reality but on the contrary, it enhances it with valuable information.

Azuma recognised 3 functional requirements that AR systems should fulfil in order to produce robust implementations (Azuma, 1997). These are:

i. Combining real with virtual information;
ii. Interacting in real time;
iii. Registering objects in 3D space.

The main functionality of AR is covered by the first requirement, which translates into synthesising and representing content from digital and actual sources. The second requirement instructs any AR system to react in real-time to any contextual changes applied on relevant real-world elements. In order to separate AR from other mixed reality domains, the third rule was introduced. It underlines that overlays should be combined with the real environment in 3D space, which implies transformations between several coordinate systems.

A classification of AR, in respect to other Mixed Reality (MR) technologies and concepts, has been proposed by Milgram et al. (Milgram and Kishino, 1994) (Milgram et al., 1994). The virtuality continuum, found in Figure 2-6, demonstrates a range of concepts that involve merging real and virtual environments, which is referred to generically as Mixed Reality (MR). The two extremes of this taxonomy are easy to comprehend and diversify because they describe two absolute environments, which do not embed characteristics from each other. Conversely, the two concepts that reside closer to the centre are more complicated to accurately describe and define, as they adopt elements from both extremes. Regarding this classification, AR utilises real-world elements as background and introduces artificial object representations, in contrast to Augmented Virtuality (AV), which uses virtual worlds as the background supplemented by photorealistic elements of the real environment.

Milgram and Kishino describe six classes of hybrid display environments (Milgram and Kishino, 1994). Broadly, they can be classified to optical and video see-through displays which can both blend real and virtual information. Video see-through mixes video and integrated graphics into the video stream of the environment that is being displayed, whereas optical see-through makes use of optical combiners to superimpose rendered graphics optically over the real view of the environment. Piekarski and Thomas (Piekarski and Thomas, 2003) mention various issues associated with both types of displays and later Papagiannakis et al. (Papagiannakis et al., 2008) classified them into three groups – (i) technological, (ii) perceptual and (iii) human factors. The types found relevant to this project are *video see-through* systems, which capture an artificial view of the real-world scene into manageable camera frames, augment these video metaphors with virtual contextualised information objects and present the blended view to the device display as a unified video stream. Most early AR implementations adopted Head Mounted Displays (HMD) for blending the visual representation of the physical world with virtual content. Optical see-through HMDs lay optical combiners in front of an actor's eyes, allowing the light to diffuse from the environment. Concurrently, they reflect the light, which presents any virtual objects, projected from head mounted monitors, so that the users visualise the unified outcome. The main difference between the two display classes is that video see-through systems combine live video streams acquired from a camera with computer graphics and present the result on the device screen, while optical see-through systems generate an optical image of the real screen including any additional graphics, which appears within the real environment or within the viewer's FOV while observing the environment. Both techniques have distinct advantages and disadvantages which influence the addressed application. An in-depth analysis of the differences between the two display environments has been accomplished by Bimber and Raskar (Bimber and Raskar, 2005).

A distinct advantage of the latest AR systems is that they enable users to ubiquitously operate in random and unfamiliar working environments without the need of accessing dedicated workbenches. The first wearable AR implementations, though, utilised bulky and custom solutions, which have proven to be impractical in many ways. The size and weight constraints of such systems, favourably based on laptop computers, are one of the factors that have discouraged personal use. Until recently, the processing capabilities for real-time AR operation were supplied only by this kind of devices. Nowadays, the CPU and graphic processing subsystems of portable computers approach the performance capabilities of desktop systems. The only restriction that does not allow portable and static computers to carry identical hardware is efficient energy management and battery technology. The previous reasons explain why most AR solutions have not obtained high commercial acceptance, in addition to the fact that such prototypes exceed the price limit that end-users may be willing to afford to receive the offered services.

The early definitions of Augmented Reality maintain that the digital entities, which can be fused in the real-world scene, are composed of one or more 3D models. Lately, though, after the development of several working AR applications in various domains, the core definition has evolved. Wagner (Wagner, 2009) provided a historic background of AR developments since 1968 for the 9th International Symposium on Mixed and Augmented Reality (ISMAR). Nowadays, the definition of AR is not so strict and allows 2D digital entities to be embedded in the virtual environment. These 2D elements may consist of text or images that can describe selected elements of the real world. Furthermore, newer definitions may also include 2D animating content as the virtual element of an augmented space and visual information retrieval as core functionality of an AR system. In most cases, although recent systems enhance reality with virtual content, it is difficult for the end-user to conceive them as element of the natural world and, consequently, reduces the immersion capabilities of those systems. As we have seen in the previous subchapter, immersion is considered as an integral feature of Virtual Environments. Reid et al. support that "*(...) the stages of immersion identified in video games can be applied to location-based experiences but the prominence of the real world environment means that the immersed states are short with continual dipping between the parallel worlds of the digital and physical*" (Reid et al., 2005). To top all these, several approaches have been proposed to track the device's 6 Degrees Of Freedom (DOF), which is the essential input in any AR algorithm. In the following subchapter we have included a brief description of the most prominent approaches but namely these are – a) *fiducial* tracking, b) *feature* tracking and c) *location & orientation* tracking. As a result, the diversifications from the early definition of AR has assisted the development of several types of applications which connect several research fields, such as Location Based Services, Information Retrieval and Image Recognition, on a wide range of devices (e.g. desktop or mobile).

During the last decade consumers have been particularly interested in mobile communication devices, which come in the form of mobile phones, PDAs, Tablet PCs and Ultra Mobile Personal Computers (UMPC). These devices are continuously upgraded in terms of hardware components and software platforms, in order to meet the always-increasing consumer needs. As a result, we have recently started to observe the first mobile applications, which employ AR in everyday scenarios. Namely, a few descriptive and notable examples available for commercial use include *Layar*, *Wikitude*, *Metro AR* and *ZipReality Real Estate*, amongst several others. So far, the main disadvantage of these applications is that their functionality is fairly limiting, especially for those considered as world browsers. The reason is that they have the potential to satisfy more user information needs, if they process the full set of information that is available to them (Chapter 3.3.1.1). These applications, though, can be considered as the foundation that can support the development of better services that can satisfy precise user needs such as navigation. Furthermore, a detailed comparison of the characteristics between the software product developed for the purposes of this project with six other commercial solutions takes place in Chapter 6.

2.4.2.1 Traditional AR Approaches

Recent advances in the field of computer vision have produced a new set of technologies that provide photorealistic images, which can be queried and adopted by real-time context-sensitive engines. In contrast to VR implementations, traditional AR interfaces employ additional input from video cameras or HMDs. Likewise, the camera in our context-aware prototype is considered to be a source of valid and up-to-date context. Traditional AR techniques have been applied on LBS and pervasive systems as a means to register the user in the synthetic environment in accordance to real position and orientation behaviour (Liarokapis et al., 2006a). The primary focus of the evolved applications is the satisfaction of the user’s urban wayfinding (Feiner et al., 1997) (Thomas et al., 1998) (Reitmayr and Schmalstieg, 2004) (Romão et al., 2004) and pervasive entertainment requirements. An extensive review of pervasive game-like applications, which embed AR functionality, has been offered by Magerkurth et al. (Magerkurth et al., 2005) but the most notable was ARQuake (Wayne and Bruce, 2002). Most implementations utilise either a marker tracking solution or a natural feature recognition method.

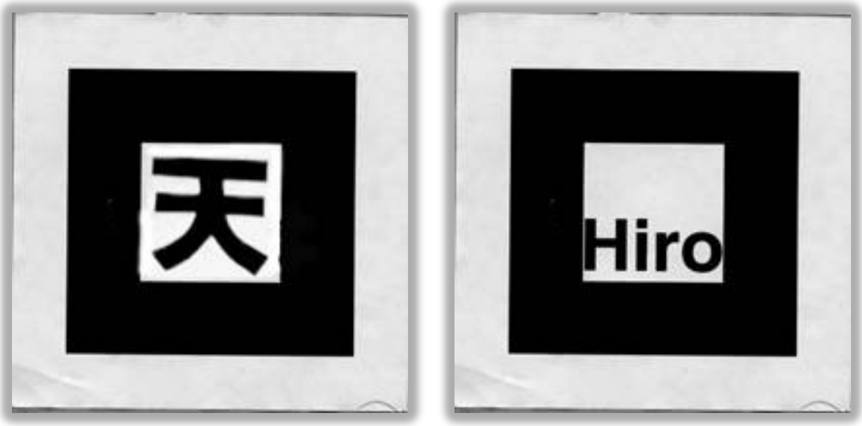


Figure 2-7: Sample ARToolKit Markers (HITLab, 2007)

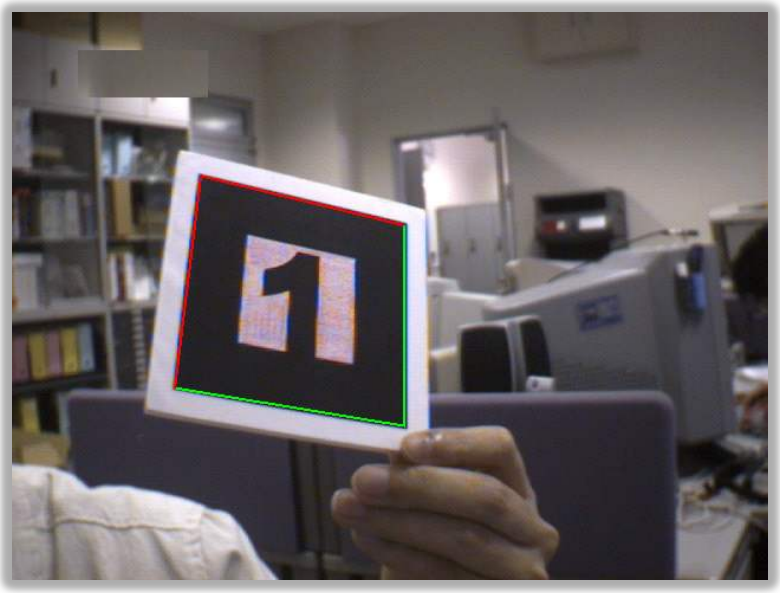


Figure 2-8: ARToolKit Marker Detection (HITLab, 2007)

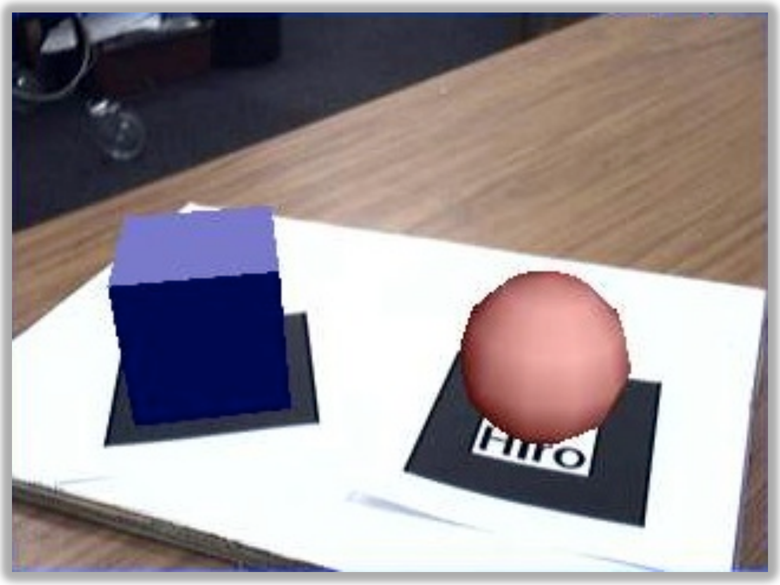


Figure 2-9: ARToolKit functionality - Superimposing 3D objects on markers (HITLab, 2007)

The fiducial point tracking approach depends on the detection of predefined patterns in the real world and the estimation of the camera pose compared to those patterns. This method enables the computation of the 6-DOF, which are required in order to accurately overlay digital information. Most patterns come in the form of planar multi-coloured symmetrical textures with some distinct characteristic for easier identification. Notable research in that field has been undertaken by Kato and Billinghurst, who produced the eminent open-source library called ARToolKit (Kato and Billinghurst, 1999). Since then, rapid evolution has been observed and supplementary software libraries have been developed. These libraries accomplish edge detection by calculating colour variations on the projected image. This method makes the operating algorithms vulnerable to image quality, light conditions and texture quality, which means that in certain scenarios tracking efficiency is severely affected (Papakonstantinou and Brujic-Okretic, 2009a). Moreover, because of the scarcity and difficulty of locating markers and populating them to the real world, further experimentation took place in the LOCUS project (Liarokapis et al., 2006c). This approach inherited patterns that already exist in the environment and identified them with pixel colour comparison techniques. An example of such an application that detects road signs is depicted in the following figures. Equally in this solution, the drawback is that the quality of the signs may not always be adequate for AR use, as with the lapse of time the texture of the objects may have suffered some sort of damage or deformation. Consequently, this method is restricted to operating in confined environments.

Figure 2-10: Pattern recognition of road signs a) Original image and b) Detected image

Figure 2-11: Pattern recognition of road signs - Superimposing 3D objects

Reitmayr has done exceptional work in the field of natural feature detection. Reitmayr and Drummond introduced a model-based hybrid tracking system that combines an edge-based tracker, gyroscope readings, measurements of gravity and magnetic field, and a back store of reference frames (Reitmayr and Drummond, 2006). The numerous sensors that are queried effectively render the use of this system exclusive to research purposes, in contrast to a subsequent research project (Schall et al., 2009). The combination of edge detection and feature recognition has been examined by Vacchetti et al. (Vacchetti et al., 2004). This method performed well but it needed additional preparation before initialisation. Another factor that makes the application of the aforementioned methods prohibitive for mobile phone operation is the processing power that is required by the CPU and graphic subsystem of the device. In 2006, Wang et al. released a software library, called *TinyMotion*, which detected a mobile phone user's hand movement in real time by analysing image sequences captured by the built-in camera (Wang et al., 2006). Lighting conditions and camera resolution is of utter importance, when targeting smooth operation, because the prototype compares the retrieved images. A coarse comparison between visual recognition modes can be found in the following table.

Recognition Mode	Range	Error	Robustness
Fiducial	0.5~2m	Low	High
Feature	2~10m	High	Low

Table 2-6: Fiducial vs Feature recognition mode (Liarokapis et al., 2006b)

The advantage of the feature recognition approach is that the operating range can be greater because it does not require preparation of the environment. Therefore, it can be applied when wayfinding is the focus of the application. However, the natural feature-tracking algorithm requires high accuracy of the position and orientation information, which can prove limiting. In contrast, the fiducial point recognition mode offers low error during the tracking process (i.e. detecting fiducial points). However, the limited space of operation, due to the need to populate the area with tags, makes it more suitable for confined areas and commentary navigation. The research suggests, however, that the combination of fiducial and feature recognition modes allows the user to pursue both wayfinding and commentary-based navigation into urban environments within a single application (Liarokapis et al., 2006a).

For the purpose of this project we developed a mobile AR system that uses location and orientation information in order to register the user's device in the environment and present information to the user about entities that reside in it. The reasons that we selected this approach, as well as its special requirements, are analytically presented in Chapter 5.5.

2.5 Location Based Services for Route Guidance

The boom of Geographic Information Systems (GIS) provided certain economical benefits for the organisations that required and adopted such functionalities. Technological progress in the fields of computing, spatial IT and digital information architectures has proved beneficial for GIS. One of the most promising applications seems to be LBS. Significant research is taking place and over the last few years elaborate experimental and commercial applications, which process spatial data, have been developed. The concept of *Digital Earth* has also been introduced and since then it has been quickly evolving. Moreover, as an extension to GIS, mobile GIS have been proposed. This concept was based on the global operability of Internet and the continuous familiarisation of people with mobile electronic devices. Raper et al. support that "*location based services (LBS) are computer applications that deliver information depending on the location of the device and user*" (Raper et al., 2007). In more detail, Chen et al. write "*LBS refer to offering information service based on geographical location for mobile terminal by utilizing GIS technology, embedded technology and wireless network communication technology under mobile environment*" (Chen et al., 2005). In addition, it is noted that because traditional desktop GIS cannot provide intelligent services through LBS's dynamic features, mobile GIS and its relevant technologies are utilised instead.

2.5.1 Navigation and Wayfinding in Physical and Virtual Worlds

Mobile computing and LBS has brought the infrastructure for providing ubiquitous route-guiding assistance to users. Golledge has separated the route guidance processes, for moving humans in physical environments, into two types (Golledge, 1999). The first process is *navigation*, which means that someone is deliberately making his or her way through some space, like a vessel in the sea. The second process is *wayfinding*, which involves selecting paths from a network. Earlier, in 1995, Golledge supported that the processes involved in navigation include "*(...) cue or landmark recognition, turn angle estimation and reproduction, route link sequencing, network comprehension, frame of reference identification, route plotting strategies (e.g. dead reckoning, path integration, environmental simplification and en-route choice, shortcutting). These processes are* *used in encoding environmental information for internal processes and use in wayfinding situations*" (Golledge, 1995). Conroy (Conroy, 2001) believes that this way Gollege tries to merge the concepts of environment cognition and spatial problem solving into a single definition of wayfinding. Therefore, in a latter publication (Golledge, 1999), he defines wayfinding in a more formal way. "*Wayfinding is the process of determining and following a path or route between an origin and a destination. It is a purposive, directed and motivated activity*" (Golledge, 1999). More precisely, it is mentioned that urban navigation is concerned with wayfinding in citywide transportation networks and it involves search and decision-making activities after processing the relevant environmental conditions and route choices.

In a more recent publication, Montello defines navigation as the coordinated and goal-directed movement through the environment (Montello, 2005). He maintains that navigation consists of two components, which are locomotion and wayfinding, and defines them. "*Locomotion is the movement of one's body around an environment (...) that is directly accessible to the sensory and motor systems at a given moment*". In contrast, he defines wayfinding as "*the goal-directed and planned movement of one's body around an environment in an efficient way*".

Darken, in order to assert that knowledge about human wayfinding in the physical world can be applied to wayfinding in virtual spaces, provides a new set of definitions. He considers wayfinding as the cognitive element of navigation, which involves tactical and strategic parts for guiding movement. Additionally he introduces *motion*, as "*the motoric element of navigation*" and defines navigation as the aggregate task of wayfinding and motion (Darken and Peterson, 2001). Similarly, Bowman et al. define navigation "*(...) as the complete process of moving through an environment. Navigation has two parts: wayfinding (the cognitive decision-making process by which a movement is planned), and travel (the actual motion from the current location to the new location)*".

As we have seen, the aforementioned definitions of navigation seem to come closer to Golledge's definition of wayfinding. Very interestingly, Conroy argues "*(...) that cognitive processes, considered in isolation of any movement through the environment, would be fundamentally meaningless. Therefore, since the act of travelling through an environment is a prerequisite component of our cognition of that environment, then the* *act of wayfinding must encompass both movement and cognition*" (Conroy, 2001). It seems that some researchers use the terms navigation and wayfinding interchangeably. Lately, though, the concepts have started to separate. Therefore, in this project, we are going to accept navigation and wayfinding as distinct terms, even though it would be pointless to have the cognitive element without any sort of movement in space by an individual.

Moreover, Darken and Sibert propose a classification scheme for wayfinding tasks, which are divided into the three categories that are presented below (Darken and Sibert, 1996b). The classifications of these tasks are mutually exclusive. However, they are often compounded into sequences.

i. *Naive search:* Any searching task in which the navigator has no *a priori* knowledge of the whereabouts of the target in question. A naive search implies that an exhaustive search must be performed.
ii. *Primed search:* Any searching task in which the navigator knows the location of the target. The search is non-exhaustive.
iii. *Exploration:* Any wayfinding task in which there is no target.

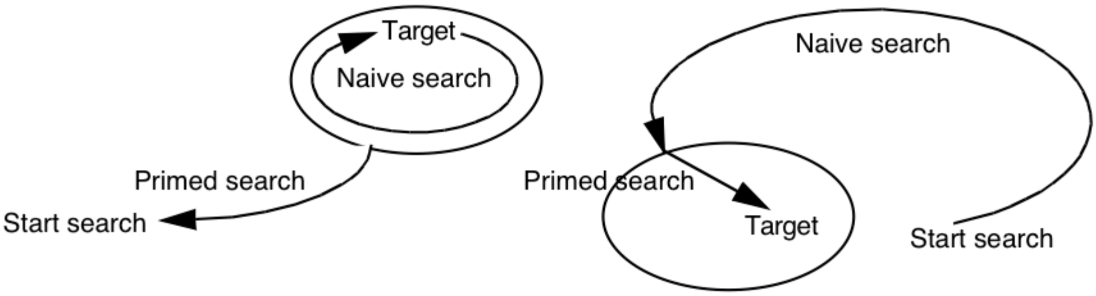


Figure 2-12: The hierarchical nature of compound wayfinding tasks (Darken and Sibert, 1996a)

In order to determine the relative effectiveness of the available navigation methods and tools, Darken uses the term *navigation performance* as the metric for evaluating such activities and this term is going to be adopted in this report as well. Two main factors may enhance navigation performance (Darken and Sibert, 1996a). The first factor is *spatial knowledge* that an actor has about the surrounding environment and the second is how well has the virtual environment been designed to cope with the wayfinding tasks. Spatial knowledge can be described in terms of three hierarchical levels of information (Howard and Kerst, 1981). *Landmark knowledge* (unique location or object), *procedural knowledge* (sequence of actions for wayfinding) and *survey* *knowledge* (interconnected topological information). Research has shown that the route-learning process in physical environments is similar to the route-learning process in virtual environments (Richardson et al., 1999). The second factor for improving the navigation performance is good *environmental design*, which directs the creation of specific elements that will be placed in a virtual world. These elements include *paths*, *edges* (i.e. boundaries between homogeneous regions), *districts*, *nodes* (route/path intersections) and distinct *landmarks*. Based on the reviewed literature, Darken has concluded into a number of organisational rules that can provide mental aid for an observer to hierarchically organise the spatial environment (Darken, 1995).

i. Dividing the large-scale world into small places;

ii. Organising places under a hierarchical structure;

iii. Providing frequent directional aids.

Moreover, in order to produce a flexible representation of the environment, a set of core map-design principles should be applied. The instructions suggest that the elements of the environment should be visible to the actor and that the user's position and orientation should be naturally represented on the interface. This means that the occupied user location has to be evident and the user orientation should be aligned with the direction that is actually lying in front of the beholder (Darken, 1995).

2.5.2 Wayfinding Tools

Although a number of mobile LBS prototypes have been developed to provide support for unprepared urban navigation, they do not conform to a specific design pattern, nor accomplish their task robustly. Current implementations have many critical differences between them, with the most distinctive one being the visualisation of spatial information. The most common applications to encounter are those that utilise 2D maps for interacting with the user. In the past few years several prototypes have appeared that establish advanced implementations of VR, AR and other kinds of novel GUIs. Namely, a few famous manufacturers, which have developed commercial navigation solutions, are TomTom, NDrive, Garmin, Magellan and Navigon. The major difference between them is how competent they are in achieving their purpose, which is to perform efficiently as navigation tools. Another fundamental difference is the amount and type of contextual information that can be effectively presented to the user (Liarokapis et al., 2006c).

An operational LBS based on the representation of 2D maps, which is well-known in its country is *EZ NaviWalk* from KDDI (Fuente et al., 2005). It is a pedestrian navigation service launched in Japan in 2003. Earlier, the *ActiveCampus* project explored wireless location-aware computing in a classroom and university campus area (Griswold et al., 2004). *ActiveCampus Explorer* tried to provide navigation support and tackle social tasks of college students. Careful analysis exposed novel behaviour and interesting results based on the relevance of proximity in social computing and the willingness to share location information between users. Another location-aware system that has been evaluated is *Campus Aware* (Burrell et al., 2002), a campus tour guide developed to allow its users to annotate physical spaces with text notes, in order to provide a sense of the activities that took place in the environment. The researchers concluded that location is more than coordinates and that touring is more than learning about a place.

To assist successful navigation, the development and use of a user's *cognitive map* has to be taken into consideration. "*A cognitive map is a mental representation of the environment*" (Darken and Peterson, 2001). It can represent humans' spatial knowledge, which resides in memory. Furthermore, Tversky (Tversky, 1981) maintains that the cognitive map of a person is resolution and alignment dependent and can produce asymmetries compared to the real world, which has also been verified in the LOCUS project (Liarokapis et al., 2006a). This is why there has to be a match of the Frames of Reference (FOR) of the cognitive environment with the registration of the user in the represented environment. In a VE a user can have multiple perspectives, ranging from egocentric to allocentric (Klatzky, 1998). These perspectives can be observed in the following figures. This makes it easier to comprehend the current location. Moreover, the use of distinct natural landmarks in a virtual world provides additional aid in the registration process. This is part of the knowledge that users have about the surrounding in addition to path integration and scene recall. These techniques have been used to enhance the process of self-localisation.

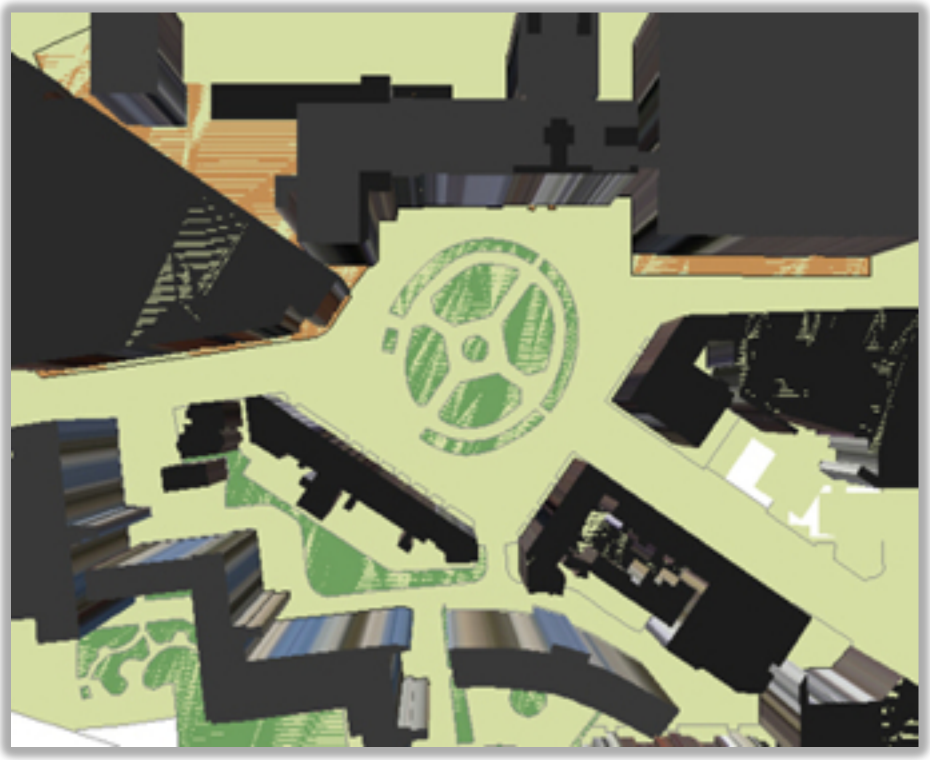


Figure 2-13: Allocentric Plan View of Northampton Square

Figure 2-14: Egocentric View of Northampton Square

By employing all the above methods, virtual environments present certain advantages over standard 2D interfaces, in the context of human wayfinding. Wood et al. (Wood et al., 2005) maintain that the increasing availability of geo-visualisation technologies (MacEachren and Kraak, 2001) to replace 2D maps has been driven by the assumption that 3D representations can provide more effective support in wayfinding scenarios than

2D maps. In their paper, Scaife and Rogers examine whether 3D representations are better than 2D and critique the disparate literature on this subject (Scaife and Rogers, 1996). Ruddle et al. maintain that virtual environments can assist the formation of a cognitive map more effectively than 2D representations, especially when landmarks are used (Ruddle et al., 1997). Furthermore, in a critical review of LBS, Raper argues that there are still open questions about the most effective representations for wayfinding, in terms of dimensions (i.e. 2D or 3D) and in terms of user interfaces (i.e. augmented reality or virtual reality) (Raper et al., 2007). In terms of user interfaces, the LOCUS project team has compared some characteristics of the aforementioned alternatives (Liarokapis et al., 2006b). Two relevant evaluations pinpointed a distinct advantage of mobile 3D maps when used in urban navigation scenarios. In the first evaluation, the researchers found that 3D representations of landmarks are easier to understand by the users compared to 2D maps (Rakkolainen et al., 2001). As mentioned earlier, landmarks can be used as reference points for effective registration in the environment and for wayfinding. Another interesting point in the aforementioned paper is that that users preferred a combined view of the environment rather than an explicit 2D or 3D view. The second evaluation obtained similar results but the researchers also noted that the developed 3D objects should be in high quality (Kulju and Kaasinen, 2002). On the other hand, Oulasvirta et al. conducted a field experiment, which produced interesting results. The purpose of the study was to examine the effectiveness of 3D over 2D maps when used in a mobile spatial task (Oulasvirta et al., 2009).

Innovative ideas may be triggered by considering test results of existing prototypes, such as *3D Layered, Adaptive-resolution and Multi-perspective Panorama* (LAMP3D) scene representation (Burigat and Chittaro, 2005), which focuses mainly on location-aware navigation issues, but with evident lack of orientation functionality. Results of a specific evaluation that have provided ground for further research include route guidance using a mobile device (Kulju and Kaasinen, 2002), following specific usability requirements decomposition for use on mobile devices as described by Vainio and Kotala (Vainio and Kotala, 2002), after examining a 3D city information system presented by Rakkolainen (Rakkolainen et al., 2001). The latter application, in addition to the one assessed by Laakso et al. (Laakso et al., 2003), utilises 3D representations of the real world, which enhance the spatial knowledge acquisition process of users. Moreover, a mobile multimodal system that employs speech as an interaction mechanism and supports indoor infrared positioning was examined by Wasinger et al.

(Wasinger et al., 2003). Initially, it was developed for navigation and exploration purposes, but later it was converted into a mobile shop assistant for describing the importance for context-aware systems to adapt their user interfaces to the surrounding environment (Wasinger and Krüger, 2004). Achieving maximum immersion of the user to the virtual environment is an important factor that needs special handling because just an abstract representation of the real world is not adequate for efficient navigation (Papakonstantinou, 2005). Distinctive landmarks can be embedded in a system in accordance to the results, which have been provided by Vinson (Vinson, 1999) and Burnett et al. (Burnett et al., 2001). Our previous work demonstrates that "*(...) the cognitive value of landmarks is in preparation for the unfamiliar and that self-localization proceeds by the establishment of rotations and translations of body coordinates with landmarks*" (Liarokapis et al., 2006b).

	3D Virtual Spaces	**Image-based City Spaces**	**Map-based City Spaces**
Feature	Detailed 3D Model	Photo-realistic images	Scalable 2D Map
World Dimension	3D	2-2.5D	2D
User Space Dim.	2D	2D	2D
Movable Space	Anywhere except obstacles	Along pre-defined routes	Anywhere
Interactivity	Very High	High	Low
Macro View	High	Not provided	Very High
Micro View	Very High	High	N/A
Real-time Information	Possible but expensive	Possible with live camera, relatively easy	Possible (GIS)
Development Cost	Very High	Relatively Low	Low

Table 2-7: Comparison of characteristics of digital cities in different spaces (Koda et al., 2005)

3 User Modelling & Requirements Engineering

This chapter presents the process, which has been adopted in order to realise the design of the framework architecture, and the milestones that were the driving force towards that direction. The issues discussed in this Chapter, and specifically in Chapter 3.3 and its subchapters, present several issues which can directly influence the outcome of the 1st and 2nd Research Questions of this project. Forming models of potential users based on their behaviour and familiarities, exploring relevant cognitive issues, modelling the immediate user environment and finding out how pervasive computing can mix with advanced visualisation techniques to promote collaboration are the user-related issues examined in this chapter. Furthermore, producing high-level framework models, which can be used to relate the previously acquired user challenges to the achievable system functionalities, is the last point described in this chapter.

Although there are several techniques which assist the process of designing and implementing a socio-technical system during the course of this project, it was found that committing to a single development strategy would not produce efficient results. The main reason supporting this statement is that the nature of the project was based on research initiatives and focuses on the satisfaction of certain requirements that are being posed by diverse domains. These requirements may affect higher-level goals, such as supporting the wayfinding process of a user, to lower-level goals like performance issues of the supplied solution. An additional reason for not being bound to a single development strategy is that the author did not envisage a dedicated system architecture, which would serve a specific application domain, but would prefer to produce a platform that could be easily extended to support supplementary, application-relevant functionalities. In effect, the software product of this research copes with the satisfaction of core requirements in relevant research fields, but also provides the foundation for new domain-specific applications to be developed.

3.1 System Development Methodology

Waterfall

The combination of two system-engineering methodologies has been selected to provide the means to achieve the required outcome for the system as a whole and for designing specific aspects of it. In more detail, the traditional *waterfall* approach was not deemed sufficient for the global design of the system because the only requirements that had been validated during initial research were those obtained through scientific publications. Furthermore, there were no actual stakeholders, in order to query them for their preferences, nor were there any specific actors to consult them on improving their current behaviour. This meant that frequent changes to the requirements definitions had taken place. As a result, certain deliverables did not conform to the instructed specifications and needed to be selectively discarded. An additional weakness of the waterfall methodology was the long timescales required to produce a working application. In this project we needed to follow a more agile process that included quick turnarounds and shorter development cycles.

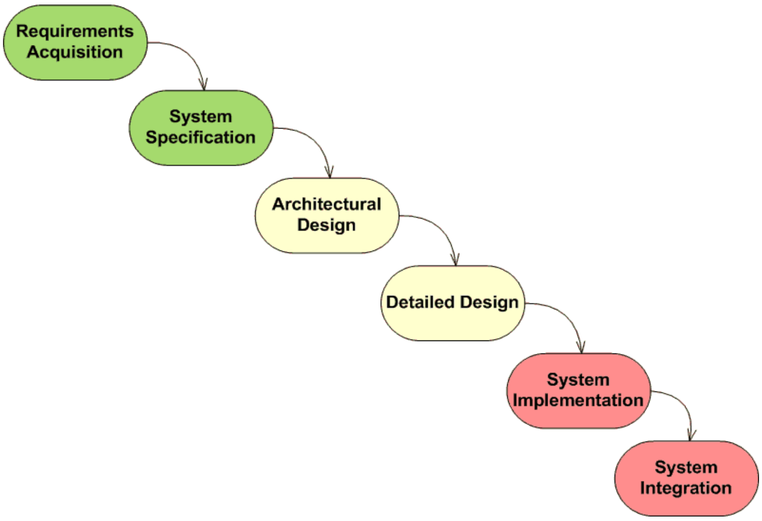


Figure 3-1: Waterfall Process Model

Rapid Prototyping

The solution for the global design of the system was found by following a custom-tailored *rapid-prototyping* approach. This method proved suitable because it could support our research-oriented progressive innovation. During the project timeline many prototype versions were developed. These repeated cycles of development and testing occurred every 6 to 8 months. Each prototype version focused on specific topics that had been previously identified and needed improvement. For instance, we had to examine the 3D interface in order to understand its advantages and disadvantages.

Doing this enabled us to comprehend which aspects of an AR interface would enhance the existing capabilities and for which reason. The loop that describes how rapid prototyping has been applied to suit the aims of the research is presented here.

- Initial research identifies research problems;
- Developing a prototype aims to solve these problems;
- Evaluation of the prototype produces informative results about the fidelity of the system;
- The results trigger new ideas for improvement;
- Further research fuses the enhancement of ideas;
- and so on.

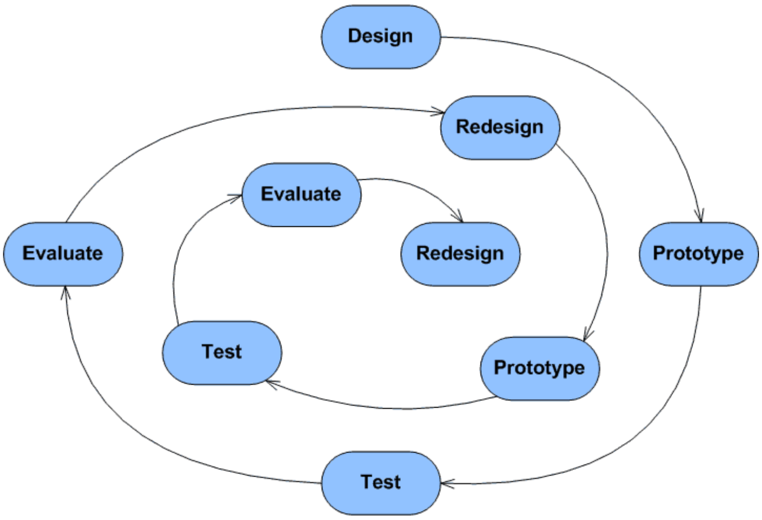


Figure 3-2: Prototyping approach

Applied Method

The combination of the waterfall and rapid-prototyping models can be found by extending the second statement of the previous loop. The development of our high-fidelity prototypes includes every stage described by the waterfall model; in essence, the conceptual design, the requirement acquisition, the actual system design and finally the implementation phases. The conceptual design is crucial because it produces answers to certain research questions of either minor or major importance. At this point we supply what we can offer in terms of cognitive solutions with the help of research conducted by other bodies or previously by the author. The break down and modelling of these solutions into quantifiable and measurable goals is of paramount importance. The application of hardware instruments and software means aims to meet the specifications of the desired product, which must satisfy the instructed goals. Thus, the waterfall approach has been followed during the development of specific system components, whereas the rapid-prototyping approach characterises the overall development methodology of our system.

3.2 Development Phases

Requirements Engineering

The Requirements Engineering (RE) phase has proven to be extremely valuable, for the research goals to materialise and for the author to become capable of testing the actual effectiveness of our prototype. Furthermore, it can inform certain decisions about further design alternatives. If we consider the complexity of the proposed system, which involves human actors and technically software-intensive sub-systems, it is obvious that an explicit method was required to guide the prescriptive process for systematic scenario-driven requirements engineering. The preferred solution was found to be *Requirements Engineering with Scenarios for a User-Centred Environment* (RESCUE) developed in the Centre for Human Computer Interaction Design at City University London (Jones and Maiden, 2005). The reason behind the adoption of this process is that it supports several elements and tasks that are required for this project, as well as a managed interaction routine between the iterations of the global rapid-prototyping process of our project.

RESCUE

The RESCUE method identifies four streams, each having a unique and specific purpose in the specification of our socio-technical system (Jones et al., 2004).

1. Human activity modelling provides an understanding of how people work, in order to baseline possible changes to it;
2. System modelling enables the team to model the future system boundaries, actor dependencies and most important system goals;
3. Use case modelling and scenario walkthroughs enable effective communication of precise and testable requirements;
4. Requirements management imposes quality validation on the acquired requirements.

The following paragraphs of this chapter contain information about the first couple of streams. In more detail, user modelling tries to identify current experience levels of potential users on certain domains that are relevant to our research. Following next, we observe certain user behaviour, which intends to assist the users in forming an understanding of the surroundings. In addition, an examination of the cognitive issues that influence the effective relation between user understandings and the information provided by the digital medium, for enhancing real-world interactions, is presented. The last part of user modelling studies the potential to enhance collaboration and entertainment between users by promoting activity in the real world. System modelling presents the conceptual and high-level feasibility of our system. In this part, we inspect how the developed system integrates with other systems that are required for successful operation. Finally, we present the core requirements, which control the functionality of the system, as well as describe the rationale of the users.

3.3 User Modelling

Observing how people perform certain tasks provides invaluable information that can be used to specify a future system, aiming to complement the existing human process. The researcher had to understand the cognitive and non-cognitive components, as well as the social and co-operative elements that are involved in the activity of an individual (Jones et al., 2004). There are several ways to collect such information from the participating users. During the course of the project we managed to organise focus groups whose main aim was the acquisition and elaboration of the system requirements. Another approach for identifying new needs was by conducting evaluations of specific aspects of the system with users in real-world conditions. Finally, targeted questionnaires proved useful, when we needed to gather specific user preferences and measure the familiarity level of the participants with current technologies and processes. Chapter 3.3.1 presents selected results that were gathered during the Requirement Acquisition Survey so that we can validate the users' expectations from a system, in terms of visualisation, interaction and collaboration features (i.e. 1st Research Question). The Questionnaire that accompanied the survey is presented in Appendix I. In contrast, Chapter 3.3.2 to 3.3.6 presents research conducted in order to fulfil the objectives of the 2nd Research Question, which are higher-level compared to those of the previous question.

The aim of this project is to assist its users to interact with each other, as well as with objects of the environment. Currently, several technologies may provide limited solutions to our problem, like VR, which offers improved user interactivity and immersion in a comparable to real environment. Such solutions have proven to be detrimental in terms of the physical and social activities performed by their users (Magerkurth et al., 2005). That is why we try to promote social collaboration between actors and information retrieval between remote entities by triggering real-world interactions. Part of these interactions includes search and exploration in an unfamiliar environment. Thus, our system needs to accommodate high-level goals, like navigation in an unknown place.

Several user activities needed to be modelled and examined. The employment of only one data-gathering method, in a single iteration, was not found sufficient for reproducing real-world user intentions, preferences and experience levels as well as for influencing activity alterations. In the following sections, an analysis of interesting information that has been gathered from users through various methods is presented. Following next, the conceptual design of the proposed system is described, including the high-level functional requirements and the boundaries that restrict its operation. Through this study, the reasoning that guided - this heavily-bound to development - research project, will become evident and the readers will be made aware of the solutions that we tried to apply to the most important issues which have been identified.

3.3.1 Illustrative Examples of Requirement Acquisition (User) Survey Analysis

Initially, the results drawn from a preliminary research survey will be presented. The approval for executing this survey was granted by the Research Ethics Committee of the School of Informatics, at City University London. The questionnaire was distributed via electronic means and aimed to cover a random sample of the population. Although it was difficult to achieve a truly random sample, a representative sample consisting of a few experts, but mostly of non-expert users, was assembled. The questionnaire was published after we had developed a working prototype and incorporated questions aiming to explicitly influence future development. The full RE questionnaire and a digital form of the collected data have been attached to Appendix I and Appendix II, respectively. The Exploratory Analysis of every recorded variable is provided in

Appendix III. The demographic background of the participants was not selected, but most of them have had higher education and this is reflected particularly by their occupation. A large proportion was in the academic field, either as researchers and lecturers or as students. The next 2 significant groups in order of popularity include participants with a background in engineering and practising psychology. The rest are individuals from diverse professional backgrounds. In total, we received 30 responses, 9 from females and 21 from males with a mean age of 30.17 years (Min: 19 & Max: 50).

The preface of the questionnaire was analysed, in order to put answers in greater context and form a user model that would describe their familiarity and expertise. This section describes and illustrates selected responses of the survey that can contribute in answering the 1st Research Question of this project. The full set of responses can be found in the Appendices. Through the preface, each participant expressed their approach in performing relevant to our research activities, which have been documented and presented in this section. Most participants have had extended experience with the use of mobile devices and this is reflected by the average time (9.68 years) that they had owned a mobile phone.

3.3.1.1 Familiarity with Mobile Devices

It was interesting to verify which mobile device the participants preferred to use for their daily communications, as well as with which platform they were most accustomed with. This was accomplished by examining the hardware and software specifications of all phones provided in this study. The following chart depicts the proportions of the participants' selection. Most participants (70%) were more familiar with the use of the Symbian platform, whereas the iPhone, Windows Mobile and RIM platforms scored 13.3%, 10% and 3.3% respectively. The examined models which were produced by NEC, Nokia, Samsung and Sony Ericsson supported the Symbian platform. In addition, Apple utilised iOS, HTC supported Windows Mobile and RIM made use of the BlackBerry OS.

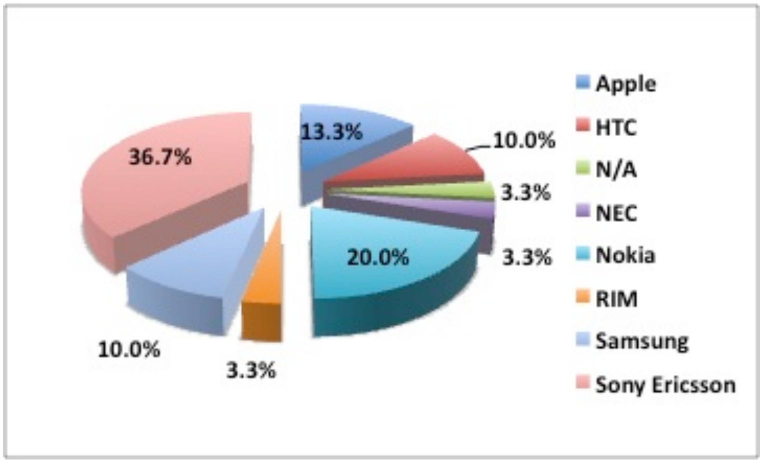


Figure 3-3: Mobile devices owned by survey participants

The participants identified which functionalities of their mobile device they mostly exploit, apart from placing and receiving phone calls. In descending order, 70% exchanged short (i.e. SMS) messages, multimedia (i.e. MMS) messages and emails, 50% managed their contacts and daily schedule, 40% listened to music and watched videos, 33.3% retrieved information from the World Wide Web (WWW) and 26.7% played games and transferred data with the help of the mobile phone. Interestingly, only 20% used navigation software or developed custom applications for their device. The results demonstrate that current location-based platforms do not fully satisfy the needs of the users and need re-engineering in order to offer better-formed contextualised services based on precise user needs. Furthermore, the proportion of individuals that manage their schedule through a mobile device reveals that valuable context about current or future activities is already available and it can be proactively processed to assist the users' tasks on a daily basis.

The following diagram presents the proportion of the participants who have been using the mobile's Internet connection to retrieve information or to communicate with other parties and how frequently they did so. It is obvious that most users either preferred not to utilise an Internet connection, due to reasons such as the cost of the service or because they had not found sufficiently beneficial mobile applications to support their needs. This gives an opportunity for advanced services to evolve, in order to bridge information exchange between static and mobile platforms.

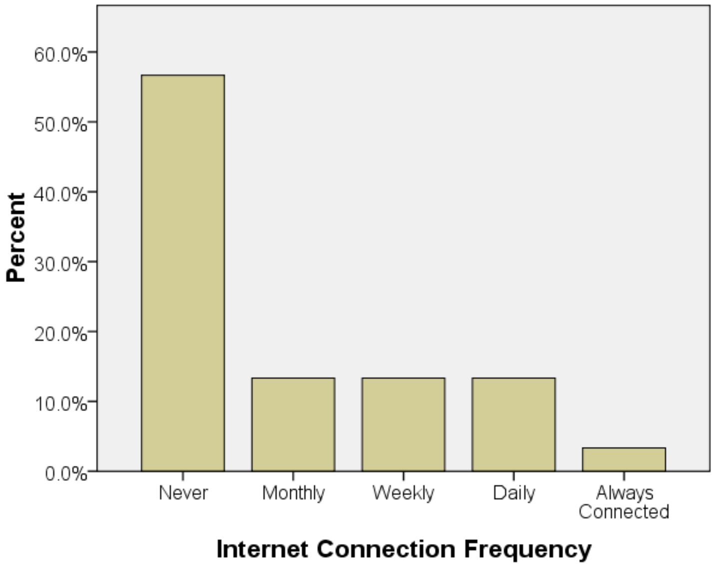


Figure 3-4: Frequency of Internet connectivity through mobile phones

The most important question in this section examines the level of assistance that mobile devices have been offering to their users during everyday activities and to what extent mobile users are dependent on their devices. The following histogram shows that the majority believed that carrying a mobile computing device has proven useful because it offers their favourite functionalities and increases their productivity.

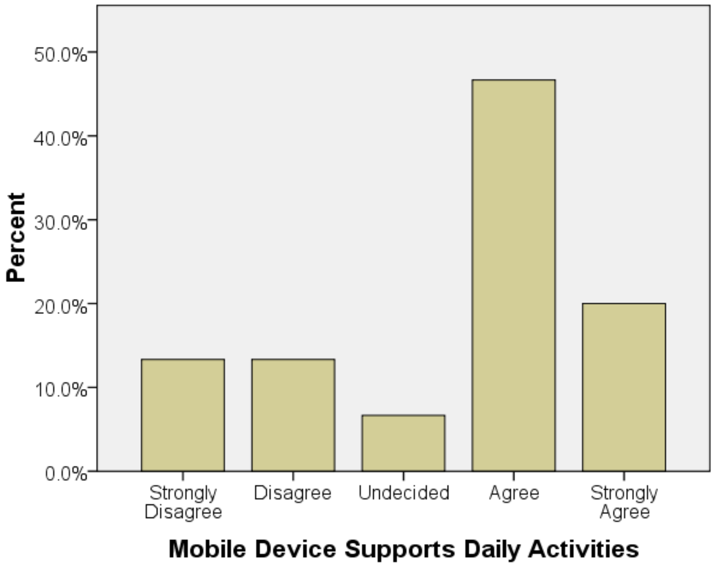


Figure 3-5: Mobile devices support in daily activities

3.3.1.2 Familiarity with Information Management

The questionnaire included two questions, which tried to detect the frequency with which users searched for a location and the frequency with which they navigated towards a location with the help of a computing device. It was found that there is a difference between the two results, showing that far more people search for a location but a lot less actually navigate there by using digital aids. Therefore, it seems that current commercial navigation mechanisms do not appeal to most users but certain features have been found invaluable for everyday use.

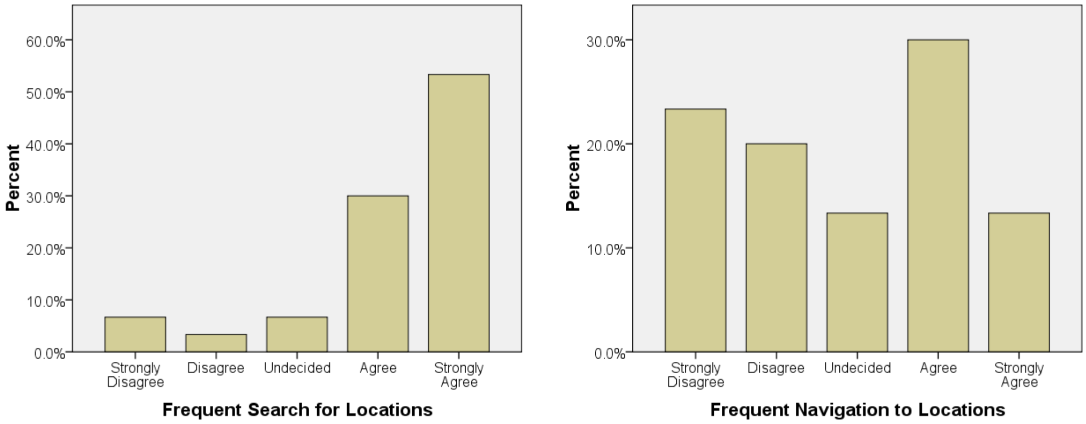


Figure 3-6: Frequency of a) Searching for a location b) Navigating to a location

An important aspect, which relates to privacy and the motivation of people to submit information to other parties, needed to be measured. The next histogram shows that people prefer not to release private information in exchange for personalised services. This means that mobile applications have to maintain users that do not wish to reveal confidential information, but still need to be able to receive the required services. Furthermore, users need to feel that they have explicit control on the selection of the 3rd parties that will receive their personal data, as well as on the type of context and the quality of information content that is going to be released.

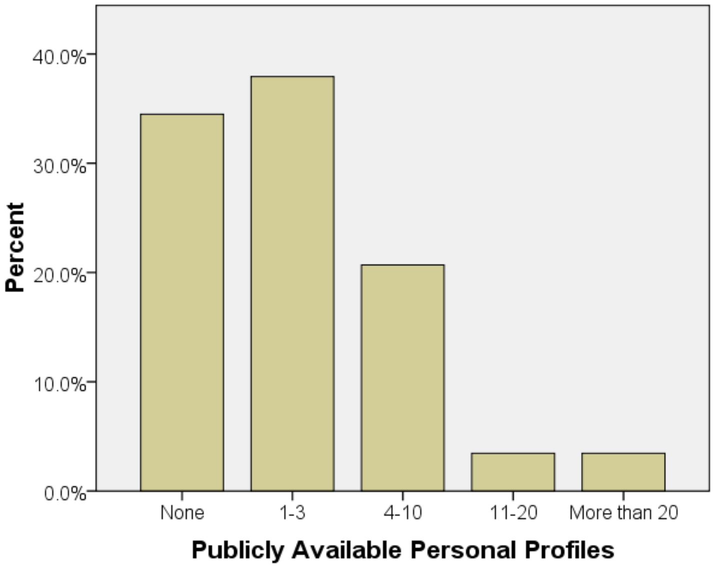


Figure 3-7: Number of personal profiles that each participant maintains

3.3.1.3 Familiarity with User Interfaces

The following two diagrams present user familiarity with advanced visualisation interfaces, which are optimised in a way that can assist the accomplishment of specific tasks. In our case this applies to the VR and AR interfaces. It seems that a lot of participants did not have any experience with either interface. This means that the introduction of an advanced interface in a mobile application could prove frustrating for the users because they should first become familiar with its functionalities and then try to accomplish any application-related task. Such interfaces, though, have proven invaluable for accomplishing certain tasks and for providing high levels of interactivity and immersion.

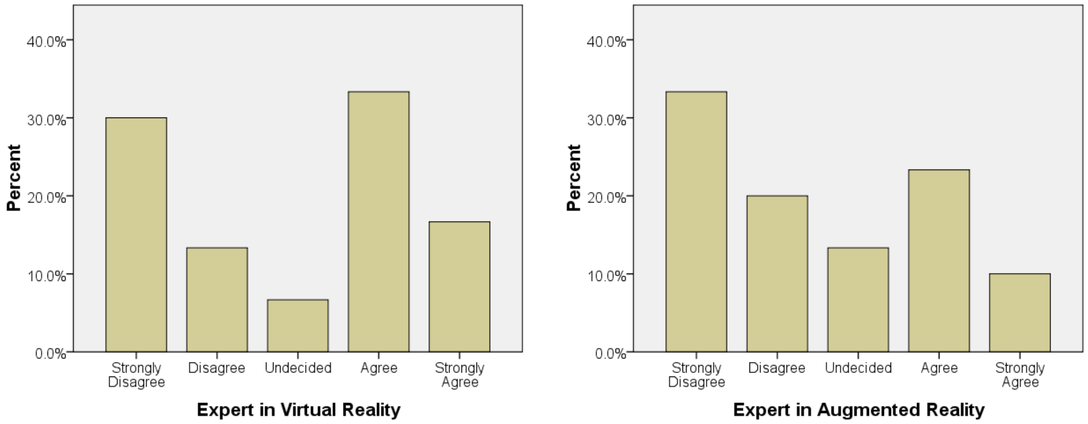


Figure 3-8: Participants' a) VR expertise and b) AR expertise

3.3.1.4 Familiarity with Interactive Applications

Currently, there are several types of applications that pursue innovative interaction methods with their users. The most recognisable, though, are 3D computer games because most individuals have either observed someone playing or actually participated in such a game. These games can be considered as a reference point, in order to verify the level of expertise with highly interactive applications. Furthermore, current gaming platforms have evolved and allow individuals to play, collaborate, or even socialise in the sphere of an entertainment application. This concept guided the formulation of a subsequent enquiry for the questionnaire. It tries to identify the skill level of the participants when they interact with other human parties and not the actual system or software agents. The results of these questions are presented below.

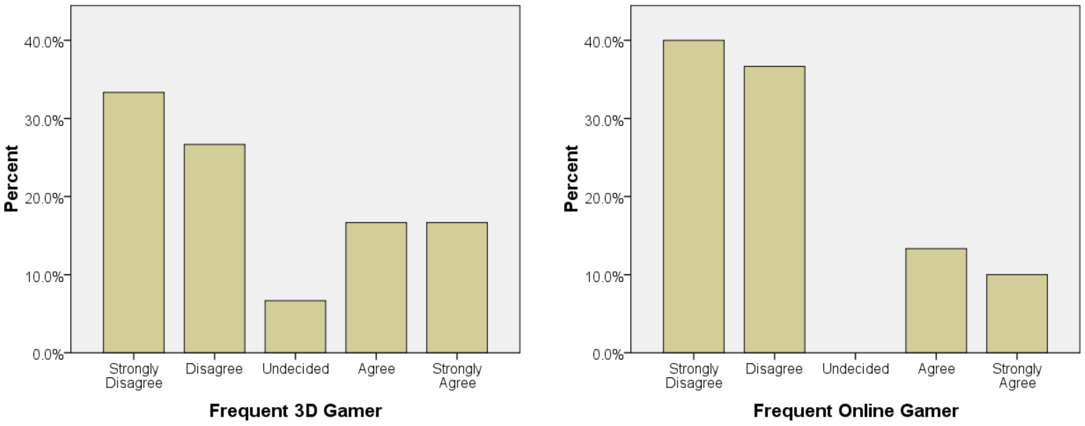


Figure 3-9: Frequency of a) 3D game playing and b) Online game playing

The previous diagrams illustrate that more people are frequent 3D gamers than online gamers. Moreover, it was found useful to verify the proportion of the participants, who have taken part in an online event that required their interaction either in a digital environment or in the real world. Out of the 30 interviewees, half reported that they have had some actual experience with real-time virtual interactions while 33.3% reported that they have participated in an event that required physical activity.

3.3.1.5 Familiarity with Collaboration Tools

The questionnaire incorporated a question that revealed the frequency of use of social networking tools. This way we could quantify the need for social communication of each individual. The following bar chart demonstrates that a large proportion of the people involved in the survey have intensively used such tools. This case does not distinguish between mobile and static users. We can assume, though, that most individuals prefer their PCs instead of mobile devices to establish such communications, according to previously expressed opinions.

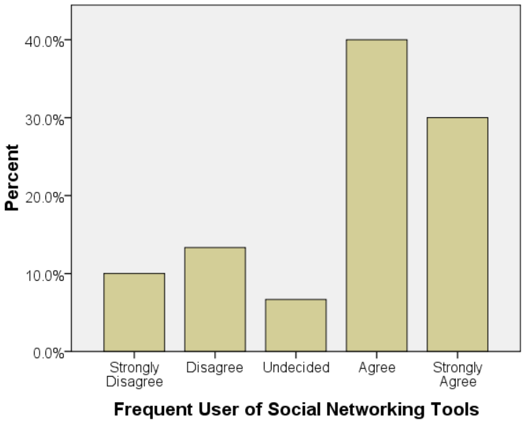


Figure 3-10: Frequency of social networking

The last question about collaboration tried to identify what fraction of the participants has taken part in an online event from a remote location. A descriptive example of this scenario could be a conference that is conducted at a distant place, but the participants have access to it through a networking tool like *SecondLife* (Linden, 2011). Out of the 30 participants, 60% have joined an event from a remote location, 36.7% have never done so while 1 provided invalid feedback.

3.3.2 Modelling the User's Environment

The previous section illustrated that current technological advances have provided users with the ability to remotely access a vast volume of data by utilising mobile computing solutions such as their mobile phones. These devices, when coupled with context-sensitive sensors, may offer advanced functionalities, which can be exploited to satisfy diverse user needs. The prospects required to satisfy these user needs is examined by the 2nd Research Question of this project and this section intends to contribute towards their realisation. Even though the volume of available information which users have access to is huge, it may prove off-putting because of the lack of a medium that would filter and manage it in a way that it would actually assist them with the current task and not overload them with irrelevant choices. That is why there is the need to engineer an innovative solution, which could be applied to currently widely accepted hardware (i.e. mobile devices). These solutions can be found in the development of new ergonomic and user-friendly interfaces, which would represent relevant to the current task information, in a straightforward way that could assist the decision-making process of the user. To produce interfaces that can adapt to contextual changes, but without becoming more complicated, an account of the users' mental model and personalisation preferences has to be taken into consideration. An element of the process to support personalisation tasks is by making users aware of their current spatial context. That is why the resulting interfaces need to be geographically referenced and capable of reflecting changes, which take place in the immediate environment or any relevant objects (i.e. POIs) of it, as well as to accurately represent the actual situation that is being experienced.

This project involves the development of a few novel interface paradigms, which try to cope with the issues discussed in the previous paragraph. The most simple and recognised solution is the 2D map representation of the environment. Non-digital maps have been extensively used by the public and especially by geographers to provide the means to locate themselves or other POIs of certain value. In addition, this medium has been widely used to assist any potential navigational needs of its users. Several software publishers (e.g. Google) have developed 2D map-like applications, which have tried to match the existing functionality of current non-digital maps, by exploiting the advantages offered by computing devices. Recently, extended versions of such applications have reached the mobile realm and started influencing the spatial knowledge of their users during common everyday activities, but without evident acceptance, as observed in the previously described survey analysis. Probably the most important factor, which has not allowed this digital 2D interface to meet the success of its non-digital equivalent, is that it offers only an allocentric (i.e. bird's eye) representation of the environment, which is useful for locating objects, but does not provide adequate cognitive navigational support. Research in related disciplines has identified several issues that apply to this concept. Chincholle observed that while maps can be extensive, mobile devices cannot support this attribute (Chincholle et al., 2002). Moreover, several features of interest in the environment, which are accurately represented on a mobile device, have proven to be of less importance for a mobile user, when observed on a map (Hirtle, 2003). This decreases the cognitive value of landmarks that effectively support wayfinding tasks such as search and exploration.

Mixed Reality techniques for visualisation and interaction have offered potential solutions to the specialised applications, which require the user to obtain an egocentric perspective of the environment. An extensive review of several geo-referenced VR and AR applications was conducted by Fisher and Unwin, who underlined the importance of gathering spatial data, such as position and orientation, and representing it to the available interfaces (Fisher and Unwin, 2002). Accurate registration of the user on the selected interface is a complicated process because there are several data aggregation issues invoved. To these issues, we can also include the processing performance of mobile devices, as well as their multimedia capabilities, which call for special handling. Matching the orientation of the available interface to the real-world perspective has proven to be valuable for the user after examining the cognitive study performed by Tversky (Tversky, 1996). Furthermore, the importance of acquiring and applying precise spatiotemporal data on a technological system, which offers navigational services through AR interfaces, has been demonstrated by Brujic-Okretic (Brujic-

Okretic, 2003)**.** In more detail, AR interfaces mix real-world information generated by a video stream with virtual elements and present visual feedback in a shared space (i.e. mobile device screen). This is accomplished in real time and by examining static or, lately, dynamic data. The main functionality of traditional AR applications is to calculate the position and orientation of the observer in relation to a specific POI (i.e. static marker) and to accurately superimpose digital information in the synthetic world. Thus, visual registration between the virtual and real-world objects directly influences the effectiveness of such applications. The evolution of global position tracking systems and the development of Micro-Electro Mechanical Systems (MEMS) sensors have provided the ability to convey the traditional functionality of AR applications, in order to ubiquitously operate even in unknown environments. This has been accomplished by replacing the source of context, from the existing visual tracking systems (i.e. object recognition or pattern matching) with the newly introduced sensors (i.e. spatial data). Following this process formed new applications, which can offer several services to their users ranging from egocentric navigation to situational awareness and remote information retrieval. The introduction of such systems to mobile platforms has offered new opportunities for users that require such features, as well as new concerns to be examined such as those described by Feiner (Feiner, 2003).

3.3.3 Representing the User's Environment

Mixed Reality interfaces are a very promising solution for LBS because they remove the need for users to make cognitive transformations from allocentric to egocentric perspectives, which is the case for 2D map-based navigation applications. In Chapter 2.5.1 and 3.3.4 of this report, we describe the cognitive processes which are required by a user to in order to acquire spatial knowledge and form spatial cognition (Richardson et al., 1999). *Landmark knowledge* can be greatly influenced by modelling an object of the environment and presenting it in 3D space. Two-dimensional representation of objects, especially smaller ones, can be confusing when presented in a 2D map. Furthermore, *route knowledge* involves an egocentric directional mental representation of the route from the beginning to the end. The last element of the acquisition process, *survey knowledge*, is based on the allocentric topographical (i.e. plan view) representation of the route based on the previous two elements. Mou et al. support that human navigation must depend on both egocentric and allocentric representations of the environment (Mou et al., 2006). Two-dimensional maps offer only a bird's eye view (i.e. allocentric, plan view) of the environment, whereas AR interfaces offer only a first-person view (i.e. egocentric). The only visualisation interface that can present the environment from various perspectives, including egocentric, allocentric and in-between views is Virtual Reality. A crucial disadvantage of VR is that it is very time consuming and expensive to produce 3D models of adequate verisimilitude, especially for larger environments. In addition, this process cannot be accomplished on the fly. Two important characteristics when simulating the real world in a 3D model are precise position coordinates and high-fidelity textures of the facades of the represented entities. A solution to this problem comes from Augmented Reality, which is the technology that can photo-realistically represent real space from an egocentric perspective. Consequently, peoples' navigational needs can be more effectively supported in a functional LBS by utilising a combination of Mixed Reality techniques. The advantages of this combinatorial use are revealed when egocentric AR is used to acquire survey knowledge, while VR supports the other two knowledge acquisition methods.

Meng and Reichenbacher support that while developing a user-centred mobile map targeted for small display devices several trade-offs have to be made (Meng and Reichenbacher, 2005). Some of these questions have been examined in this research project. Those are (i) the alignment of map orientation with the moving direction, (ii) the determination of the map scale in accordance with the moving speed, (iii) the maximally allowed visual load on a mobile display device and the minimum amount of information required by the user for a certain moment, (iv) the maximum number of visual signs a user can recognise within a certain time limit and the minimum number of information units he can efficiently remember and, finally, (v) the conventional design solutions (e.g. topographic map) and egocentric presentation styles.

Several researchers have explored in detail the advantages and disadvantages between egocentric and allocentric presentation styles, especially in the context of navigation (Klatzky, 1998) (Mou et al., 2006) (Oulasvirta et al., 2009). Egocentric navigation in an environment results in more accurate orientation and route estimation compared to exclusive allocentric (i.e. topographic map) navigation. Furthermore, Meng supports that the egocentric design can be regarded as a typical personalisation approach and that the individual user profile is embodied as the ego centre in the corresponding map (Meng, 2005). In that paper, Meng addresses the importance and necessity of egocentric geovisualisation, in contrast to the traditional allocentric map design for the purpose of promoting personalised services to the users. Dransch supports that egocentric representations can be a good basis to select and present the most relevant spatial information to an acting person (Dransch, 2005), whereas Reichenbacher explores the adaptation of mobile maps to current user activity (Reichenbacher, 2005). The aforementioned publications maintain that egocentric representations of the environment can prove valuable for mobile users, especially if the user's context is processed. Therefore, an egocentric visualisation interface is considered indispensible for being embedded in our mobile context-aware system.

Consequently, we have implemented two egocentric solutions - a photorealistic Augmented Reality interface and a non-photorealistic Virtual Reality interface. Both are presented in more detail in following chapters of this report. The AR interface represents the environment exclusively from an egocentric perspective, whereas the VR interface in our system supports three visualisation modes; (i) *egocentric straight angle (i.e. eye of the beholder)*, (ii) *allocentric oblique angle* and (iii) *allocentric plan view (i.e. bird's eye view)*. We consider the allocentric plan view in VR as a more elaborate substitute to the traditional 2D map representations which are replaced in the context of this project. The allocentric oblique perspective resides between the two extremes (i.e. horizontal to vertical views) and has certain advantages and disadvantages compared to the other perspectives, especially in the context of wayfinding. After conducting an experiment, Ohmi concluded that the acquisition of a cognitive map is not improved by presenting more content in an oblique perspective, even though performance of wayfinding is facilitated (Ohmi, 2000). The evaluation of our system provided indications about the effectiveness of each perspective in a wayfinding scenario.

At this point, it is important to clarify that one of the features of the developed system is that it reacts to real-time user context. This means that all developed interfaces and, consequently, the available visualisation perspectives respond to context changes. In see-through AR, the position and orientation of the mobile device is continuously calculated so that the relevant features of the immediate environment are superimposed on the device display. Therefore the heading of the camera always matches the heading of the user (i.e. forward-up) (Aretz and Wickens, 1992). In contrast, for the reasons described in Chapter 4.3.4, we have implemented additional functionality for every perspective in VR. The application can operate either in *sensor-controlled* or in *user-configurable* modes.

The normal sensor-controlled (i.e. context-sensitive) execution in every perspective of VR is described in detail in Chapter 5.4.3 but we are going to mention it briefly here as well. (i) In the VR egocentric perspective, the camera's position and heading in the VE matches the position and heading of the device in the physical environment even when the user moves. (ii) In the VR allocentric oblique perspective, the camera's position is slightly raised and moved behind the sprite that simulates the user's position in the VE. Heading in the VE matches the user's heading in the physical environment but there is a slight declination on the pitch angle so that both the sprite and the area that lies in front of the user are visible at the same time. Whenever the user moves or rotates in the real world the camera follows him or her accordingly in the VE. (iii) In the VR allocentric plan view, the camera's position in the VE is directly over the sprite that represents the user position in the VE. The position of the sprite in the VE simulates the position of the mobile device in the physical environment. The heading of the camera in the VE simulates the heading of the device in the real world but the pitch is vertical so that the user is positioned at the centre of the image. Whenever the user moves or rotates in the physical environment the camera moves over him or her accordingly in the VE.

Due to the fact that the virtual environment (i.e. 3D model) resides on the mobile device, the user should be able to explore the 3D world by using the device input mechanisms. This feature allows the users to improve their decision-making process, especially during wayfinding (Papakonstantinou and Brujic-Okretic, 2009a) because they can view the world from an arbitrary location. The users can switch to the manual operational mode so that they can travel around and survey the environment that is represented by the 3D world before visiting the potentially new locations. This mode of operation is only available in VR. Browsing the virtual environment can take place by using the three perspectives which are also available in the sensor-controlled mode. Namely, the egocentric perspective, the allocentric oblique perspective and the allocentric plan view. The users can select the desired visualisation view of the environment, through the user interface controls, and move in virtual space either by using the touch screen of the device or the navigation buttons (i.e. up, down, left and right). This operational mode does not accumulate measurements from the sensors, which means that the actual position and orientation does not change according to context changes. By interacting with the input mechanisms of the mobile device, the users have full control of the camera in the VE. In the egocentric perspective they can explore the 3D world from a first-person view. In the allocentric oblique perspective they can browse the world in any direction but with a slight declination of the pitch angle (e.g. 45°). Finally, in the allocentric plan view the whole 3D world can be vertically explored from a bird's eye view just like a 2D map.

In every visualisation perspective of both operational modes, the camera's position (i.e. 3-DOF) can be adjusted in the VE - in the sensor-controlled mode according to the user location context and in the user-configurable mode according to the user manual input. On the contrary, in all visualisation perspectives of both operational modes, not all orientation parameters (i.e. 3-DOF) are directly accessible by the user. The only orientation parameter that is not directly accessible by the user or the sensors is the pitch angle. This means that in every visualisation perspective of both operational modes the pitch angle is constant. When the egocentric perspective is selected, the pitch angle is equal to 0°. Pitch angle suffers a slight declination (e.g. 45°) when the allocentric oblique perspective is selected. Finally, the pitch angle is equal to 90° when the user browses the VE from the allocentric plan view. Although additional custom visualisation perspectives could be supported by our application, such as those found in computer games and motion pictures, we did not explore this option further. The developed solution is one of the few applications that support three distinct visualisation perspectives for supporting navigation in a LBS. In the evaluation of our solution we compare the available technologies, such as photorealistic AR and low-fidelity VR, in order to examine the benefits of each. The evaluation tasks assisted us in making informed decisions about the suitability and applicability of each technology and perspective combination.

3.3.4 Cognitive Issues

During the course of the research, it was found important to examine the cognitive issues that are involved in the immersion of a user in a virtual environment, which simulates the real, and to model how spatiotemporal knowledge is processed and experienced by individuals. Collaboration with the LOCUS research team proved invaluable for examining these effects, which have been jointly published (Papakonstantinou and Liarokapis, 2007), after producing first-cut AR and VR prototypes, applying them to the existing urban navigation platform (i.e. WebPark) (Camineo, 2004) and evaluating how information affects potential user decisions. The following paragraphs present the cognitive user requirements and demonstrate that a single environmental representation technique or interface is not adequate to form a complete simulation of the user's surroundings for supporting their high-level goals. Therefore, this section assists in understanding which are the users' expectations of a mobile context-sensitive framework that may have several ubiquitous applications (i.e. Research Question 2).

An account of the user's cognitive environment is needed to ensure that environmental representations are not just delivered on technical but also on usability criteria. The key concept for egocentric geographic applications is the *cognitive map* of the environment held in metal image form by the user. After conducting a study on this issue Tversky (Tversky, 1981) came to the conclusion that cognitive maps:

- Have asymmetries (i.e. distances between points are different in different directions);
- Are resolution-dependent (i.e. the greater the density of information the greater the distance between two points);
- Are alignment-dependent (i.e. distances are influenced by geographical orientation).

Hence, calibration of application space concepts against the cognitive Frames Of Reference (FOR) is vital to usability. Reference frames can be acquired from the egocentric (i.e. the perspective of the perceiver) to the allocentric (i.e. the perspective of some external framework) (Klatzky, 1998).

People can have multiple egocentric and allocentric FOR and can interchange between them without information loss (Miller and Allen, 2001). *Scale by contrast* is a framing control technique that selects and makes salient entities and relationships at a level of information content that the perceiver can cognitively manipulate. Whereas an observer establishes a *viewing scale* dynamically, digital geographic representations must be drawn from a set of preconceived map scales. Inevitably, the cognitive fit with the current activity may not always be acceptable (Raper, 2000).

Alongside the user's cognitive abilities, improving their spatiotemporal knowledge is vital for developing ubiquitous applications. This knowledge may be acquired through *landmark recognition*, *path integration* or *scene recall*, but will usually progress from declarative (i.e. landmark lists), to procedural (i.e. rules to integrate landmarks) and then to configurational knowledge (i.e. landmarks and their inter-relations). There are quite significant differences between these modes of knowledge, requiring distinct approaches to application support on a mobile device. Hence, research has been carried out on landmark representation (Michon and Denis, 2001) and on the process of self-localisation (Sholl, 2001) in the context of navigation applications. This work demonstrates that the cognitive value of landmarks is that it assists the registration process of a user in an unknown environment and that self-localisation proceeds by the establishment of rotations and translations of body coordinates according to landmarks. Relevant research has also been accomplished on spatial language for direction-giving, presenting, for instance, that path prepositions such as *along* and *past* are distance-dependent (Kray et al., 2001). These results advise that the mobile application should enhance the user's knowledge and apply it in real wayfinding scenarios. Höll et al. (Höll et al., 2003) illustrate the achievability of this aim by demonstrating that users who had pre-trained for a new routing task in a VR environment made fewer errors than those who had not.

3.3.5 Collaboration and Entertainment Issues

The previous chapters described how advances in technology revolutionised the way in which computers, including mobile devices recently, allow the users to acquire and manipulate complex multifaceted information in real time and to interact with each other at various levels (Papakonstantinou and Brujic-Okretic, 2009a). This subchapter intends to further examine the higher-level user collaboration and entertainment issues, which can contribute to the expectations that users have from a context-aware framework (i.e. Research Question 2). The range of appealing applications is increasing rapidly and it spans across urban navigation, sudden events management and cultural heritage information – through to entertainment and peer-to-peer communications. Since the early video games in the 1970s (e.g. Pong), technological progress in AI and Computer Graphics (CG) engines has also affected the nature of user interactions, something which is particularly reflected in game playing patterns. New sets of technologies have not only managed to simulate games that were traditionally played in the real world, but have also enabled new types of games based on the exploitation of technological assets, which have proved to be equally enjoyable to the participants.

Recent applications, like *Crysis*, have utilised advanced graphic engines in order to provide high interactivity and immersion into the virtual world, but have also proved to be detrimental in terms of the physical and social activities of their users (Magerkurth et al., 2005), in comparison to traditional games. Similarly, mobile versions of such games have been developed, with reduced processing demands, but with evident potential to become mainstream products.

The introduction of concepts such as ubiquitous computing (Weiser, 1993) and mixed reality (Milgram and Kishino, 1994) have contributed towards the development of new methodologies for entertainment and provided the foundation to bridge the gap between independent and social user behaviour (Cheok et al., 2006). Consequently, there has been a natural expansion towards the spatial, temporal and social connotations that any solution needed to address, including the consideration of their advantages and disadvantages (Montola, 2005). Acquisition and management of quantifiable user-related and environmental parameters - that we refer to as context - was found to be a means for achieving this kind of functionalities and for providing the resources to connect the real with the artificial world, where the game takes place. An analytical review has been published by Rashid et al., who describe applications that employ location context in mobile gaming scenarios (Rashid et al., 2006). The examined applications run on mobile devices, rendering them operational in any physical environment or while in motion. Similar systems that can explore virtual environments (Burigat and Chittaro, 2005), augmented environments (Reitmayr and Schmalstieg, 2004), or both (Liarokapis et al., 2006a), have been introduced with application in navigation. The advantage of these engines is that they can combine virtual and physical space and assist the decision-making process of each user through advanced user interfaces.

Several attempts have been made to classify entertainment solutions, which are partially persistent in a computer-generated environment and partially deployed on the real surroundings. Most of these classifications loosely use the term *Pervasive Game* as discussed by Nieuwdorp (Nieuwdorp, 2007). Interesting reviews of such applications have been given by Magerkurth et al. (Magerkurth et al., 2005), who expands on the concept and defines various systems with the main goal set to be the amusement of their users. The term *Ambient Game* has been used to define solutions that conform to a specific balance between the commitment required by the user and the distance that is travelled (Eyles and Eglin, 2007). *Trans-Reality* has been used to describe a subset of ubiquitous applications that use MR techniques for implementing various genders of games (Lindley, 2004). All examined types fit to the conceptual frameworks drawn by Walther (Walther, 2005) and Hinske et al. (Hinske et al., 2007), as well as any applicability concerns presented by Capra et al. (Capra et al., 2005). A description of several issues in the underlying technical infrastructure, in terms of the interfaces, is provided by Broll et al. (Broll et al., 2006) and in terms of subject localisation by Benford et al. (Benford et al., 2004). Ultimately, an evaluation platform capable of examining ubiquitous entertainment solutions has been released by IPerG, an EU-funded project (Benford et al., 2005).

After examining the results produced by the aforementioned research, new considerations relevant to the social behaviour of a user in pervasive entertainment scenarios have surfaced. Users are not particularly interested in how any underlying technologies cooperate in order to achieve the promised outcome. Their main concern is the feeling that they receive through the overall experience. People tend to have a holistic view about their experience. They are not only interested in the moment they interact with a device, but how it is embedded in the broader field and assists them with their activities. In order to enhance and influence user experiences, potential applications need to offer increased aesthetics and sensation.

A way to improve the effect that a technical framework offers to its users, while supporting pervasive entertainment functionalities, is to employ real-time context variables and to register the user in a number of immersive environments, which simulate and enhance the real surroundings. This way, physical and a few social interactions that occur in the real world can be modelled and presented through the visualisation interfaces. The user interfaces should complement each other and ideally allow the user to obtain a customisable perspective of the environment. Egocentric, allocetric oblique and allocentric plan views can be achieved through the combinatorial configuration of 2D map-like representations, VR interaction engines, or custom-tailored AR interfaces. Additionally, further user activity in the real world can be promoted by running such systems on a mobile device platform. In this case, however, ubiquitous system operation needs to be supported.

The users expect the framework to provide a common platform that will promote further interactions (Papakonstantinou and Brujic-Okretic, 2009a). These interactions should take place either between the user and relevant objects of the simulated environment or between the actors of the system. In the latter case, the system needs to act as a mediator who provides the *rules of engagement* between the participating entities. These rules must be based on user-specified criteria (i.e. user context) and established through the interface presentation mechanisms. There are several potential capabilities of such interaction and visualisation environments, but we examine those that can literally or metaphorically, *bring people together*. This can happen by applying pervasive game-like scenarios, which will trigger social interaction between the participating entities. According to each scenario, further activity in the real world is necessary. Physical activity with the guidance of a computing device that handles context parameters is the basis of pervasive computing. Engaging users becomes more effective, when the application is executed in multiuser settings rather than in single-user mode. In comparison to independent modes of play, social interactions can intensify user engagement.

3.3.6 User-Centred Design

In the Introduction, we mentioned that one of the goals of this research project is to examine the commercialisation perspective of the developed product. Our mobile context-aware system does not only have to be useful and usable for a potential customer but it must also provide coherent and comprehensive User eXperience (UX) in order to be selected. This can directly influence the objectives of the 2nd Reseaech Question of this project. Figure 1-1 presents the model that was adopted for the development of our prototype system. It is obvious that the primary focus was based on the techniques and components (i.e. Implementation) through which the system performs its function. The secondary goal of the development effort is to describe the functions that the system may offer in the life of a customer and present how it can be useful to them (i.e. Role). The last goal, in terms of importance to the project, is to examine the sensory experience while a user is operating the system (i.e. Look & Feel). There is not any market, though, that a non-functional solution would be successful, no matter how well its interfaces are designed, if it does not have utility. Furthermore, the usability of our system depends on two factors (i) the *ease-of-use*, such as its functionality, visualisation and interaction features; and (ii) the *availability* of content and certain resources which can be accessible by the user, such as POIs, 3D models or other network users. The mobile device, which the application is going to operate on, is another factor that can affect a user's perception about the usability of our system. Therefore during the Extensive Evaluation of the framework, when we had to formalise and apply the product on selected Use Cases, we took into consideration the aforementioned issues and tried to present the best possible solution to the users.

In order to design an interactive system, we needed to explore the user needs. We mentioned in the previous paragraph that our primary objective was to combine the available technologies in order to produce a system that has good utility for its users. The secondary objective while designing the system was to meet some core usability goals and, finally, to enhance user experience by introducing advanced visualisation and interaction interfaces. The research fields of *Usability Engineering* and *User eXperience* are distinct concepts, each one having distinct goals and methodologies in the design of a system. Several researchers have extensively explored Usability Engineering (Nielsen, 1994c) (Norman, 2002) and User eXperience (Hassenzahl and Tractinsky, 2006) (Law et al., 2009), especially during the last decade when the users started selecting applications that are not only functional but also provide high levels of user-friendliness and satisfaction. The following figure presents the relation between utility, usability and user experience. The outer circles, named *Desirability* and *Brand Experience* are those describing the UX that a system provides.

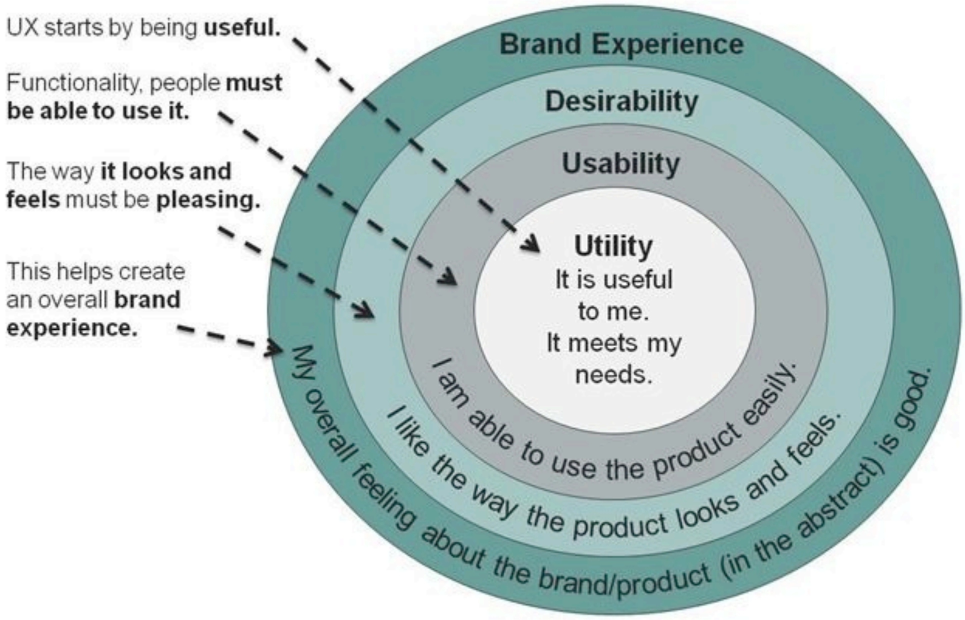


Figure 3-11: The Relationship between Usability and User experience (nngroup, User Experience Conference, Amsterdam, 2008)

Furthermore, the usability and UX goals have been included in international standards in order to support the development efforts of new products. ISO 9241-11 defines usability as "*the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*" (ISO 9241-11, 1998). ISO also defines user experience as "*all aspects of the user's experience when interacting with the product, service, environment or facility*" (ISO 9241-210, 2010). The usability and UX goals differ in terms of how they are operationalised. Usability goals are concerned with meeting specific usability criteria and UX goals are concerned with explicating the quality of the user experience (Preece et al., 2002). A new standard for software quality (ISO/IEC 25010, 2011) separates *quality in use* into (i) *usability in use* which defines usability as effectiveness, efficiency and satisfaction (ISO 9241-11, 1998), (ii) *flexibility in use*, which examines if the product is usable in other contexts, including accessibility, and (iii) *safety* which is concerned with minimising the undesirable consequences. Bevan (Bevan, 2008) offers a very descriptive list that describes the elements that influence the *quality in use* of a system.

- Quality in use
    - Usability in use
        - Effectiveness in use
        - Productivity in use
        - Satisfaction in use
            - Likability (satisfaction with pragmatic goals)
            - Pleasure (satisfaction with hedonic goals)
            - Comfort (physical satisfaction)
            - Trust (satisfaction with security)
    - Flexibility in use
        - Context conformity in use
        - Context extendibility in use
        - Accessibility in use
    - Safety
        - Operator health and safety
        - Public health and safety
        - Environmental harm in use
        - Commercial damage in use

Bevan also supports that *quality in use* contrasts with the interpretation of usability as the features of the user interface that make a solution easy to use (Bevan, 2008), such as Nielsen's classification which supports that a product can be usable, even if it has no utility (Nielsen, 1994c). But for having system usability, user interface usability is an important requirement. Furthermore, UX can be described by four elements which are listed below (Bevan, 2008).

i. *UX attributes* such as aesthetics, designed into the product to create a good user experience;
ii. The user's *pragmatic* and *hedonic UX goals* (i.e. individual criteria for user experience);
iii. The actual *User eXperience* when using the product;
iv. The measurable *UX consequences* of using the product (i.e. pleasure, and satisfaction with achieving pragmatic and hedonic goals).

Hassenzahl perceives interactive systems along two different dimensions (Hassenzahl, 2003). *Pragmatic quality* is the capacity of a solution to support the achievement of certain functional goals (e.g. finding a POI) and explores its utility and usability in relation to potential tasks. *Hedonic quality* is the capacity of a solution to support the achievement of goals that are related to a user, like why does somebody use a specific application. This classification is particularly useful if we, also, take under consideration the stakeholder of the requirement. In most cases, the owners of the system mostly worry about pragmatic goals, whereas end-users need to have both types sufficiently covered to be satisfied. Therefore, in the design of our system, we defined pragmatic goals in terms of requirements. What does the system have to do in order to have good utility for its users? In Chapter 3.4, we present these requirements and model the association between the actors of the system. Furthermore, Chapter 3.4.3 models the main functional requirements, as well as the soft-goals that received particular focus during the design of the system and what is required in order to achieve them. Soft-goals are considered the hedonic goals in our system design, which will contribute positively towards the users' satisfaction. Soft-goals are depicted as clouds in the SR diagrams of Chapter 3.4.3.

We believe that the use of advanced user interfaces and the process of real-time context (e.g. user and other types) will contribute towards providing good UX, better than those provided in the currently available LBS. It is not sufficient just to apply such technologies on a system if there are not intelligently managed so that they can satisfy the user needs. These user needs have been collected either through our subjective user *Requirements Acquisition Survey* (Chapter 3.3.1) or though the research accomplished (Chapter 3.3.2 to 3.3.5) during the course of this project. During the initial survey (Chapter 3.3.1), we made an effort to collect usability and UX goals that were found important for the participants when using a system like the one we have been developing (i.e. usability heuristics). Then we translated these goals to requirements that can be met by using the available technologies and real-time context. These requirements, including the prerequisite information and technology combination, are presented in Chapters 4.1 and 4.2. Finally, during the Extensive Evaluation, except examining the utility of the system, we also examined certain aspects of the usability and UX provided by developed framework so that we could make an informed decision on which interface is better for accomplishing the user goals.

3.4 System Modelling

Inspection and Verification

The main reason for using a formal specification language to describe the analysis and design of our system is that it enables inspection and verification of the acquired requirements. Additionally, given a system specification and a programming language definition, it is possible to prove that the application conforms to the specification. Furthermore, the increasing importance of component reuse in the development of the system, by reusing smaller components such as ActiveX controls, means that further customisation of the existing solution, in order to satisfy new user needs, is possible. Another advantage of specifying requirements is that it enables us to identify and generate evident test cases, thus providing better lifecycle guidance, as well as to pinpoint potential applications for our platform, such as navigation and entertainment.

Modelling Requirements Dependencies

The description of the *context diagram* reveals the importance of dependencies between actors involved in the system operation. A common problem that has been identified is that some stakeholder and user requirements contradict each other and there is not a single solution, which can satisfy all of them, in every possible application of the framework. Therefore, in this project we had to make complex trade-offs between certain requirements, in order to specify the optimum solution that satisfies the majority of user needs, but without being bound to any specific application-related solution. Thus, we have produced a platform that can be customised to serve more advanced requirements, generated by potential stakeholders as well. During the design of *Aura*, the focus was explicitly drawn on the decision-making techniques, but we also examined the requirement-related factors that influence this decision-making. Some typical factors that have been considered were the requirements priorities, importance, risk, cost, delivery time, and dependencies on other relevant high-priority user requirements. Because requirements dependencies are complex, in order to effectively understand them, it was found essential to model dependencies between actors, their goals and tasks.

*i** Goal Modelling

Eric Yu (Yu and Mylopoulos, 1994) (Yu, 1997) suggested an interesting approach for modelling requirements. He developed the syntax and semantics to support the modelling of goals and their relationship with tasks, resources and requirements, as well as other goals. A novel concept of the *i** approach is the uncoupling of hard goals from soft goals. Hard goals are functional requirements, which are either met or not met (i.e. arrows from a task to achieve a goal). On the contrary, soft goals are considered as non-functional requirements that diverse solutions may contribute to, either positively or negatively. Therefore, in *i** models, arrows show dependencies from tasks to the soft goal, with positive or negative association. An important element of the *i** approach is that it enables developing complex requirement models, that present dependencies between requirements and other elements. These models have influenced the requirement acquisition phase as well as the calculation of the negative effects after realisation. Another novel concept of this approach is that it aims to model dependencies between elements and not particular elements.

The primary element to be represented is the actor. The actor does not only perform actions, but has intentional aspects such as objectives, rationale, and commitments. These intentional aspects can be categorised into 4 process elements; *goals*, *soft goals*, *tasks* and *resources*. Actors depend on each other in order to perform and achieve tasks, as well as receive resources. A goal represents a condition or state of the world that can be achieved or not. Goals are in effect functional requirements, which are either met or not. A functional requirement defines what a system must be able to do or what kind of behaviour it should have. A task represents a certain approach of achieving a goal. Therefore a task can be considered as a detailed activity of how to accomplish a goal and produce changes in the real world. Resources are used to model objects in the real world. These can either be physical or informational and can be used by the actors. The concept of the goal is directly related to the concept of goal achievement. There are secondary goals, though, which are complex to define, such as goals that describe properties or constrains of the modelled system. Conceptually we define soft goals as the non-functional requirements of our system.

3.4.1 Aura's Context Model

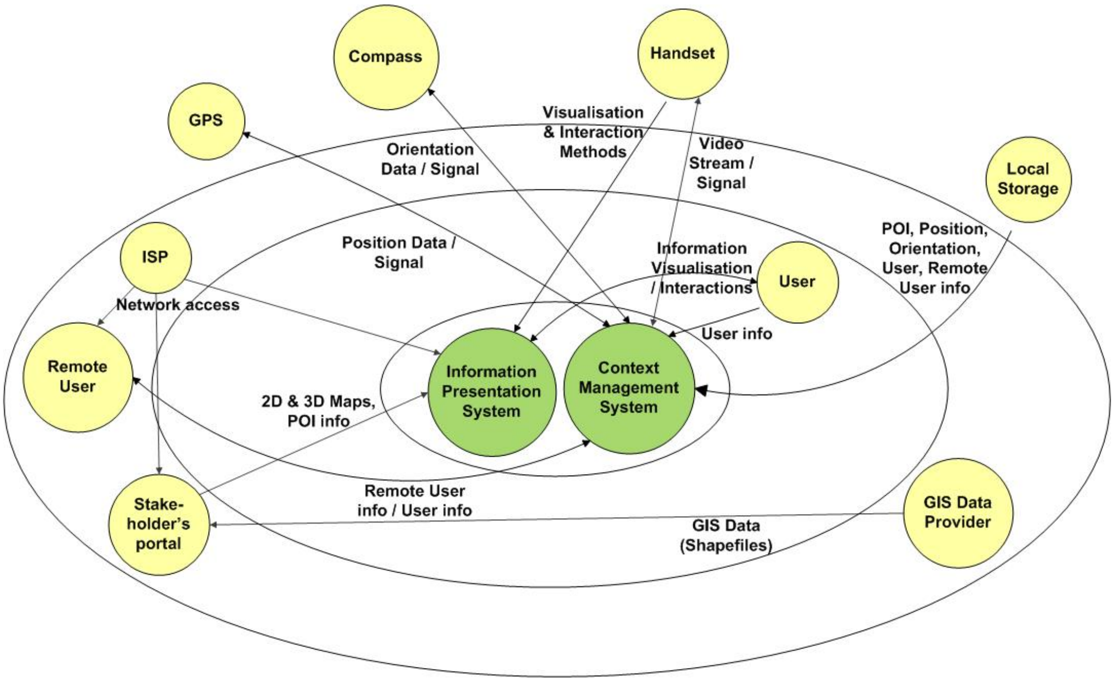


Figure 3-12: Context model for Aura

Aura is the socio-technical system that has been designed and developed. It is drawn in a circle at the centre of the diagram. The other actors who interact with *Aura* are defined around it. These include other human actors who collaborate with *Aura,* such as the users and other software systems, such as the stakeholders' portal. The arrowheads indicate the direction of the data flow between the actors. For instance, the GIS data provider supplies data to the stakeholders' portal but receives nothing in return. On the contrary, there is a two-way information flow between the user and the Information

Presentation System (IPS). The previous diagram defines *Aura's* system boundaries. However, there are more factors to consider about system boundaries.

Indicating System Boundaries

We had to extend *Aura's* context diagram by defining different boundaries. The reason was that during the RE phase we found that some ideas have been within the design remit while others have not. For instance, the design of a GPS parser, which accepts National Marine Electronic Association sentences (NMEA 0183), is within the design remit because it provides vital location information, whereas satellite constellation servicing is clearly beyond the design remit, which means that it must be treated as a domain assumption during the RE phase. However, some aspects cannot be directly redesigned, but we can seek to influence their behaviour by using our design. An obvious example in the domain of *Aura* is the user. Whilst a stakeholder cannot make a client behave in a certain way, they seek to influence his or her behaviour by using the provided services, such as offering custom 3D models with high level-of-detail, improved usability and lower cost. In the design of *Aura*, we explore this grey area by defining four system boundaries (Maiden et al., 2003).

1. The technological systems, expressed in terms of software and hardware actors, which have received the main software development focus;
2. The redesigned work system, expressed primarily in terms of human actors. Their actions are changed due to the introduction of *Aura*;
3. Other hardware, software and people systems that are directly influenced by the redesign of the new system. These systems will need to change to accommodate the new system and its users, but are not dependent on it;
4. In addition, the context diagram also specifies the systems that interact with *Aura* but are not influenced by its redesign. These receive no consequences due to the introduction of *Aura*.

Figure 3-12 illustrates an elaborate version of the context diagram that describes *Aura*, which includes the additional system boundaries. In the centre circle, *Aura's* main entities have been laid on green colour. The primary user is represented at the next level. In this project, the actions performed by the user have been redesigned, in order to conform to the use of *Aura*. At the third level, there are several other software systems, which although beyond the design remit of our research, we sought to influence their behaviour. Namely, these are the GIS data provider, which offers spatial information about the environment, the Internet Service Provider (ISP), which offers networking services to the requesting entities and the stakeholder's portal, which makes personalised services to the users of the system available. Additionally, another user may participate. That is the remote user, who communicates with the primary actor via the core system over specified collaboration patterns. At the fourth level come the systems, which are beyond the design remit and influence of the research. These are mainly the hardware resources such as the mobile device, the storage system and the available sensors (i.e. GPS and compass).

3.4.2 Strategic Dependency Model

In this section the *i** Strategic Dependency (SD) model (Yu and Mylopoulos, 1994) for *Aura* is presented. The SD model describes the actors that participate in *Aura's* operation. The model focuses on resource and goal-type dependency links and does not include task and soft goal dependencies. Dependencies between local actors and systems have been modelled and treated as transitive, in order to avoid modelling duplicate dependencies. *Aura's* context diagram can be considered as a simple version of the more elaborate SD model.

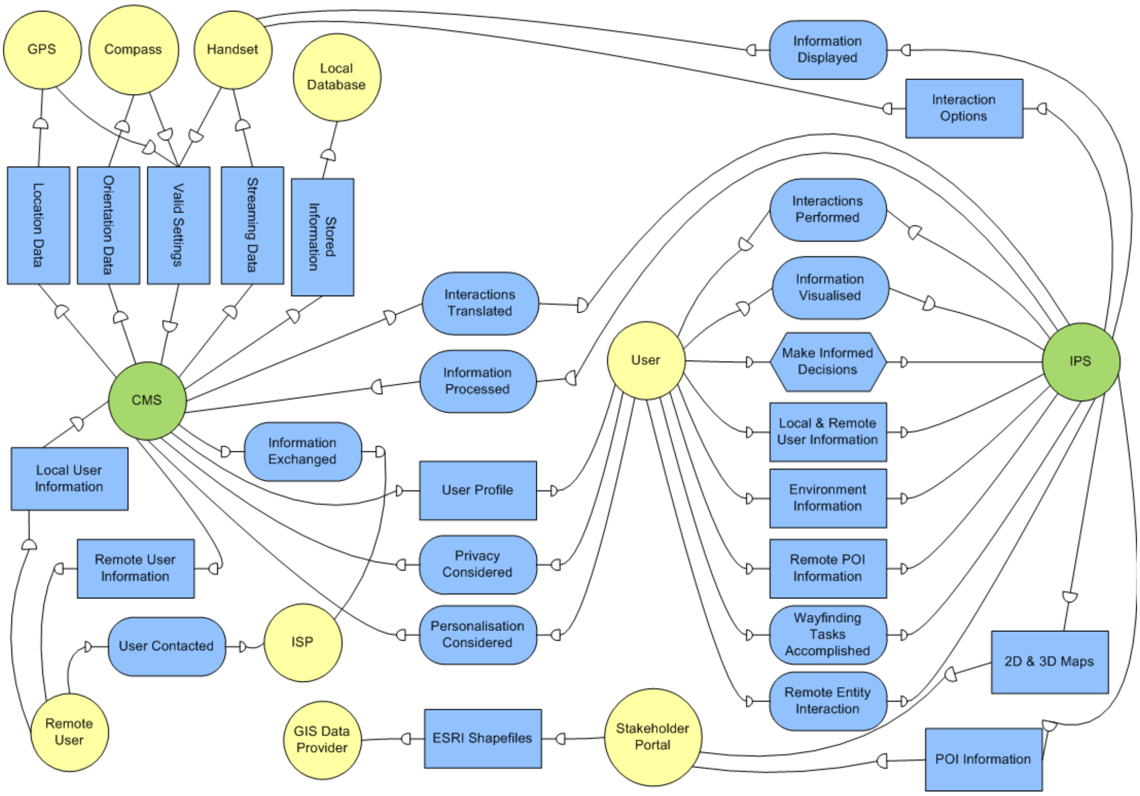


Figure 3-13: Aura's SD model

The main objective of this diagram is to represent dependencies between the local user and *Aura's* core, which is formed out of two main entities; the Context Management System (CMS) and the Information Presentation System (IPS). In the previous section, we defined the four boundaries of the complete socio-technical system. This section focuses only on the first two layers, whose design and implementation we can directly influence, discarding the rest of the participating actors and sub-systems. In more detail, we have examined the dependencies between the CMS, the IPS and the local user.

The users expect to get assistance for explicit tasks in the real world by using the proposed framework. In order to accomplish these tasks, which are either vague or more specific, there are several dependencies between the actor and the main software components. The nature of these dependencies relates directly to the functionality of each sub-system. *Aura* was designed in a way that could decouple the processing from the presentation of information. The main contributor for achieving such functionality was the separation of interactions and visualisations into two distinct streams. In total, there are three groups of conceptual requirements, which the framework attempts to satisfy. They are described in the following list.

1. Acquisition and management of contextual information, which is handled by the CMS entity;
2. Visualisation of information, which is handled by the IPS entity;
3. Interaction with information, which is accomplished either automatically by the CMS or manually by the user, through the IPS.

CMS and IPS

The main dependencies between the 2 core software components, relate to the internal exchange of information. Hence, the IPS depends on the CMS to process information and receive a controllable format. In contrast, the CMS depends on the IPS to present the information that has been accumulated in a way that is constructive for the user.

User and CMS

Furthermore, the CMS depends on the user to create a valid profile and pass any relevant preferences to the system. While this sensitive information is being processed, the user depends on the CMS to respect the privacy considerations that have been applied and not transmit such data to another party, which has not been authorised. The user may decide to release this kind of information, in exchange for certain personalised services. Thus, the CMS has to proactively accumulate new information and notify the user of any potential features of immediate interest. It must be noted that the physical activity of the user generates the context that is acquired indirectly by the CMS.

User and IPS

The user depends on the feedback received from the IPS to make informed decisions about any task at hand. This is accomplished by accepting up-to-date, reliable and accurate information about himself (e.g. position and orientation), about other users (e.g. preferences), about other objects of interest (e.g. proximity) and about the environment. Hence, the user depends on the IPS to represent all relevant information in a constructive and rewarding way, which will not obstruct but enable more effective decision-making. Eventually, the user relies on the IPS to interact with any remote entities, in order to retrieve information about them, or even collaborate in case these are other human actors. Although the framework supports environmental representations, this is not adequate to sustain spatial interactions in the virtual or real world. Therefore, the user relies on the IPS to successfully accomplish any wayfinding tasks. These include primed and naïve searches and even exploration of the immediate environment. In contrast, the IPS depends on the user to interact with the user interface, in order to offer data representations and efficient assistance.

3.4.3 Strategic Rationale Models

Strategic Rationale (SR) models provide a description of processes in terms of process elements and the relationships or rationales linking them. A process element is included in the SR model only if it is considered important enough to affect the achievement of a goal (Maiden et al., 2007). Actors may be able to accomplish a task either by themselves or by depending on other actors. Similarly to the SD model, the SR model has 4 main types of nodes, which are goals, soft-goals, tasks and resources. A link between the 4 nodes can be established by using any of the four available relationships. These are the *Dependency* link, the *Task Decomposition* link, the *Means End* link and the *Contribute to Soft-goal* link (Yu and Mylopoulos, 1994).

Context Management System (CMS)

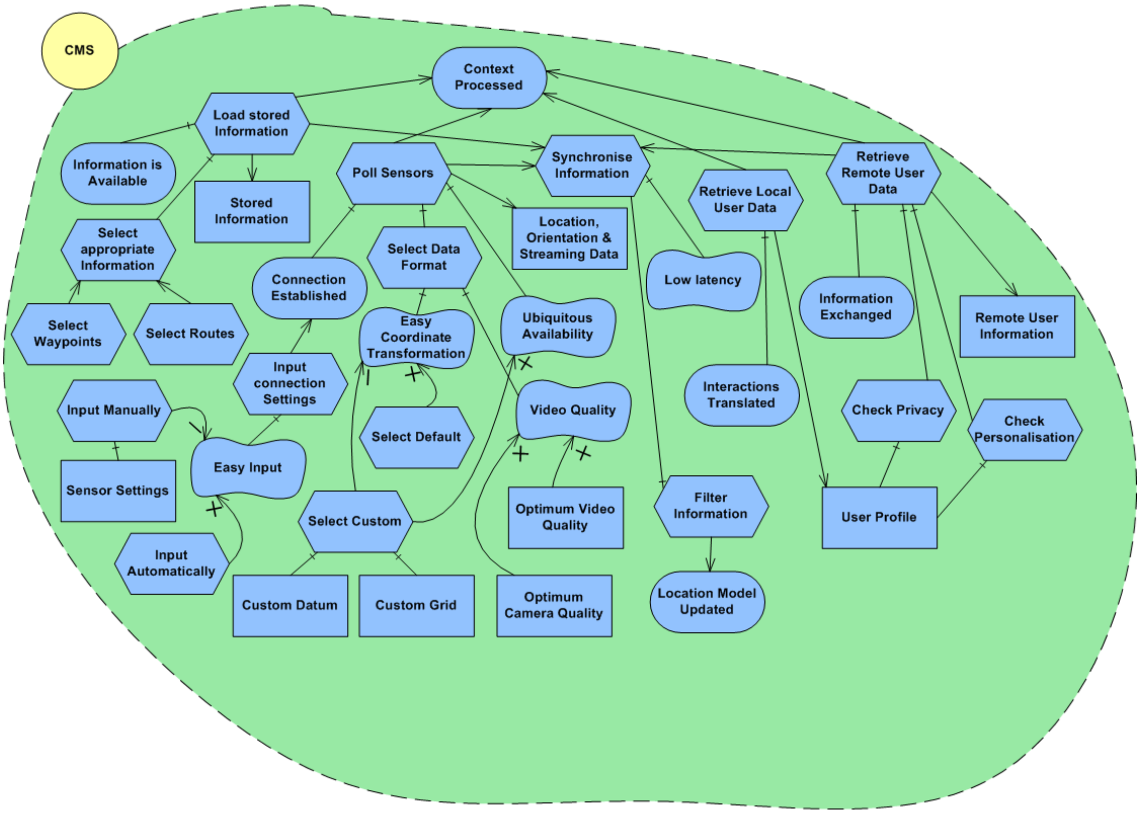


Figure 3-14: SR model for Context Management System

Figure 3-14 illustrates the SR model, which describes the core processes that are involved in the successful operation of the context management sub-system of *Aura*. The reader can observe the high-level tasks that can be performed in this subsystem. The most important of all is the polling of the attached sensors. Every sensor poses its own requirements regarding the type of feedback and connection settings. Hence, an automatic mechanism should enable the operation of these sensors without making the user manipulate every parameter manually. In contrast, manual control of these parameters may offer advanced system functionalities, such as ubiquitous operation through compatibility with international standards. Additionally, this subsystem must handle the retrieval of stored context. The data types that can be saved in persistent storage contain information about objects of interest in the environment or specific routes on how to reach distant locations. Another core functionality is the negotiation between the local and a remote user. Communication can take place only if the privacy criteria of both users are matched and their personalisation preferences are verified. Finally, the geometric location model, which is the foundation of all spatial interactions, is updated with information that has been acquired and has passed the synchronisation and filtering sub-processes.

Information Presentation System (IPS)

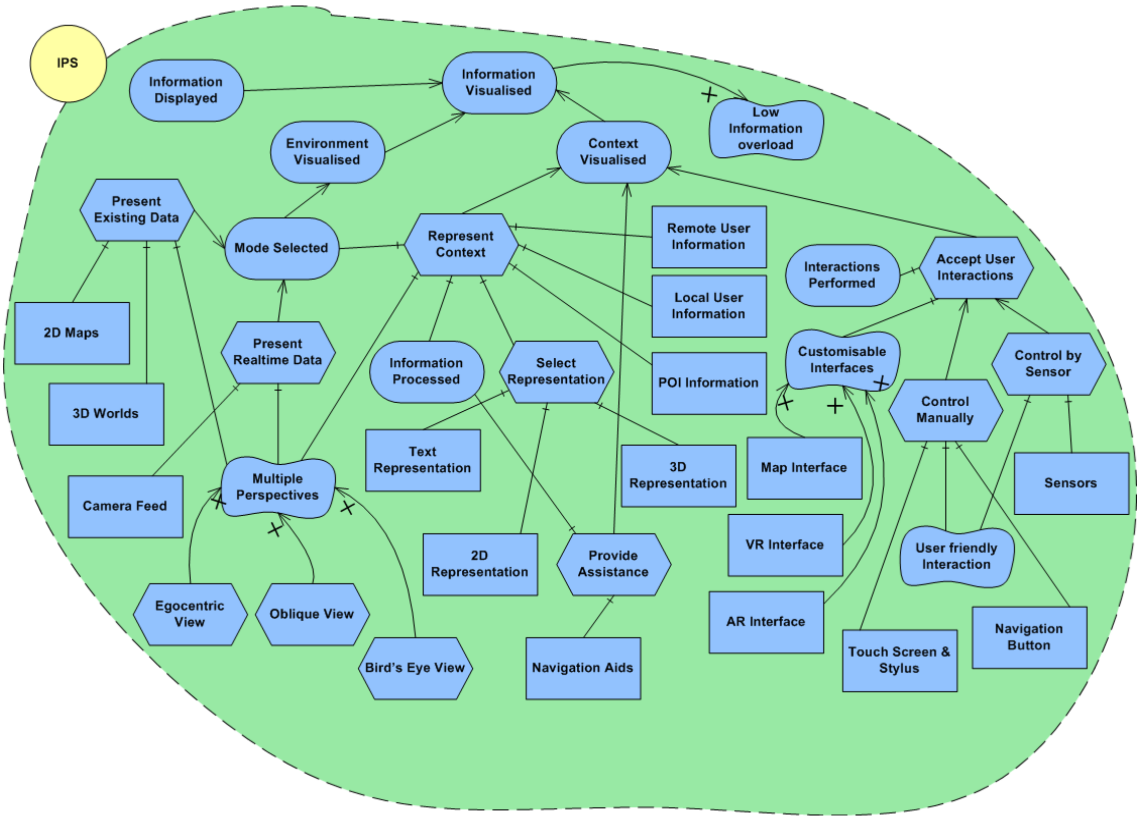


Figure 3-15: SR model for Information Presentation System

The primary goal of the Information Presentation subsystem is to display relevant information to the user by accumulating all sensed, derived and explicitly provided context. The presentation of this information must take place in a geo-referenced environment, which can be assembled from 2D maps, 3D worlds and camera feeds. Thus, environmental and context representations need to be individually handled. That is because the representation of an object can support many forms according to the interface that represents it. In effect, the representation of a specific POI can be different in the map that supports only 2D overlays compared to a VR scene, which supports full 3D detail. The VR engine should support a wide range of observation perspectives, because this way correlation between the virtual and real world is more effectively achieved. Further navigation assistance is offered to the user by visualising real-time navigation aids such as signs placed at decision points or lines that overlap with the path to be followed. Assistance becomes feasible after processing the information retrieved from the CMS unit. The utilisation of three distinct user interfaces, which can support spatial interactions, allows the user to interact with the virtual world and any elements that exist in it, as well as with other remote users. The availability of every interaction and visualisation option depends strongly on the underlying hardware, but if the system aims for optimum platform compatibility, then several techniques need to be implemented. For operation on a mobile device, the standardised interaction mechanism is the navigation button, which supports at least four directions. In addition, nowadays, several device manufacturers embed a touch screen, which enables more freedom and pixel-level accuracy. This allows the user to control all interactions with the device either manually through the device-specified mechanisms or automatically by examining real-time context changes.

User

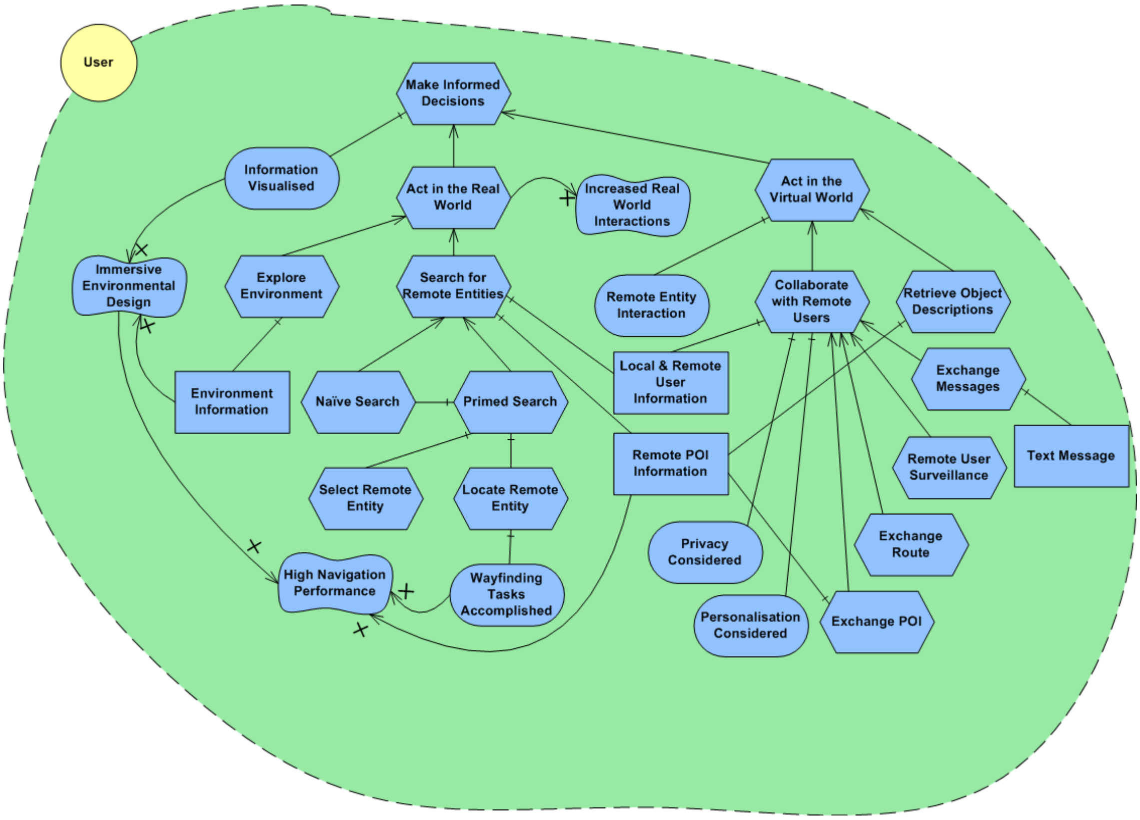


Figure 3-16: SR model for User

In this diagram, the reader can observe the high-level goals, which must be satisfied by using *Aura*. The main objective is to assist the decision-making process of the user in order to act accordingly. To influence this process, the user needs to receive and visualise information applicable to the task that he or she is currently involved in. This way information is converted to knowledge, which can be applied to effectively adjust the current behaviour. In order to change the behaviour, the user must act accordingly. *Aura* provides assistance to the user by supporting real and virtual world activities. In the real world the actor has the ability to explore an environment and gather information about it, so that he can gain a better understanding of the surroundings by obtaining valid landmark knowledge. Furthermore, procedural knowledge may be enhanced while searching for an object, because specific assistance may be triggered on how to locate the remote entity. Finally, when the user follows the framework’s navigation aids, their survey knowledge is enhanced because they obtain an evident tactic to reach the destination by following the interconnected topological elements. Activity in the virtual world supports interaction between the main actor and any modelled object of interest. In this case, the user can make informative decisions, which may influence physical activity. This happens because further descriptions about an entity can be acquired, as the user is not limited by the visual appearance of an object. For instance, the services provided by a shop can be obtained before actually visiting the shop - just by examining the data on file or by connecting remotely to an online information source. Collaboration between two actors can also take place. In technical terms, collaboration may be achieved through various forms. Namely, these include message exchange, POI exchange, pre-followed route exchange and real-time remote user surveillance. When all visualisation and interaction functions are combined, the user immerses in the selected environment in order to accomplishing several tasks, while being mobile in an unfamiliar location.

4 Prototype Development Methodology

In this chapter, the reader can discover specific issues that affected the analysis and design of the developed system. The framework architecture, which includes the hardware and software components that were put in use, is described here. Examining the available technological products, standards and protocols has produced the hardware specifications required for the operation of the proposed system. The selection of the software components and platforms that sustained the development efforts is also presented in this chapter. Furthermore, the desired system functionality, which is expressed in functional and non-functional requirements, is laid out. The chapter concludes with the delivery of the system design that illustrates certain core aspects inherited and enabled by the realisation of the system architecture.

The chapter demonstrates how a specific mobile mixed reality system, *Aura*, was designed as part of this research project. Technical details include hardware requirements in terms of devices and external sensors, client-to-client network architecture and the use of current standards to promote interoperability over multiple platforms. The architecture of *Aura* integrates a variety of software components, in order to study novel concepts in the field of Information Science. This work aims to inform the design phase of most theme-based applications, ultimately seeking to implement these concepts. According to the Requirements Acquisition Survey and the Preliminary Evaluation results and other research projects which were found relevant, we seem to be in a position to make an informed choice on the selection of the suitable platforms and available technologies. The Requirements Acquisition Survey produced a set of user requirements which can influence the results of the 1st Research Question that was presented in Chapter 1.1 Furthermore, the Preliminary Evaluation provided valuable outcomes about the 2nd Research Question. Thus, a technical specification can evolve which can be considered as the answer to the 3rd Research Question. This Chapter summarises the specification, in terms of requirements and required by the end-users functionality, and presents the answers to the 1st and 2nd Research Questions, in terms of a design solution. At length, our goal is to introduce a mobile context-aware MR application, which could be adopted for use by commercial audiences and support applications in various domains.

4.1 System Architecture

Due to the increased adoption of high-tech mobile devices by end-users, hardware manufacturers and software developers have committed to the growth of the mobile industry. The result of this progress has produced the technical bedrock that can effectively support the development of robust mobile applications. Most developments in the *mobile* field have been affected by developments generated for the *static* field, such as the amplified CPU frequencies and data storing technologies. Likewise, several data communication protocols have evolved, which can explicitly serve diverse communication requirements, without the need of being physically connected to the transport medium. Namely, in descending order according to their effective range, these protocols include Universal Mobile Telecommunications System (UMTS), General Packet Radio Service (GPRS), Wireless Fidelity (Wi-Fi) and Bluetooth. Furthermore, commercial display panel manufacturers have produced a wide range of products, which can support very high pixel resolutions in confined sizes, as well as offer True colour representations. The Liquid Crystal Display (LCD), Thin Film Transistor (TFT) and Light-Emitting Diode (LED) technologies are those that are mostly embedded in the latest mobile devices. The continuous decrease in Printed Circuit Board (PCB) sizes, and specifically the evolution of MEMS technology, has enabled the development of computing systems and sensors, which are more practical and portable than ever before.

The value of certain technological achievements has been indispensable for the evolution of the functionality that mobile platforms offer. For instance, the introduction of low-cost Position Determination Technologies (PDT), like GPS sensors, has rendered the acquisition of location context quite trivial and, nowadays, it is implicitly or explicitly used by an increasing number of mobile applications and users. The required functionality of this technology is accomplished by measuring the duration that a satellite-generated signal needs in order to arrive to a terrestrial receiver, as long as the position of the satellite is known. This technology has been mostly adopted on mobile devices because their position is constantly altered, rather than on static computing devices, whose location is infrequently changed. An extensive list of positioning determination technologies has been provided by Raper et al. (Raper et al., 2007).

The increasing use of position determination technologies and their association with mobile clients has provided the ground for certain applications and services to evolve, with the most notable example being LBS. The reason why certain PDT has positively affected the development of LBS is that the main contextual value that is processed is the position of the user who operates the system. The information visualisation component of most LBS depicts the representation of the environment from an allocentric plan view, where the spatial representation adopts a frame of reference that is external to the observer (Klatzky, 1998). This way, the user observes the surrounding area and any objects of interest from a bird's eye view, which can be considered as the digital substitute of 2D maps. Certain features of allocentric representations have proven valuable for several GIS-related applications that are executed on stationary workbenches, but do not effectively support the use of mobile platforms. Mountain and Macfarlane named two reasons for which the allocentric approach is not adequate for mobile LBS. In most cases, dedicated workbenches have larger monitors attached, which can represent more spatial information for a particular area in high quality. Following next, for desktop users, the area of interest that is being browsed on a map is less likely to coincide with their physical location when compared to mobile users, whose information tends to be more dependent upon their surroundings (Mountain and Macfarlane, 2007). After examining the results of the *Requirements Acquisition Survey* analysis, we also observed that static devices are more frequently used to locate a point or area of interest rather than navigate towards it. Thus, an allocentric perspective is found more efficient to locate a remote entity, whereas an egocentric perspective is more helpful for the users when trying to find their way and interact with the immediate surroundings. By taking into consideration these reasons, we can conclude that the allocentric view may not convey well to mobile applications, where the implementation of an egocentric perspective that complements the viewpoint of the device beholder, may be more suitable. A distinction between the available visualisation perspectives that were found important for the development of our system is provided in Chapter 3.3.3. Broadly, the three visualisation perspectives which can offer distinct advantages in the context of navigation are (i) the *egocentric straight angle perspective*, (ii) the *allocentric oblique angle* perspective and (iii) the *allocentric plan view* perspective. In the evaluation part we explore the circumstances under which each perspective is suitable for fulfilling the user's information needs.

In this project, we have explored novel approaches to represent spatially referenced information for use in context-aware systems and consequently in other applications that offer a direct effect on user activities, such as pervasive entertainment applications.

Certain user interfaces (i.e. MR) can present blended information from the physical environment, as observed by a mobile device beholder, complemented with contextual information. This approach is dependent upon the synthesis of multiple sensors readings in real-time, which can track a user's natural behaviour within an external 3D Cartesian coordinate system and correlate peripheral information relative to the real-world landscape. This approach can display information in situ, eliminating the need for mobile users to associate between the actual real-world viewpoint and a remote one, such as the one supported by traditional LBS (Mountain and Liarokapis, 2007) (Liarokapis et al., 2006a).

The proposed system architecture was implemented on a mobile platform. In subsequent paragraphs, the reader can find the rationale that guided the development of this system and certain issues surrounding it, like why native C++ development was preferred instead of Java. During the evaluation phases, we tested compatibility with several recent devices from various commercial manufacturers. Limitations and variations of devices that affect the interaction and visualisation interfaces are described in terms of hardware interaction methods (i.e. touch screen, buttons, screen size), processing performance (i.e. CPU, RAM, ROM and GPU), sensor connectivity (i.e. Bluetooth or embedded) and network connectivity options (e.g. GSM, GPRS, UMTS, Bluetooth and 802.11).

4.1.1 Hardware-based Components

Mark Weiser believed that the most important technologies are those that work transparently from the user and provide contextualised services through devices and sensors that are distributed in the physical environment (Weiser, 1993). These interconnected devices constitute the hardware layer, which enables ubiquitous system operation. The hardware requirements are described in terms of mobile device specifications, the underlying network infrastructure and additional sensors required for context acquisition and management.

4.1.1.1 Mobile Devices

Recent commercial trends have demonstrated that the most usable and publically accepted solutions tend to be mobile communicating devices such as handheld PCs, PDAs and mobile phones. Some of the specifications of the latest devices include high frequency processors, exceeding 1.0GHz, a lot of available RAM (i.e. more than 256MB) and excessive volatile memory dedicated for storing user content, as well as expansions slots for various memory card types that can extend the default storage capacity. Some other crucial features found on recent mobile devices are the utilisation of high resolution displays supporting 1280x1024 pixels, or even more, and dedicated graphics acceleration mechanisms similar to those found on larger computers. The aforementioned specifications, and specifically the last set, enable the provision of advanced user-friendly interfaces that can offer elaborate interaction options for the users. A review of the hardware specifications of several modern smartphones, which can satisfy most operational needs of the developed framework, can be found in Appendix IV of this report. This type of mobile devices can reflect current technological progress and produce a good performance to size ratio, which is an essential element for modern Commercial Off-The-Shelf (COTS) tools. Despite the advantages of such equipment, there are several limitations, which can affect the user experience (Papakonstantinou and Brujic-Okretic, 2009a). The relatively small display size, for example, needs to be efficiently managed in order to accommodate any interaction requirements, as well as to visualise relevant information. The framework has been deployed and executed over a range of devices in order to verify compatibility. The following list presents the hardware solutions, which have been tested and successfully work with *Aura*.

- Dell Axim x51v
- HP iPAQ hw6915 (HTC manufactured and re-branded)
- HTC Touch Diamond P3700
- i-Mate JASJAR (HTC manufactured and re-branded)
- MIO A501 Digi-Walker
- Sony Xperia X1 (HTC manufactured and re-branded)

The functionality of our technological framework can be divided into two main categories. The first category facilitates the acquisition and management of contextual data, whereas the second category introduces visualisation of information to produce better decision-making effects for the user. In certain scenarios, though, which are described in following paragraphs, there is no need for every device to conform to the same specifications. The requirements for certain use cases may call for some devices to offer functionalities that belong only to a single category. In more detail, when operating in a surveillance scenario, one device may offer context retrieval and transmission of data, while a second device may offer visualisations of that information to a remote user. Such functionality reduces the hardware requirements that each device is expected to meet, with the aim of fitting in a single category. Thus, some equipment may offer certain services to the framework, while leaving the remaining services for other devices to process accordingly. In every case, though, if a device can satisfy all of the requirements, which are presented in the following paragraphs, it is found suitable for executing *Aura*. Devices that offer only context acquisition and transmission are referred to *as thin clients*, whereas devices that offer full functionality are referred to as *thick clients* for the purpose of this report.

Thin Client Devices

In the previous paragraph, we observed that two types of client equipment are supported. The thin client can be described as an input device that does not provide visualisation options. It can be used for collecting data from sensors and distributing it through the established network infrastructure for post-processing. Additionally, it should be capable of creating track logs in the form of XML documents and storing them for a certain span of time. The purpose of such minimal functionality is that only basic hardware and software resources need to be utilised. This type of device can work with our framework even if the basic platform does not conform to the proposed architecture. The only constraint for such operation is the compatibility with the networking protocol that has been developed and described in Chapter 5.6. During the course of the project, certain developments took place towards that direction but eventually they were abandoned because they were considered surplus to the requirements of the research. The primary reason was that the full version of *Aura* could sustain all required functionality.

Development on the thin device took place mainly on a Nokia 6230 mobile phone. The product was a lightweight application, which has been reported to work on several devices supporting the Series 40, 2nd edition, Symbian platform. The application was developed in J2ME and the thin device that ran it supports MIDP 2.0, CLDC 1.1 and JSR-82, JSR-185, JSR-135 and JSR-120. These J2ME technologies were the minimal standards for devices in the market during 2006. The device specifications do not include any 3D rendering and, even if they did, the processing and visualisation subsystems would not be capable of handling textured representations of the real world. Bluetooth (JSR-82) was found to be the optimal interface for connecting the mobile device to the sensor instruments.

Thick Client Devices

The thick device types are the hardware platform that sustained most developments. This device type receives most interaction by the users, immerses them into a VE and collects context values, in order to represent - as accurately as possible - the currently experienced user situation. During most development cycles of the rapid-prototyping approach that has been pursued, the researcher tried to introduce one of the latest mobile devices as the new platform, which further features of *Aura* would be implemented on. At this point we present, in chronological order, which devices have been utilised to produce certain features. The specifications of each device represent the technical progress, which has been observed during the project duration.

- Initially, a *Vodafone v1640* (i.e. i-Mate JASJAR) was employed to develop the first-cut VR interface and connect it to the sensor-controlled interaction engine. Additionally, further manual interaction mechanisms were designed, to provide the user with complete control of the virtual environment. This device did not include the software drivers, which manage the on-board camera, in the same way as it did not offer adequate performance in terms of graphics rendering.
- The second device that was acquired was a *Hewlett Packard iPAQ hw6915*. On this hardware, the first-cut AR interface was developed because HP provided detailed camera drivers, which did not originate from the original manufacturer. At this point, both the augmented and virtual reality interfaces could be concurrently operating, either under sensor or manual control. This version produced an application that offered a large subset of the virtuality continuum, in terms of visualisation, interaction and immersion features. Performance, though, in both environments, was not very efficient as the reproduction of graphics was being accomplished by software means.
- The last and most hi-tech device, which was put on the development bench, was an *HTC Touch Diamond*. On this device, the networking infrastructure had been produced, which could interconnect two users and allow them to exchange contextual information in real time, as well as simple text messages. Graphic

performance was found adequate for the purpose of the framework functionality because the embedded microprocessor (i.e. Qualcomm MSM7201A) offered hardware acceleration of the graphics pipeline and included dedicated OpenGL ES drivers.

The described portable devices, which were used for the development of the software framework, are based on Windows Mobile version 5.*x* and 6.*x* PocketPC edition with battery capacity close to 1200 mAh. The rest of this subchapter introduces technical requirements that were deemed critical for the smooth operation of *Aura* and all of its subcomponents. The specification of the requirements documentation is based on the Volere template (Robertson and Robertson, 2010).

Portability Requirement 1

Identifier	**PortR1**
Description	The mobile device shall be able to support a compatible Operating System
Type	Portability Requirement
Fit Criterion	The device satisfies the requirement if the embedded OS is Windows Mobile 5.*x* or higher, either Smartphone or PocketPC edition. This requirement is met if the device firmware has been developed by using a subset of the 32-bit Windows CE 5.*x* or later product family.

Portability Requirement 2

Identifier	**PortR2**
Description	The mobile device shall embed a compatible processor
Type	Portability Requirement
Fit Criterion	The device satisfies the requirement if the embedded CPU is based on the RISC processor structure. The CPU should operate on a 32-bit data bus and support the ARM instruction sets. This requirement is met if the processor belongs to the following, or similar product families: Intel XScale, Marvell PXA, Qualcomm MSM, Texas Instruments OMAP and Samsung S3.

Device Requirement 1

Identifier	**DR1**
Description	The mobile device shall be lightweight
Type	Device Requirement
Fit Criterion	The device satisfies the requirement if its weight does not obstruct the user to perform any natural interactions, while holding it with one hand. Natural interactions are defined as lifting, panning, rotating and rolling. This requirement is met if the weight of the device is less than 300g.

Device Requirement 2

Identifier	**DR2**
Description	The mobile device shall be small
Type	Device Requirement
Fit Criterion	The device satisfies the requirement if its size does not obstruct the user to perform any natural interactions, while holding it with one hand. Natural interactions are defined as lifting, panning, rotating and rolling. This requirement is met if the dimensions of the device are less than 15cm in length, 10cm in width and 5cm in depth.

Device Requirement 3

Identifier	**DR3**
Description	The mobile device shall include a relatively large display
Type	Device Requirement
Fit Criterion	The device satisfies the requirement if the screen size allows the user to visualise the presented information, without the need to change the observation distance while holding the device. This requirement is met if the diagonal size of the display is more than 6cm.

Device Requirement 4

Identifier	**DR4**
Description	The mobile device display shall support high pixel resolutions
Type	Device Requirement
Fit Criterion	The device satisfies the requirement if the screen resolution allows viewing a lot of information without the need for the user to perform additional actions. Furthermore, the display resolution should be equal or more than the resolution provided by the camera feedback, in order to present the whole video stream between the screen boundaries. This requirement is met if the display resolution is more than 320x240 pixels.

Device Requirement 5

Identifier	**DR5**
Description	The mobile device display shall support a wide range of colours
Type	Device Requirement
Fit Criterion	The device satisfies the requirement if the display supports high colour depth configurations, in order to represent real-world information in an acceptable and easy manner for the user to associate with. HighColor is defined as 65,536 (65K) and TrueColor as 16,777,216 (16M) available colours per pixel. This requirement is met if the display supports either the 16-bit R5G6B5 or the 24-bit R8G8B8 pixel format.

Device Requirement 6

Identifier	**DR6**
Description	The mobile device shall offer adequate power
Type	Device Requirement
Fit Criterion	The device satisfies the requirement if the available power capacity does not obstruct the operation of the platform for at least half an hour of continuous operation, while having all features enabled. There are two main issues for this requirement to stand. The available current capacity of the battery should be relatively high, offering more than 900mAh. The processor and the operating system should include functions that allow energy-efficient operation of the device.

Device Requirement 7

Identifier	**DR7**
Description	The mobile device shall offer adequate storage capacity
Type	Device Requirement
Fit Criterion	The device satisfies the requirement if there is enough space in the ROM to store volatile and non-volatile data that is needed for smooth operation of the platform. The following lists present the essential data and approximating measurements of the required space. Compulsory elements • Platform installation (approximately 5MB); • Configuration files (≤2MB). Optional elements • ESRI Shapefiles for 2D environment representations (variable); • VRML Models for 3D environment representations (variable); • Images for texturing the VRML models (variable); • Local User information (≤100KB); • Remote User information (≤100KB per user); • Point of Interest information (variable); • Route information (variable). This requirement is met if the available memory on the device conforms to current manufacturer intentions, to produce devices with more than 64MB of available to the user memory.

Device Requirement 8

Identifier	**DR8**
Description	The mobile device shall offer alternative input mechanisms
Type	Device Requirement
Fit Criterion	The device satisfies the requirement if there is at least one method for the user to accurately input information. Interaction with the platform needs to be precise and meet all inherited user needs presented in this chapter. This requirement is met if the device embeds either a navigation button or a touch screen. The navigation button should

	support at least 4 directions, in 90° margins. The touch screen should conform to all display requirements and especially to DR4, which reflects the accuracy of this input mechanism.

Performance Requirement 1

Identifier	**PR1**
Description	The mobile device shall have a fast processor
Type	Performance Requirement
Fit Criterion	The device satisfies the requirement if it is capable of concurrently* executing at least 8 resource intensive threads, without the user observing any latency issues during their interactions. This requirement is met if the processing frequency is over the specified threshold of 400 MHz. *Note: Currently, only a few mobile multiprocessor devices are available to commercial audiences.

Performance Requirement 2

Identifier	**PR2**
Description	The mobile device shall have a lot of available memory
Type	Performance Requirement
Fit Criterion	The device satisfies the requirement if it can load large textured 3D models, without crashing or slowing down the application. The VR interface is the most resource voracious component of the platform and is considered as a reference point for the operation of the system. This requirement is met if the available RAM is over the specified threshold of 50MB.

Performance Requirement 3

Identifier	**PR3**
Description	The mobile device shall offer graphics acceleration
Type	Performance Requirement
Fit Criterion	The device satisfies the requirement if visualisation of all interfaces and their components are presented to the user without evident delay. All interactions that take place in the 2D, VR and AR interfaces should be reflected in real time and take no longer than 1 second to refresh the display with the latest available information. This requirement is met if the device has a hardware accelerator in the form of a standalone solution (e.g. NVIDIA GeForce 5500 or Intel 2700G chipsets), or it embeds graphic acceleration on the main CPU, with enhanced instruction sets. Ideally, the drivers for this component should be available from the manufacturer and must be compatible with the commercially available development tools. The nominal frame rate should be not less than 5 FPS in any interface.

4.1.1.2 Sensors

Apart from the specifications of the mobile device platform, additional sensor equipment is necessary for the implementation of mobile context-sensitive systems. These sensors operate in conjunction with client devices, in order to fuse context into the framework engine. Some sensors are associated with the capture of the physical surroundings and some with the processing of the user-generated behaviour, relative to the natural environment. For the implementation of *Aura*, these sensors namely include a GPS receiver, a digital compass and a video camera. The synchronous processing of information generated by several sensors in real time is a complex procedure, which has not been extensively researched in the context of mobile MR applications.

Accurate representation of the user's situation in any interface including augmented and virtual reality requires the exact position and orientation of the device's viewpoint (i.e. 6-DOF) to achieve registration between the physical scene and virtual information. Commonly used image analysis and pattern-matching techniques can determine the position and orientation of a user in relation to a fiducial marker or natural feature. A novel advantage of the approach that we decided to follow during the design of *Aura* is that it should work in unfamiliar environments, within which there will not be any features, which the system has been trained to recognise.

For the implementation of *Aura*, a preferred alternative to capture the user's viewpoint is by processing the readings generated by dedicated sensors. The latest position information can be acquired by polling a GPS sensor. The result of this process produces either a 2D or a 3D set of position coordinates when three or more than three satellites are in the sensor's line of sight, respectively. Most recent mobile devices are shipped with GPS receivers embedded by the manufacturer. Furthermore, the devices, which do not support such functionality, can be straightforwardly transformed to location-aware through a connection to an external sensor. A range of GPS sensors has been tested and found to work sufficiently accurately with *Aura*. The following list presents the standalone sensors, which have been examined, according to the provided accuracy in everyday scenarios.

- Pharos iGPS 360 (SiRF starIIe/LP microcontroller)
- EMTAC CRUX II BTGPS (SiRF starII microcontroller)
- Holux GPS Slim 236 (SiRF starIII microcontroller)

The latest mobile devices, which have been used as a development platform, embedded their own GPS sensor. This was found particularly useful because they released vital network resources for use with additional sensors. Further discussion about this topic is provided in the following subchapter. The following list presents the mobile devices that have an embedded GPS receiver and its type.

- HP iPAQ hw6915 with SiRF starIII
- HTC Touch Diamond P3700 with Qualcomm gpsOne

The GPS receiver of both devices provided results residing between the expected accuracy boundaries and, surprisingly, offered partial indoor positioning, which was not in the manufacturer specification. The accuracy of indoor positioning was not great, but evidently the measurements retrieved from the *gpsOne* sensor were far more stable. Additional software offered free of charge by the device manufacturer, in the form of a *QuickGPS* mobile application, allowed us to update the expected position of satellites in orbit. This resulted in a faster position fix, which has been reported to be one of the main reasons that prohibited a wide use of early location-based systems.

Furthermore, the framework should support Differential GPS (D-GPS) and EGNOS signalling in cases where improved accuracy is required. The only drawback in this scenario is that the receiver needs to be compatible with these variations. Both models described earlier employed Assisted GPS (A-GPS) technology, which was the underlying technology that enabled indoor positioning functionality. A-GPS exploits the Cell ID of GSM-like networks to triangulate and locate the sensor. In some cases, when a few satellites are visible, A-GPS functionality proved unnecessary because the device discarded GPS input and processed only Cell ID readings. Consequently, Cell ID information should complement the GPS functionality and should not be the primary PDT of *Aura*.

Research in MEMS technology has evolved to a great extent during the past few years. Many types of sensors have been produced, which initially were only available for military or industrial exploitation. Currently, several vendors, whose products address independent consumers have manufactured devices that reach even the entertainment and telecommunication industries. Accelerometers and other kinds of inertial sensors have been attached to gaming consoles, like *MotionPlus* of *Nintendo Wii*, and even in high-end mobile phones. The common ground that these consumer devices share is that they offer a new level of interaction experience between the user, the device and the surrounding environment. To satisfy the requirements of *Aura* we have introduced a digital compass, which can provide real-time orientation information. Another way for retrieving such context would be to derive the orientation out of already sensed information. This functionality can be achieved by calculating former and present GPS location coordinates, for a specific span of time. In the case of a user who is standing still, though, this information cannot be reliably measured. That is why a dedicated sensor, which describes 3 Degrees Of Freedom (3-DOF), was found appropriate for use in our system. The digital compass that has been employed belongs to the HMR3300 series of magnetic sensor products (Honeywell, 2003). The compass outputs three-value sentences, which correspond to heading, pitch and roll information. The manufacturer reports that the achieved heading accuracy is in the region of 1.0°, with 0.1° resolutions. Real-time heading is an important measurement because it aligns the virtual viewpoint of the observer with the physical pose. Pitch and roll information was deemed not to be equally important for navigation scenarios, but may prove invaluable for accomplishing gesture interactions in certain other scenarios. The only mobile device that has been tested and embedded an accelerometer was the HTC Touch Diamond. This sensor presented only pitch and roll raw data but not in an easily exploitable form (i.e. OS registry updates). Additionally, accuracy was found efficient only for gesture interactions, which led to the conclusion that orientation should be triggered only by the dedicated compass, which has been selected for that purpose.

There are two distinct technologies, which can be employed to visually blend real with artificial information - optical see-through and video see-through systems. Optical see-through systems require the use of a costly HMD, which renders this method prohibitive for everyday use by most consumers. Alternatively, the mobile device panel can become the component, which accommodates the visual output of the process, in video see-through solutions. In such a case, the mobile device's digital camera becomes the visual tracking mechanism of the system. Therefore, the camera is considered an indispensable source of real-time context. For triggering AR functionality in the developed system, it was obligatory to gain access to the video feed that is generated by the embedded camera of the device. For efficient operation, the mobile devices that enclose a camera also contain, in the registry of the OS and in permanent storage, a series of preinstalled software drivers. Most of these drivers are customised for the specific image retrieval chipset that has been built-in. They can provide advanced functionality, such as high-resolution video frames, auto-zooming, anti-blurring, flash and auto-focus. Although it sounds very promising to work on top of these hardware-specific mediums, the disadvantage of not being compatible with each other makes searching for a better solution imperative. Almost every device that was tested had a camera integrated, apart from Dell Axim x51v. The average image resolution ranged from 1.3 to 3.2 MP. It is important to note that static output formats support higher resolution, when compared to animating content, because most cameras offer reduced quality for streaming configurations. The maximum resolution that has been received from any utilised imaging sensor was 2 MP. The graphic pipeline, starting from the camera driver and ending at the display, is one of the most demanding components, in terms of hardware resources. That is why the development efforts in this field must be extremely efficient, in order not to overload and restrict the smooth operation of other system components. In the rest of this section the reader will find an analytical description of the requirements that need to be met, in order to form the sensory components infrastructure of the proposed context-aware solution.

Device Requirement 9

Identifier	**DR9**
Description	The mobile device shall be connected to a positioning sensor
Type	Device Requirement
Fit Criterion	The device satisfies the requirement if a connection is established with a sensor, which allows access to the data provided by the global positioning system. Location information should be available to the platform through this communication channel to support potential user requests. This requirement is met if the mobile device can query a GPS sensor for real-time data and the sensor is either connected via Bluetooth or is embedded in the device itself. The communication should take place over a serial protocol (COM Port), supporting at least 4800 BAUD rate.

Device Requirement 10

Identifier	**DR10**
Description	The mobile device shall be connected to an orientation sensor
Type	Device Requirement
Fit Criterion	The device satisfies the requirement if a connection is established with a sensor, which allows access to orientation data. Orientation information should be available to the platform through this communication channel to support potential user requests. This

	requirement is met if the mobile device can query a digital compass or accelerometer for real-time data and the sensor is either connected via Bluetooth or is embedded in the device itself. The communication should take place over a serial protocol (COM Port), supporting at least 4800 BAUD rate.

Device Requirement 11

Identifier	**DR11**
Description	The mobile device shall be connected to a camera
Type	Device Requirement
Fit Criterion	The device satisfies the requirement if a connection is established with a digital camera, which allows the video see-through display to operate. Streaming content should be available to the platform through this communication channel to support potential user requests. This requirement is met if the mobile device can query a digital camera for real-time images and the sensor is embedded in the device itself. In addition, the drivers for this component should be available from the manufacturer and must be compatible with the commercially available development tools.

Performance Requirement 4

Identifier	**PR4**
Description	The positioning sensor shall offer measurements in low latency
Type	Performance Requirement
Fit Criterion	The positioning sensor satisfies the requirement if the latency of accumulated position estimations provides the ability to the framework to offer real-time data processing. Real-time processing is accomplished if several measurements per second can be fused to the framework. This requirement is met if the positioning sensor can reply to more than 1 positioning requests every second.

Performance Requirement 5

Identifier	**PR5**
Description	The orientation sensor shall offer measurements in low latency
Type	Performance Requirement
Fit Criterion	The orientation sensor satisfies the requirement if the latency of accumulated orientation estimations provides the ability to the framework to offer real time-data processing. Real-time processing is accomplished if several measurements per second can be fused to the framework. This requirement is met if the orientation sensor can reply to more than 1 positioning requests every second.

Performance Requirement 6

Identifier	**PR6**
Description	The positioning sensor shall offer accurate measurements

Type	Performance Requirement
Fit Criterion	The positioning sensor satisfies the requirement if the accuracy of the positioning estimations offers actual representation of the user location and does not prohibit or restrict in any way the smooth operation of the framework. Precise data gathered from this sensor is required for the visualisation of real-time information, as well as for further storage and distribution to other entities. This requirement is met if the positioning sensor accuracy is: • Up to 2 metres when operating in optimal conditions o Optimal conditions are defined as a clear sky, with more than 3 GPS satellites in sight, in a specific instance. The height or material of any surrounding objects should not obstruct GPS signal reception. In optimal conditions, 3D positioning should be available. • Up to 5 metres when operating in nominal conditions o Nominal conditions are defined as a partially clear sky, with 3 GPS satellites in visible range. The height or material of surrounding objects may obstruct GPS signal reception to some degree. In nominal conditions, 3D positioning should be available. • Up to 10 metres when operating in the worst case conditions o Worst conditions are defined as a heavily obstructed sky, with not less than 3 GPS satellites in visible range. The height or material of any surrounding objects may block GPS signal reception to a large degree (i.e. urban canyon). In worst case conditions, 2D positioning should be available. Accuracy of this sensor is vital for offering quality system functionalities.

Performance Requirement 7

Identifier	**PR7**
Description	The orientation sensor shall offer accurate measurements
Type	Performance Requirement
Fit Criterion	The orientation sensor satisfies the requirement if the accuracy of the orientation estimations offers actual representation of the user viewpoint and does not prohibit or restrict in any way the smooth operation of the framework. Precise data gathered from this sensor is required for the visualisation of real-time information, as well as to enable advanced interface functionalities. This requirement is met if the orientation sensor accuracy is: • At most 1 degree, with 0.1 degree resolution and 0.5 degree repeatability, for heading measurements. Value range between 0.1° and 360.0°. • At most 1 degree, with 0.1 degree resolution and 0.2 degree repeatability, for pitch and roll measurements. Value range ±60°.

	Accuracy of this sensor may enable the user to perform gesture-based interactions.

Portability Requirement 3

Identifier	**PortR3**
Description	The positioning sensor shall output NMEA ASCII sentences
Type	Portability Requirement
Fit Criterion	The positioning sensor satisfies the requirement if the output is compatible with GPS standard for location coordinates, National Marine Electronics Association (NMEA) 0183 format. This requirement is met if the positioning sensor can translate data signals to the following sentences, described in ASCII format. 1. A parsed NMEA sentence containing a GGA (Public Positioning System Fix Data) command. Latitude, longitude and altitude can be derived from this object. 2. A parsed NMEA sentence containing a GLL (Geographic Position – Latitude/Longitude) command. Latitude and longitude can be derived from this object. 3. A parsed NMEA sentence containing a RMC (Recommended Minimum Specific GNSS Data) command. Latitude, longitude and ground speed can be derived from this object. 4. A parsed NMEA sentence containing a GSA. This sentence provides details on the nature of the fix. It includes the numbers of the satellites being used in the current solution and the dilution of precision (DOP). 5. A parsed NMEA sentence containing a GSV (Satellites in View) command. Data about the satellites that the sensor has found is based on the viewing mask and almanac data.

Portability Requirement 4

Identifier	**PortR4**
Description	The orientation sensor shall output comma separated ASCII sentences
Type	Portability Requirement
Fit Criterion	The orientation sensor satisfies the requirement if the output is compatible with the conditions set by the framework. This requirement is met if the sensor can provide ASCII sentences, composed out of 3 values, which are separated by a comma. The numeric variables should have minimum value of 0.1 and maximum of 360.0, calculated in degrees. The produced sentences should conform to the following format: “Heading, Pitch, Roll”

Device Requirement 12

Identifier	**DR12**
Description	The camera shall offer high resolution images
Type	Device Requirement
Fit Criterion	The camera satisfies the requirement if the resolution of the produced images does not compromise the quality of visual feedback that is presented to the user. The resolution is equally important for the operation of the AR interface, because pixel-level accuracy is required for examining video data and presenting additional information on precise locations in the image coordinate system. This requirement is met if the camera resolution supports QCIF or larger frame sizes. QCIF is defined as 176x144 pixels per frame.

Almost every camera, embedded on the mobile devices that have been tested, produced video content in lower resolutions than the default display resolution. This resulted in non-fullscreen windows, which did not cover the whole display surface. This happens to be the case with most devices currently available on the market, but we speculate that in the short-term future this is going to change. *Aura* was designed to exploit the highest camera resolution available on any device and several functions for cropping and reformatting the video content have been developed. They can fit the video to the given display without the need for introducing additional requirements in this domain.

Device Requirement 13

Identifier	**DR13**
Description	The camera shall offer images in multi-colour pixel formats
Type	Device Requirement
Fit Criterion	The camera satisfies the requirement if it supports high-colour depth configurations, in order to represent real-world information in an acceptable and easy manner for the user to associate with. Additionally, the operation of the AR interface is dependent on the colour-bits per pixel, which are supported by the produced images. HighColor is defined as 65,536 (65K) and TrueColor as 16,777,216 (16M) available colours per pixel. This requirement is met if the camera produces images either in the 16-bit R5G6B5 or the 24-bit R8G8B8 pixel format.

4.1.1.3 Network Infrastructure

In the Literature Review, we examined several ubiquitous context-aware systems that have been developed on top of a range of architectures. Some systems are implemented as standalone applications, where the mobile device solely handles the acquisition of information and, consequently, produces the required functionality. For instance, most commercial Personal Navigation Devices (PND) run autonomously, without the need of establishing a connection to a remote source of context (Raper et al., 2007). The advantage of this method is that it can produce fast, in terms of performance, and robust solutions. Conversely, some systems have been deployed as network-based solutions, where mobile clients make requests, such as identifying POIs in close proximity, to remote nodes. This method can be more informative since the centralised node can aggregate data from various sources (e.g. database or another client device). In contrast, low network bandwidth can severely affect the user experience and the total loss of connectivity would render the service inaccessible. Furthermore, certain modern location-based services (e.g. LOCUS) have adopted an alternative approach, which tries to solve most identified problems and build on top of existing architectures. These solutions establish three-tier architectures, which attempt to capitalise on the benefits of the standalone, as well as the two-tier models (Raper et al., 2007), but also inheriting some of the drawbacks of both methods.

Being involved with the design and development of the LOCUS research project, the author gained invaluable insight and experience about various issues that affected the operation of such context-aware internetworking systems. LOCUS operates on top of a three-tier architecture, distributed between the mobile device and one or more remote servers. The first tier consists of the mobile web browser, customised with additional rendering components. The primary responsibility of this tier is to handle the presentation of the mixed reality user interfaces. Mobile web browsers, such as *Pocket Internet Explorer* (PIE) and *mobile Opera*, are highly sophisticated and their latest versions can fully support the advanced functionalities of client-side scripting (e.g. JavaScript) and COM integration. The ability to distribute some of the application logic to the client improves user interaction with the web application, which implies that the need for data exchange between the client and server is minimised. The second tier consists of a local HTTP server, also installed on the client device. This is quite a popular option for mobile web applications, which adopt the client-server model and benefit from adherence to web standards. The local HTTP server receives requests from the local-client tier and responds to these requests directly, if the local cache of data and logic allow it to do so. If the local HTTP server cannot satisfy the request it then acts as a client, relaying the request to the third tier, which is formed out of one or more remote servers. Communication between the local server on the mobile device and the remote servers takes place via HTTP requests sent over mobile data connections, such as GPRS or UMTS. The benefits for adopting this three-tier architecture provide much flexibility for the implementation of such systems, allowing the developers to choose where to store data and from where to execute the functionality. This way, performance optimisation is promoted according to several criteria, like reduced latency. A practical approach for the user is to download and cache what is expected to be the most commonly accessed and relevant data when they obtain a high-bandwidth low-cost connection. The most relevant data is likely to be derived by a user's current or future location including local data, such as 2D mapping, 3D virtual scenes and themed content in the form of spatially referenced information. The local cache serves two major purposes. Initially, it ensures that some data is available in every case, regardless of the current wireless connectivity status, and it also enables prompt responses to user queries by accessing the local cache when available. However, only a small subset of potentially relevant data can be stored at the local layer. To access richer datasets or real-time dynamic data, the request must be passed on to the remote server tier. By caching logic at the client tier, the local-server tier or the remote-server tier developers can enhance the performance of mobile web applications. Some applications, which run entirely on the local device and have no need to execute equivalent logic on a remote server, may benefit from the reduced latency associated with the transfer of data over mobile networks. Alternatively, certain elements of the application logic may be too sophisticated for the mobile device to process, thus execution needs to exclusively take place on the remote entity. Further information about the architecture of the LOCUS project can be found either online at the official website or in the 8th publication listed in Chapter 1.7.

Although the architecture of LOCUS has proven to work quite well when applied on scenarios that need to satisfy certain user needs, such as wayfinding in unknown environments, for the development of *Aura* a different approach was necessary. The main reason for the introduction of an original approach is the need of *Aura's* users to communicate with various kinds of entities. Fundamentally, LOCUS was designed to query a centralised architecture (i.e. a remote web server) and retrieve spatially referenced information about relevant points or objects of interest. However, the specifications of *Aura* directed supplementary communication with independent environmental entities in a peer-to-peer fashion. The common factor that *Aura* and LOCUS share is that wireless network infrastructures can form the platform, which enables local clients to transparently connect and share context with distant entities.

However, for *Aura* these entities may either be remote actors sharing similar interests or participating in a group task or a centralised system that distributes suitable information. In this sense, *Aura* does not require the application logic to be stored in a remote location, because it only needs to exchange descriptions and spatiotemporal context that describe the remote entity. As a result, the tweaked standalone model, which has been introduced in *Aura*, comprises both the logic and data that is required for successful operation in the local system. Thus, the framework acts as a mediator and controls the *rules of engagement* between the participating entities.

Our framework regards the network as a valuable source of context, which provides resources to the context-management engine and, consequently, to the visualisation interfaces. Each networking stack introduces its own boundaries (e.g. operational range) that may restrict the omnipresence of mobile applications (Papakonstantinou and Brujic-Okretic, 2009a). Furthermore, wide-range service providers apply restrictions on arbitrary data exchange. Specifically, P2P interactions and service hosting on mobile devices are very challenging and costly to implement because telecommunication proxies mask network addresses, thus rendering the distribution of content limited to request/response mode, always client-triggered, and not proactive. Some other issues that pervasive system developers need to control while developing multi-user Collaborative Virtual Environments (CVE) are those inherited by the distributed networking research described in more extent in the Literature Review. Namely, these include high latency and low bandwidth management, heterogeneity of infrastructure, scalability issues, synchronisation, failure handling and security and privacy issues.

A middleware component has been introduced in the proposed framework, in order to satisfy the requirement of invisibility (Saha and Mukherjee, 2003). The purpose of this entity is to establish a connection between the network nodes, which act as source of context, and the local context-management layer. Communication on this channel is bidirectional. Data received from remote entities is transformed and fused into structures that can be queried by the application and vice versa (Papakonstantinou and Brujic-Okretic, 2009a). This way, the higher layers and, effectively, the user do not need to keep any specific information about the remote entities or how to contact them. The LOCUS architecture uses a widely available and tested protocol for exchanging data, the HyperText Transfer Protocol (HTTP). This protocol implements the application layer of the Open Systems Interconnection (OSI) networking stack. The rationale behind this selection is mainly attributed to the compatibility of this protocol with several browser-based applications on which LOCUS architecture has been based. However, the architecture of *Aura* required a more efficient way to distribute information without transmitting redundant data like *meta-tags*. This way, the amount of exchanged information could be effectively minimised, resulting in a less expensive communication model. That is why a custom protocol has been designed and implemented in the middleware component for the purpose of the research.

Likewise, the networking protocol developed for *Aura* works on top of the TCP/IP network stack. It can be considered as an application layer protocol capable of exchanging in real time either text messages between users or text descriptions about POIs, as well as position and orientation information for the participating users and POIs. Network packets are switched by using UDP between the *transport* layers of remote systems, whereas datagrams are used to transfer the UDP packets between the Internet layers of the involved systems. The *physical* layer (i.e. level 1 of the ISO OSI reference model) is responsible for transferring single bits. Loosely speaking, this process is realised by the modem of the mobile device. The wireless communications medium provides a service to the physical layer, namely, it transfers radio signals from one extreme of the medium to the other. The modem at the transmitting device takes bits and encodes them as signals on the transmission medium. The modem at the receiver interprets the signals it receives as either one-bits or zero-bits. In most cases, this interpretation will be correct but sometimes corruption or interference of the signal on the medium will lead the receiving modem making an inaccurate interpretation of the data. This is known as the error of the service. It is the job of the device manufacturer and, by extension, of the developer to achieve the minimum error rate, subject to cost and other constraints. Typical error rates vary from 10^{-4} (i.e. low grade telephone line in the 1960's) to about 10^{-12}. The service provided by the physical layer is therefore the unreliable transfer of bits.

Currently, the optimal protocol for interconnecting distant parties is UMTS, which is based on Wideband Code Division Multiple Access (W-CDMA). *Aura* is not bound on this protocol but it is the current standard in European 3G mobile telephony. Moreover, network protocols offering less bandwidth would be characterised as inappropriate, because of the continuous and high volume of data that may be exchanged (e.g. 3D virtual scenes) through our application. The maximum bandwidth of UMTS is 21Mbps but the nominal, which is currently offered by network operators, is between 3.6Mbps and 7.2Mbps. UMTS implements the first 3 layers of OSI and can provide both circuit-switched and packet-switched data communication. AAL2 handles circuit switching and AAL5 packet switching. The flexibility of UMTS is great and could be a global telecommunication protocol if its routing capabilities were fully taken advantage of. To demonstrate that the application-level protocol utilised by the middleware is not dependent only on UMTS, as well as to be able to accomplish everyday debugging and testing procedures, another wireless short-range protocol has been put to work. This is the IEEE 802.11 WLAN family of protocols, which are implemented on most current mobile devices. Every device that was tested during the course of this project offered such wireless functionality, although the latest supported high-speed versions (i.e. 802.11g) without observing any compatibility issues between them. The maximum throughput of 802.11a, 802.11b, 802.11g and 802.11n is theoretically 54Mbps, 12Mbps, 54Mbps and 108Mbps respectively, even though none of the tested devices supported the draft-*N* version. Apart from the throughput, some other differences between these variations include the operating frequency bands (i.e. 802.11b/g/n - 2.4GHz, 802.11a/n - 5GHz) and the variable modulation types, such as Direct Sequence Spread Spectrum (DSSS) and Orthogonal Frequency-Division Multiplexing (OFDM).

The Literature Review and this chapter demonstrate that the optimum underlying telecommunication topology, which can support our goals, is a decentralised one. The middleware component of *Aura* employs a specific subset of this topology. In more detail, peer-to-peer connections have been established according to a hybrid distributed topology, which is presented in Figure 4-1. Long-range communication between the entities of the system can take place according to the patterns that are illustrated in Figure 4-2. Figure 4-1 presents two device types, a thin and a thick client. The thin client supports only context acquisition and transmission services, whereas the thick one is also capable of advanced information visualisations. Both have been extensively described in a former paragraph. In turn, Figure 4-2 depicts the possible communication channels between two clients and between a client and a server.

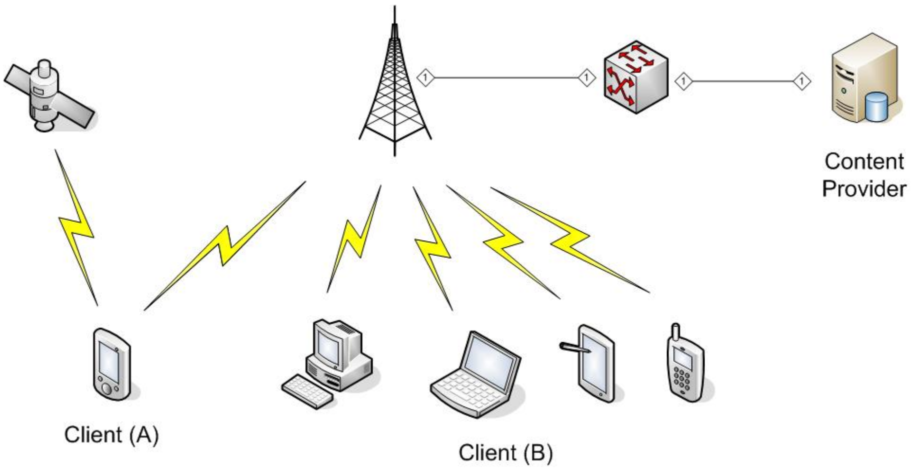


Figure 4-1: Hybrid Decentralised Network Architecture

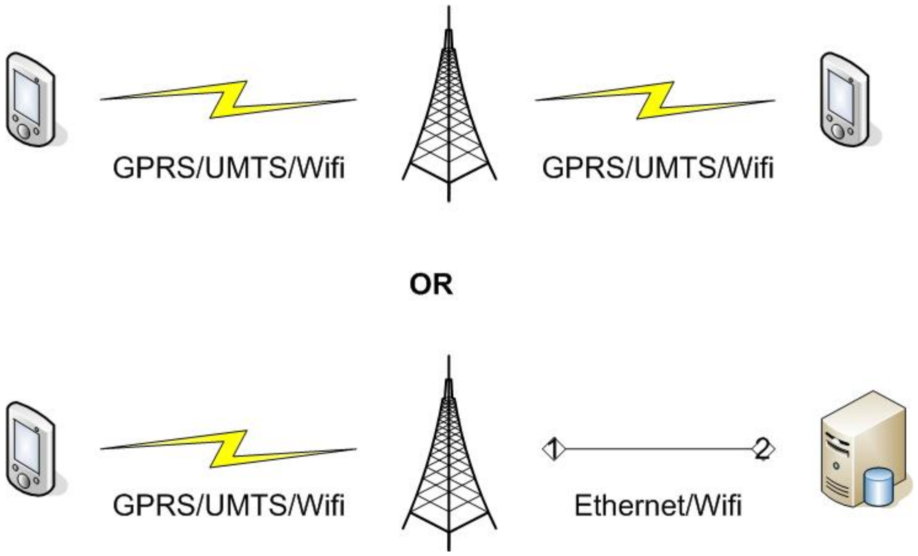


Figure 4-2: Supported Peer-to-Peer Communication

Device Requirement 14

Identifier	**DR14**
Description	The mobile device shall offer wireless connectivity with remote entities
Type	Device Requirement
Fit Criterion	The device satisfies the requirement if a wireless connection is available, enabling networking with remote entities. The remote entities could either be thick or thin client devices, which can offer only limited context acquisition and transmission services. In addition, if a centralised architecture is available, communication should predominantly take place over this wireless channel. Wireless connection with the remote entities should be accessible over any of the following protocols, which are currently available in the European Union for commercial use. Long Range • Global System for Mobile communications (GSM) • General Packet Radio Service (GPRS) • Enhanced Data rates for GSM Evolution (EDGE) • Universal Mobile Telecommunications System (UMTS) • High-Speed Uplink Packet Access (HSUPA)

	- High-Speed Downlink Packet Access (HSDPA) Short Range - Wireless Local Area Network (WLAN - 802.11a/b/g/n) This requirement is met if the mobile device can act as a transceiver of the aforementioned signal types and interpret them to contextual information, which can be translated by the framework.

Every protocol that has been mentioned offers packet-switched communication, enabling the development of a dedicated custom high-level protocol that can transfer information between remote entities. These protocols have been selected because they implement the lower layers of the OSI networking stack. As a result, higher TCP/IP services can be developed and governed by the framework.

It is evident that there is a difference in the operating range of the two groups of protocols. In certain scenarios, introducing wireless repeaters compatible with the current Wireless Distribution System (WDS) standard, described in RFC 802.11, can extend the range of WLAN connections. This can broaden the effective communication range, but it requires additional equipment (i.e. network bridges) to be populated in the environment.

Operational Requirement 1

Identifier	**OR1**
Description	The networking middleware shall be able to work in a request/response mode
Type	Operational Requirement
Fit Criterion	The middleware satisfies the requirement if it can transmit requests for specific information to another party and receive responses, which could then be translated and stored by the framework. The connection should take place between two entities in peer-to-peer fashion. This requirement is met if the middleware can act either as a server or as a client and this variable functionality does not prohibit or restrict in any way the flow of information. Additionally, the user should not be affected by the different modes of middleware operation, apart from initiating or accepting a potential link to a remote entity.

Operational Requirement 2

Identifier	**OR2**
Description	The networking middleware shall be able to work in a connectionless/ asynchronous mode
Type	Operational Requirement

Fit Criterion	The middleware satisfies the requirement if it does not need to establish a connection with a remote entity but is capable of exchanging arbitrary messages at random intervals or continuously for the required time span. This requirement is met if the networking protocol can transmit and receive information only when it is proactively required by the middleware or explicitly by the user.

Due to the nature of mobile communications and ISP networking architectures, a static connection would require continuous exchange of data just to keep the connection alive. Such functionality would obviously increase the cost of communication. This happens because currently the charges for most mobile broadband packages are calculated based on the volume of exchanged data. This would render system operation prohibitive for a wide commercial audience, due to high price tags.

Operational Requirement 3

Identifier	**OR3**
Description	The networking middleware shall be able to exchange text messages between users
Type	Operational Requirement
Fit Criterion	The middleware satisfies the requirement if the developed protocol can transmit, as well as receive text messages generated by the users of the framework. Text messages are defined as strings of characters. The maximum length of the string can be up to 256 bytes. This requirement is met if the networking protocol is capable of transmitting and receiving such messages, disassembling them into words and passing them to the middleware, so that they can be presented to the local user.

Operational Requirement 4

Identifier	**OR4**
Description	The networking middleware shall be able to exchange text descriptions of POIs
Type	Operational Requirement
Fit Criterion	The middleware satisfies the requirement if the developed protocol can transmit, as well as receive text descriptions generated automatically or manually by remote users of the framework. Text descriptions are defined as strings of characters. The maximum length of the string can be up to 256 bytes. This requirement is met if the networking protocol is capable of transmitting and receiving such descriptions, disassembling them into words and passing them to the middleware, so that they can be presented to the local user, through the selected user interface.

Operational Requirement 5

Identifier	**OR5**
Description	The networking middleware shall be able to exchange location information of selected entities
Type	Operational Requirement
Fit Criterion	The middleware satisfies the requirement if the developed protocol can transmit, as well as receive, position information generated automatically or manually by remote entities. Position information is defined as strings of characters, which describe the latitude, longitude, altitude, datum and grid of a specific POI. The maximum length of this string can be up to 256 bytes. This requirement is met if the networking protocol is capable of transmitting and receiving such information, disassembling them into position structures and passing them to the middleware, so that they can be presented to the local user, through the selected user interface.

Device Requirement 15

Identifier	**DR15**
Description	The mobile device shall offer Bluetooth connectivity with local sensors
Type	Device Requirement
Fit Criterion	The device satisfies the requirement if the Bluetooth networking stack is implemented both in hardware and software terms. The device should have an embedded chipset that offers Bluetooth connectivity to allow the framework to establish connections with the local sensors. The software drivers should enable utilisation of any number of Bluetooth services, but - most essentially - the serial communication protocol (COM) capable of translating ASCII sentences generated by the underlying hardware. This requirement is met if the mobile device can act as a transceiver of Bluetooth signals and interpret them to contextual information in the operating range of 2 metres.

Older mobile devices, which have been tested for compatibility issues, allowed only one serial connection to be established between the device and the remote sensors. This limitation prohibited concurrent connections between the device and more than one sensor at any given instance. As a result, the user had to select the most vital sensor for system operation. In most cases this was the GPS device, which resulted in the loss of orientation information. More recently released devices have managed to overcome this challenge by becoming capable of realising two or even more concurrent connections. As a result, both sensors could be paired with the framework and offer the required data without any evident problems. This could result in the addition of an extra requirement, which would specify the minimum number of serial channels, but a large proportion of the devices that were affected by this issue have embedded a GPS sensor and did not pose an actual problem to the system operation. In such a case, position data was retrieved through the internal communication channel and orientation data through the external channel.

4.1.2 Software-based Components

In the previous paragraphs, some differences between *Aura*, the framework developed for this research project, and LOCUS, a system developed for the purposes of an EPSRC research project, have been described. Nevertheless, the most important differentiation between the two architectures has not yet been explored in detail. LOCUS objectives instructed that compatibility with existing web standards should be respected. In contrast, *Aura*'s objectives direct that the system should be designed in a way that it would be available to commercial audiences, without the application of any additional requirements. Commercial acceptance of mobile applications depends on several factors, mainly affected by the preferences of end-users. Many research products have failed to achieve commercial growth because end-users need to acquire practical benefits from the use of the system. Apart from LOCUS, the architecture and functionality of several similar projects have been examined, as seen in the Literature Review. Some of the requirements of these projects were similar to those of *Aura*, but most did not satisfy all of our needs. That is why some essential technical conditions have been studied and an effort has been made to apply them in the development of our framework, in order to gain some of the advantages, which have been previously validated.

The main architectural difference between LOCUS and *Aura* is that the former is based on web browser integration, separating the logic, data and presentation between a server and a client, whereas the latter can work as a standalone system with similar effects. A major reason behind the adoption of this model is that the framework has the potential to work in realistic ubiquitous scenarios. User experience can be severely affected by mixing information that is either persistent in the real world or artificially produced by the mobile device, when the result is visualised by the user on this medium. The benefits of adopting this methodology have been reproduced in various applications, spanning across location-based services and events management through to entertainment. Certain technological advances in the field of mobile computing have created the technical infrastructure for current devices, which are able to sense context, manage it accordingly and offer immediate support to the user by presenting valuable content through advanced user interfaces. The design of *Aura* tries to benefit by utilising the technical capabilities of the mobile device, without the need to contact a remote server for further processing. Contacting a remote entity is not cheap according to current long-range communication pricing models and should take place only when remote data is considered vital for the information needs of the user and under explicit manual control.

4.1.2.1 Development Platform

Portability Requirement 5

Identifier	**PortR5**
Description	The framework shall be compatible with the Windows Mobile platform
Type	Portability Requirement
Fit Criterion	The framework satisfies the requirement if it can be installed and executed on any device that runs Windows Mobile 5.*x* or later, Smartphone or Pocket PC edition. The same version of the framework should be compatible with all devices supporting the mentioned operating systems. This requirement is met if the framework can run without any problems affecting its users on every device that supports a subset of the 32-bit Microsoft's Windows CE 5.*x* or later OS product family.

At the beginning of the system design process, the author initiated an exploration for the most suitable platform, which could sustain performance-oriented and resource-efficient development on mobile devices. The available options at that time were Symbian's *SymbianOS* and *UIG*, Palm's *PalmOS* and Microsoft *Windows Mobile* platforms. Google's *Android*, Apple's *iPhone*, Qualcomm's *BREW* and *BlackBerry* platforms were not available when the research commenced. Experience and further research on these issues indicated that commercial mobile devices available in 2005/6 were not very powerful, in terms of processing power and graphics rendering capabilities. The underlying requirements, which had been identified at that time, illustrated that the development platform should support new device models that were going to be produced in the rapidly evolving mobile market. In addition, integration with non-mobile platforms was inspected, in order to verify if potential mobile functionality could be blended in a non-mobile platform. In case this happened and the author had selected a different platform, certain aspects of this research would immediately be considered obsolete. The last factor, which governed the selection of a single platform, was concerned with the integration of 3rd party software components that would potentially be utilised in the development process of the system. This was clearly a personal choice, which sprung out of existing development experience, as there was an evident need to avoid re-engineering of already available technologies. The optimal solution was found to be the Windows Mobile platform because it did satisfy the requirements, described earlier and because it also offered good support for application developers.

There is a variety of different programming language types available for the Windows Mobile development but only few can offer the potential to produce autonomous mobile applications supporting every requirement. The available languages are *Visual C++*, *Visual Basic .NET*, *C#* and *Java*. Developers can choose between native, interpreted and managed code, although every device does not support all options. Interpreted development is not recommended and was discarded, mainly for performance issues. This was verified after having worked with the LOCUS platform, which focused the development efforts on scripting technologies like *JavaScript*.

In most cases, developing in *C* and *C++* allows the programmer to access very low-level functions, as well as specific memory locations. Development on such low-level languages produces the fastest possible solution, but it also presents the greatest challenges in terms of stability and development time. It is easy to make mistakes in a C program, while C++ is slightly better organised and provides object-oriented capabilities. Moreover, conceptual mistakes, which can cause critical problems in the long-term (i.e. memory leaks), need to be identified early and avoided. A large proportion of the time required to develop the framework was spent on debugging, for finding such errors and fixing them. Development on the device took place by using embedded C++, with the support of Microsoft's Visual Studio 2005/08 IDE. An application which runs on Windows Mobile 5.*x* or 6.*x* (either PocketPC or Smartphone editions) will work on other devices, which depend on that platform, regardless of manufacturer or model. Conversely, there is occasional incompatibility between certain versions (especially Windows CE 4.*x* and downwards) and the choice of development environment must depend on the targeted devices.

Java has long been promoted as the best development language for mobile devices, in terms of platform independence, and certainly dominates mobile phone software. While Java was originally developed for resource-constrained devices, it quickly became clear that it was far too resource-voracious for executing advanced visualisation engines. While the strength of the language has enabled it to successfully spread into web services and server-side management a different approach was required, on limited-resource devices. Java did not use to be very well supported on mobile devices and even today many PocketPC devices either do not have a Java Virtual Machine (JVM) installed as default or it is provided on accompanying software for the consumer to install individually if required. Manufacturers of PocketPC devices seem to have difficulty selecting which Java profiles to implement, even when the hardware specifications of the device enables advanced functionalities. MIDP offers a wider range of available software, while Personal Java is more resourceful. This means that a developer has no guarantee that a specific Java profile will be available on every PocketPC device and, as a result, Java development on the Windows Mobile platform is effectively restricted to research or enterprise use. In contrast, on other kinds of mobile devices programming support is based on Java and specifically on MIDP 2.*x* over CLDC 1.1. Interestingly, there are not many devices available that support CDC and the only compatible models that we could gain access to were Sony-Ericsson P990, M600i, P1i and Nokia 9*xxx* Communicator Series. The following section of the report offers an explanation of the importance of CDC.

4.1.2.2 Software Libraries

A number of external libraries have been introduced for the implementation of each client interface. To present the VR environment, ParallelGraphics *Cortona* System Development Kit (SDK) was selected. It can load VRML-format models and offers operations and properties that can lead to the realisation of the required 3D functionality. Cortona provides an API that enables the integration of ParallelGraphics 3D technology into the application by using several programming languages that support COM. Virtual Reality Modelling Language (VRML) is considered as a web inter-exchangeable format for 3D graphics, which can be expanded to hold geo-referenced information about the surrounding environment and its contents (i.e. GeoVRML extension). Unfortunately X3D (Web3D Consortium, 2011), the successor to VRML, is not supported by *PocketCortona* that is used on mobile devices. This does not pose any major restrictions to the quality of our system because in newer versions we attempted to replace these standards with the more advanced OpenGL rendering libraries, which are presented in more detail in a following paragraph of this chapter.

Furthermore, in order to receive context from the GPS receiver, the employment of a group of classes that facilitate such operations was required. This was accomplished by selecting Franson *GPSToolsCE* ActiveX library. It is a cost-efficient way to develop GIS applications that work on mobile platforms. Moreover, it provides access to GPS position, satellite information and derived context by supporting the NMEA 0183 protocol. Another important feature is that it can convert positions between several coordinate systems by using various grid and datum options. In version 2.20, surprisingly, two additional components have been introduced, *GpsShapeCE* and *GpsViewCE*. By exploiting them, 2D map visualisations and a small amount of GIS functions could be enabled. The map component complies with Shapefiles (i.e. .shp, .shx), which is a very common format for exchanging spatial data. It is the basic format exported by ESRI *ArcGIS* (ESRI, 2011).

For acquiring and presenting live video streams on the AR interface, the most satisfying solution was found to be the one that produced a reusable component (ARIE), which can be integrated in any host application, with possible diverse lower-level prerequisites. The component enables the introduction of a media-streaming layer that resides between the camera device drivers and the presentation interface. Microsoft's *DirectShow* (Microsoft, 2010) was proven sufficient because it is currently embedded in the device operating system and its functions are delivered through native commands. The advantages of DirectShow include its flexible for development and easily expandable architecture and the integration of reusable mechanisms that are called filters. These filters handle the streams of multimedia content such as video, sound and still images. DirectShow filters are the basic building block, which receive video content from a capture device, encode/decode multimedia streams and fuse content to the graphic and sound hardware. The filters are classified into 3 distinct categories, the source, the transform and the rendering filters. In our case, the source filter enables the camera caption functionality on top of the camera drivers. This filter has only output pins that provide video content. Initialisation of the video capture filter with the default camera driver is a procedure that takes place during the beginning of execution. Saving animating content is not a requirement of *Aura*, thus the rendering filter has been set to preview mode, which minimises the reserved operational resources. In addition, the development of a transform and rendering filter was required in order to operate the AR interface. The transform filter has been used by the framework to retrieve and augment the original video frames, whereas the rendering filter's functionality is to present the aggregated video content. Further analysis of the filters developed during this project can be found in Chapter 5.5.6. ARIE has been integrated as a fundamental component in the development of the architecture of LOCUS, in order to establish the required AR functionality.

Overlaying additional elements on the AR interface requires the introduction of a rendering programming element. The available APIs, which allow such functionality to be executed on the Windows Mobile platform, are *Mobile 3D Graphics* (M3G), *OpenGL ES* (OGLES) and *Direct3D Mobile* (D3DM). M3G (JSR-184) is the standardised API that is used to access low-level 3D functions in applications that are implemented in Java. Alternatively, applications that are written in native C/C++ code may have access to the 3D subsystem either by using OpenGL ES or Direct3D Mobile. Both M3G and D3DM are high-level rendering libraries, which are built on top of the OGLES low-level rendering engine. This is found only on mobile configurations. Additionally, D3DM drivers can be found only on Windows Mobile devices, which effectively renders this API unsuitable for an open standard. An extensive comparison between these APIs has been published (Pulli et al., 2005), which illustrates that native assembly code is the fastest solution on mobile platforms. The following figure illustrates the obtained results.

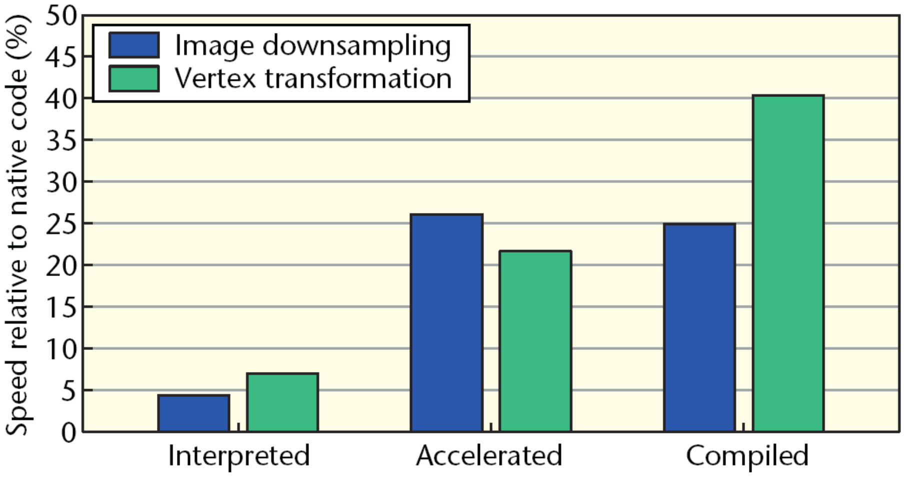
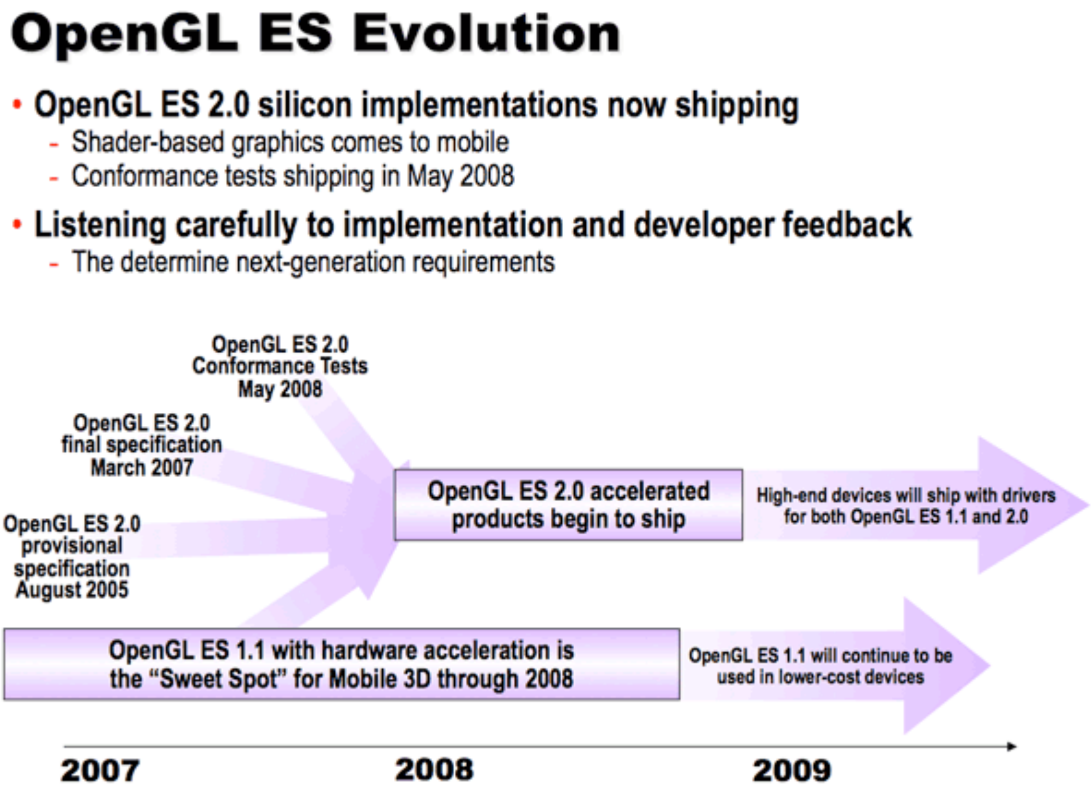


Figure 4-3: Execution speed of interpreted, hardware accelerated, and just-in-time compiled Java bytecode relative to native assembly code

As a result of the previous observations, the best recognised solution to implement the graphics engine of *Aura* was found to be OpenGL ES (Embedded Devices). Khronos Group has developed this API. It aims to adopt the functionality of the full OpenGL API, but with the restrictions applied by mobile device hardware. OGLES supports two versions. OGLES 1.*x* targets fixed function hardware and is defined as relative to OpenGL 1.5 specifications, whereas OGLES 2.*x* targets programmable hardware and is defined as relative to OpenGL 2.0 specifications (Khronos, 2011). Furthermore, OGLES 1.*x* supports two profiles, *Common* and *Common Lite*. The former requires a mathematic co-processor to exist on the hardware, which enables the conversion of double-type to float-type variables and introduces fixed-point data types. The latter profile does not use decimal point variables and all values need to be translated to integer types. Freeware implementations of OpenGL ES have been provided by the *Vincent3D* rendering library (Vincent Pervasive Media Technologies, 2008) and by the *PowerVR* SDK (Imagination Technologies, 2011). Both implementations target OpenGL ES 1.*x*. The latest Windows Mobile devices support this version. During the development of *Aura*, only HTC Touch Diamond offered hardware acceleration of 3D graphics and included drivers for the Qualcomm chipset, developed by ATI.



OpenGL ES Evolution

- OpenGL ES 2.0 silicon implementations now shipping
  - Shader-based graphics comes to mobile
  - Conformance tests shipping in May 2008
- Listening carefully to implementation and developer feedback
  - The determine next-generation requirements

Figure 4-4: OpenGL ES Evolution (Khronos, 2011)

The distributed software infrastructure intended for the development of the system has been a complicated issue. That is because it is heavily dependent on the underlying operating system and hardware. The initial choice was to build the architecture on top of Java RMI. It appeared suitable because it is provided free of charge by Sun/Oracle and the development can take place on almost any platform. Evidently, the utilised programming language would be Java (J2SE) for server and J2ME for mobile devices. For the server, the RMI package is included in J2SE specification. This is not the case with the mobile environment, though. J2ME considers RMI as an Optional Package (OP). This would not be a challenge if no other restrictions existed. RMI OP must run over MIDP's Content Distribution Internetworking (CDC) profile. The availability of devices supporting this technology is very limited, as described in a previous paragraph and it definitely does not reflect the average specifications of the majority of devices. Another disadvantage of RMI OP is that it poses some restrictions, in comparison with the desktop equivalent, and while it is method-based, objects can be passed only by value. Additionally, DCOM from Microsoft was not appropriate because the solutions that utilise this technology can only run on windows-based desktop clients and mobile phones are still not supported. While it would not be a consideration if clients consisted exclusively of high-end smartphones, inheritance is not supported and objects are stateless in DCOM. In contrast, CORBA is supported on all platforms and does not have any development-associated issues. However, we avoided developing a CORBA solution because of its complexity and because there have been other, more modern solutions available. Mobile agents were considered as a very good alternative. The advantage of mobile agents is the mobility of code, which can bear large volumes of data, as well. While this minimises the frequency of achieving a connection, it lengthens the amount of time required to transfer data. Moreover, web services and servlets are the current trend in mobile communications. Certain solutions based on Jini and other server-side technologies for XML management and distribution were also considered. A review of such technologies and the advantages and disadvantages that have been identified in their implementations can be found in the Literature Review.

4.2 Desired Functionality Analysis

In 1991, Weiser stated that the most useful technologies are transparent to the user (Weiser, 2002). The reason that makes them invisible is *tacit dimension* (Polanyi, 1966), which relates to knowledge management research. Currently, we observe this phenomenon in many aspects of life with the use of mobile phones as being the most distinctive case in point. It is considered a communication device with nice stylish touches, but people tend to forget how many layers of diverse technologies are effectively cooperating. Weiser realised that the devices employed in Xerox PARC labs needed explicit user attention because they were conceived as isolated objects of the environment. This made computing technologies more difficult to embed in daily activities. Current mobile devices, though, are recognised as indispensable accessories for almost every person willing to obtain advanced information and communication services.

Interestingly, Weiser (Weiser, 1993) (Weiser, 2002) talked about virtual reality as the technology that sustained the potential to make computing devices invisible to the user. He also identified two main reasons that prevented it from achieving that goal. Firstly, that it could not produce an adequate simulation of the real world that could immerse the user in the simulated environment and secondly that the primary challenge of VR was abstract simulation and not better integration of computers into human activities. During the time that Weiser published that, advances in VR and computer-generated graphics had not reached today's high point of development (Lawton, 2006) and augmented reality was still, conceptually, in its infancy (Milgram and Kishino, 1994), without the progress described in the survey of Azuma et al. (Azuma et al., 2001).

Some very demanding, in terms of advanced UI development, types of pervasive systems are pervasive entertainment applications. The reason that such systems depend strongly on interfaces is that, apart from the spatiotemporal expansions that they need to handle, it is equally important to accommodate social behaviour between various users (Montola, 2005). In these publications (Magerkurth et al., 2003) (Roecker et al., 2007), the authors argue that traditional computer games require the user to focus on the interaction platform, which decreases actual social communication between people. This is one of the requirements, which pervasive games are called to fulfil and attract people that are not familiar with such social activities. There are many kinds of pervasive entertainment applications (Magerkurth et al., 2005), but we focus on truly ubiquitous ones that can be played in large unconstrained environments. Most location-based games that have been studied by Rashid et al. (Rashid et al., 2006) belong to this category, but there is potential to offer more pervasive, towards the user's surroundings, interfaces. Hinske et al., include their own definition of pervasive games and underline the need that that they must take place in mixed reality environments (Hinske et al., 2007). These pervasive applications require average physical, mental and social behaviour, compared to the other kinds, but their immersion capability is greater than the physical or virtual equivalents. MR is a concept that tries to connect the physical with a virtual world through technologies such as AR and VR, which augment the real world scene with artificial enhancements.

Our prototype system utilises the dominant virtuality techniques (i.e. AR and VR), in addition to 2D representations of the physical world. Three different visualisation and interaction interfaces have been designed due to the diverse needs that have been described. Each interface has its own requirements and provides advantages and disadvantages over the others when put to work. More specifically, their input and output mechanisms are fundamentally dissimilar and serve different purposes. Each interface can be called for input, output or both. The next three sections analyse in detail the challenges between users, mobile devices, sensors and pervasive services that have been identified while designing the interaction framework. These issues are explored in terms of system requirements.

Documentation of the proposed requirements was triggered by two sources; either by research in the field of context management, information visualisation and interaction design or by the results produced by the *Requirements Acquisition Survey* conducted for the purpose of this study. User familiarity with existing systems and applications has been described in the previous chapter, whereas in this section the reader will find the needs expressed by potential users, derived from the main part of the user survey.

4.2.1 Context Acquisition & Management Requirements

Context-awareness and pervasive computing are two interrelated terms. Mostéfaoui et al. (Mostefaoui et al., 2004) argue that the ground for associating these concepts springs from the participating elements. Specifically, the attributes of *heterogeneity* and *ubiquitous* presence are those that trigger pervasive systems to adapt to modification of measurable context. In a more formal approach that guides the design process of pervasive applications Grimm et al. (Grimm et al., 2004) name three global requirements. *Embracing contextual changes* in a pervasive architecture is crucial because the user has to be minimally distracted while performing an action. Additionally, *ad hoc composition* and real-time *context sharing* has to be automatically managed by the underlying framework. In this research project, the former requirement is employed in terms of variable number of elements of one implemented type, rather than of multiple random types. In more detail, the following system analysis describes composition and management of entities, such as additional users and not new context-sensitive sources, like sensors. Exchanging contextual information of any type has to be accommodated in a manner that links not only to the interface, but also to the middleware and network availability of the pervasive service. Perceiving and transparently adapting to context changes is therefore one of the primary goals of pervasive system development.

It is doubtful that a solitary communication interface residing between the user and the mobile device will be able to detect and accommodate variations in all events triggered by complex context-aware engines. This happens because simulation and processing of the acquired data needs to be relevant to the targeted functionality of the service. Many researchers (Schilit et al., 1994) (Dey et al., 1999) (Schmidt et al., 1999) (Chen and Kotz, 2000) (Gwizdka, 2000) (Myrhaug and Göker, 2003) (Korpipaa et al., 2003) (Reichenbacher, 2007) tried and successfully classified context variables, but as Mostéfaoui et al. stressed most early classifications are domain-specific (Zipf, 2002) or have been examined from a different research perspective (Mostefaoui et al., 2004). It is important to take all of these categories into consideration, when we select the kind of contextual information to include in our system in order to create abstraction mechanisms that manage and distribute it effectively. For mobile applications, where "*the term mobile can be distinguished as applying to those systems designed to support terminals that are in motion when being used*" according to Anderson (Anderson, 2003), what is considered to be most valuable is location context. The main reason is that the service availability is not inherited exclusively from the place or service provider but also from the surrounding objects of interest and physical entities. In addition, Margerkurth et al. believe that traditional computer games do not trigger physical or social interaction between their users (Magerkurth et al., 2005). Consequently, there is a need for truly ubiquitous applications and games that utilise obtained context, like social activity, while still taking advantage of the processing and interaction capabilities of mobile devices.

A solution for determining the context type is to examine the method that was used in order to acquire it. Hence the categorisation scheme established by Mostefaoui et al. includes *sensed*, *derived* and *explicitly provided* context (Mostefaoui et al., 2004). Sensed context can be retrieved through hardware sensors like the position of an object through a GPS device. Derived context is calculated out of previously stored context.

The velocity of an object can be perceived by implementing the simple formula of the distance completed (i.e. distance between two points) divided by the required time (i.e. $v = (s_2-s_1)/t$). For derived context, more than one measurement is required. The plethora of attributes is directly proportional to the quality of the available sample. Explicitly provided context is passed to the device, through an interaction interface by the user. Depending on the interface options, the user has several means of applying the state that he or she is currently experiencing, as well as other private or public knowledge. Because the discussion is about pervasive applications, we have to note that providing context explicitly to the system may not fulfil the requirement of invisibility that is discussed extensively by Satyanarayanan (Satyanarayanan, 2001). Likewise, the enabling technology for ambient game-like applications demands unobtrusive interfaces, which offer to their user the option to act or ignore the underlying system request for input (Eyles and Eglin, 2007). Even though determining the source is an evident process for acquiring and handling context, it may prove ambiguous because some attributes may fit into more than one category. For instance, orientation of a moving object can be obtained by calculating the direction of movement between the previously obtained position and the current. It can also be precisely sensed through a digital compass. This demonstrates that the classification of context into distinct categories can be vague and deems the work of pervasive application developers more challenging. By expanding this issue, we observe that sometimes context can be retrieved recursively and pervasive middleware or applications should embed fallback mechanisms in order to calculate critical context from alternative sources. Obviously, this may need supplementary processing resources due to higher computing complexity.

For the provision of accurate services, the quality of retrieved context is critical. Enhancing context acquisition and delivery can make interaction frameworks operate more efficiently and offer better balance between pro-activity and transparency. It is unavoidable that some context sources may reside in remote locations, whereas some others are local. Storing frequently updated information can pose a performance issue for the client application, as well as the service provider. Due to the vast volume of exchanged data, an architecture that supports distributed processing is essential. Current implementations offer solutions ranging from text file and XML-based logging to more complicated distributed database implementations. The underlying architecture is of the utmost importance (i.e. performance and usability factors), when the ultimate requirement is to obtain the most recent data in order to make a prospective decision.

This is why precise context synchronisation is required. A synchronisation and normalisation entity must exist in all participating sides and its performance can directly toggle the value of service. Consequently, synchronised context has to be relevant to the spatiotemporal attributes of each subject. To improve pro-activity of the application, prediction of information is necessary (Ashbrook and Starner, 2003). Actual prediction may not be completely accurate but based on the existing local and remote information content, it can prove efficient after being elaborated by software-based agents. Therefore, the need of keeping redundant context in the application pool is a determining factor, which directs the development of distributed, mobile caching paradigms. Applying Knowledge Discovery in Databases (KDD) and Data Mining (DM) functions on remotely sensed context might extract value-added information. Yuan et al. (Yuan et al., 2004) maintain that location influences interaction patterns, while interaction patterns influence the location of entities and activities. This may prove useful for identifying structures within a user track and finding specific geographic points where behaviour changes after short-term decision-making, as observed by Ashbrook and Starner (Ashbrook and Starner, 2003). Applying these techniques to user context, could transform pervasive interfaces to *active* by altering their behaviour towards the user. Using KDD and DM techniques for geographic research is very well documented by Gahegan (Gahegan, 2003).

Functional Requirement 1

Identifier	**FR1**
Description	The CMS shall be able to acquire user data
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if it can communicate information to the users through the bound user interface, in real time. The users should explicitly provide such information in order to enable the use of further services or internal management. Acquisition of user information should be available at any time, without restricting system operation in any way. User information is divided into the following categories. • Personal information (e.g. name and age) • Contact information (e.g. email and address) • Current activity (e.g. travelling) • Calendar information (e.g. meeting dates and time) • Professional skills • Entertainment or Communication preferences This requirement is met if minimal system functionality can be supported without the addition of any user data. If a user provides such data, the CMS should be in place to accept and make use of it for the defined purpose.

Functional Requirement 2

Identifier	**FR2**
Description	The CMS shall be able to acquire user-generated spatial data
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if it can communicate spatial information with the users through the bound user interface, in real time. The users should explicitly provide such information in order to describe virtual or real elements of the environment. Acquisition of spatial information should be available at any time, without restricting system operation in any way. User-generated spatial information is divided into the following categories. • Location of POIs • Location of people nearby or friends This requirement is met if minimal system functionality can be supported without the addition of any supplementary spatial data. If a user provides such data, the CMS should be capable of accepting and making use of it for the defined purpose.

Functional Requirement 3

Identifier	**FR3**
Description	The CMS shall sense location data
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if a valid connection has been established between the CMS and a position determination sensor (DR9), and a continuous stream of data fuses the update of the current user location. Sensing location data should occur in real time for the duration defined by the user.

Functional Requirement 4

Identifier	**FR4**
Description	The CMS shall sense orientation data
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if a valid connection has been established between the CMS and an orientation sensor (DR10), and a continuous stream of data fuses the update of the current user orientation. Sensing orientation data should occur in real time for the duration defined by the user.

Functional Requirement 5

Identifier	**FR5**
Description	The CMS shall be able to share user information with remote entities
Type	Functional Requirement

Fit Criterion	The framework satisfies the requirement if a remote entity can discover the identity of the local user by examining the records obtained through the framework transaction, or vice versa. User-related information that may be exchanged is defined by FR1 and the availability to the remote party is bound to the security and privacy restrictions defined explicitly by the local user. The format of this communication pattern is described in OR3. This requirement is met if a local user wants to transmit individual information about himself to a remote party, the communication between the two parties is established successfully and the remote party disassemblies successfully the data into the same data structures that have been utilised by the source.

The results of the main part of the *Requirements Acquisition Survey* (i.e. Question 1) revealed that users need specific motivation to release the information described in requirement FR1. The sink of information can be either the system itself, which may form a personalisation profile and direct potential services through that or a remote user, who wants to exchange such information so that further interactions may take place. For the purpose of this research we divided such motivations in two groups, work and social-related motivations. The following table describes the motivations required by the survey participants to submit each data type.

Motivation	**User Information**					
	Personal	**Contact**	**Activity**	**Calendar**	**Professional**	**Entertainment**
None	20.7%	20.7%	41.4%	27.6%	13.8%	20.7%
Work	17.2%	41.4%	10.3%	37.9%	58.6%	6.9%
Social	10.3%	0.0%	17.2%	0.0%	0.0%	44.8%
Both	51.7%	37.9%	31.0%	34.5%	27.6%	27.6%
Total	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

Table 4-1: Motivation required by users in order to share sensitive data

Functional Requirement 6

Identifier	**FR6**
Description	The CMS shall be able to share location information with remote entities
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if the local user can share information about the location of selected points or object of interest (OR5) and their descriptions (OR4) with remote entities by examining the records obtained through the framework transaction. Location

	information that may be exchanged is defined by FR2 and the availability to the remote party is bound to the security and privacy restrictions defined explicitly by the local user. This requirement is met if a user (i.e. local or remote) wants to transmit location information to a remote party, the communication between the two parties is established successfully and the receiving party disassemblies correctly the data into the same data structures that have been utilised by the source.

Users need certain incentives to release information about their location or about the location of objects and people that they know. The following table describes the results of Question 1, which are related to the potential release of spatial information, obtained from the participants of the *Requirements Acquisition Survey*.

Motivation	**Location Information**		
	User (latest)	**POIs**	**Other People**
None	24.1%	31.0%	34.5%
Work	34.5%	6.9%	3.4%
Social	10.3%	37.9%	34.5%
Both	31.0%	24.1%	27.6%
Total	100.0%	100.0%	100.0%

Table 4-2: Motivation required by users in order to share location information

Functional Requirement 7

Identifier	**FR7**
Description	The CMS shall be able to exchange messages between remote users
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if its users can exchange text messages between them. Instead of verbal communication, users may communicate by using the instant messaging feature. This is useful to share additional information or request user-related and location information from remote entities. The format of this communication pattern is described in OR3. This requirement is met if the framework can support real-time text message handling after establishing network communication and conforming to the privacy and security rules posed by each party.

Established communication and information exchange between two parties needs to be governed by explicit privacy rules. This happens because certain information, in conjunction with a mobile device and an application may characterise a specific user. The identity of the person on the other end of the communication channel needs to be acquired before submitting any information to them. That is why *Aura*'s operational model was designed in a way to allow the protection of its users from potential fraudulent transactions. This motivation has also been verified as a required feature for any similar application by the results of the conducted *Requirements Acquisition Survey*. In Questions 2 and 3 of the main part of the survey questionnaire, the participants provided sweeping replies about their consideration of the security and privacy mechanisms available in an application, before using it to exchange work or social-related information, respectively. The results are presented in the following bar charts. Additionally, every system that aims for end-user adoption must form and promote the element of trust between the potential user and the application itself.

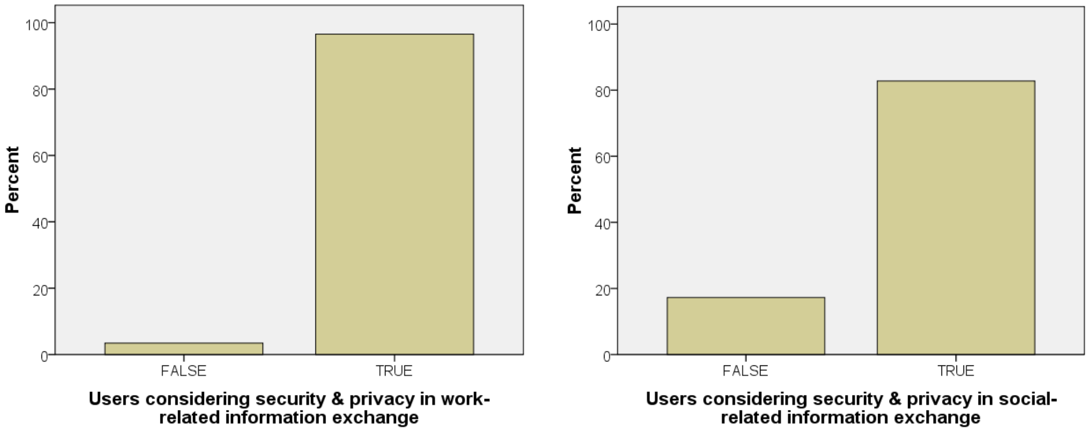


Figure 4-5: Users considering security and privacy specifications of an application before exchanging a) work-related and b) social-related information

Functional Requirement 8

Identifier	**FR8**
Description	The CMS shall respect user's privacy rules for information exchange
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement when the release of information to a remote party does not take place without explicit user approval. Such information is either defined by the user profile or any position track logs, currently or previously acquired. This requirement is met if, after establishing network communication and before the actual data exchange, the CMS entity validates the communication channel by querying the user-specified rules. If the profile and location security parameters are enabled then the data is ready for release. If any piece of information is not explicitly authorised, then the specified communication channel is discarded.

Functional Requirement 9

Identifier	**FR9**
Description	The CMS shall store a minimum local user profile
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if it can store a minimum user profile, which can be queried when the framework is executed. This profile should keep the latest settings applied by the user of the system. Its purpose is to provide initial functionality for the system to operate, without making the user apply the same settings to every execution. This profile should be composed out of the following information. • Name, Surname or Username • Age • Privacy Settings o Profile distribution availability o Location distribution availability

After examining the results of the survey on a usability question regarding the way that users preferred to input their personal preferences, we found out that they matched the results that govern mobile interactions in general. In more detail, mobile users spend less time inputting data but they do it more frequently, in contrast to non-mobile users. The vast majority of participants wanted to type their profile once and forget about it. Following next, in terms of proportion size, come the participants who do not want to form a global user profile, but prefer to answer questions every time the application requires new data. Only a small minority of users wanted to update their profile every time the application got executed. The following diagram represents the replies that were retrieved during the survey analysis from Question 20.

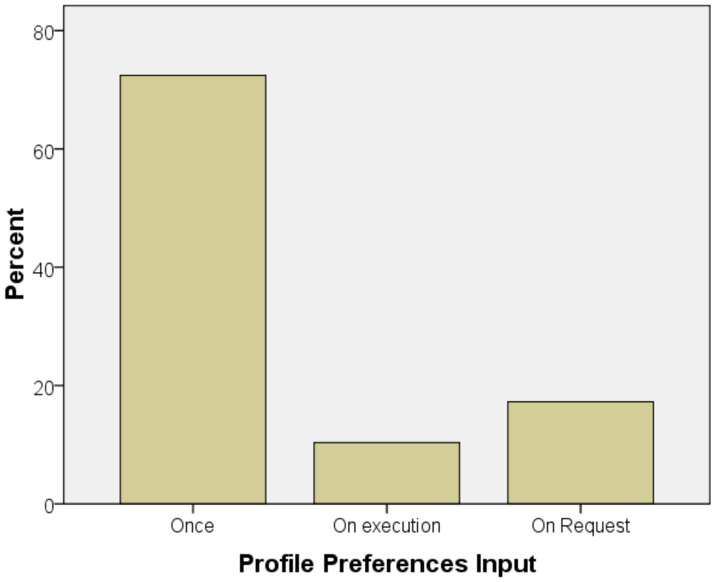


Figure 4-6: Profile creation preference

Functional Requirement 10

Identifier	**FR10**
Description	The CMS shall be able to store spatial information
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if the underlying software infrastructure is in place, in order to save spatial information when the user requires that functionality. Such information is defined as user position track logs or position of waypoints, objects and remote people. The persistent storage mechanism should query services on the content and it should conform to one of the following techniques. • Database • XML-based log files (e.g. GPX) • Text-based log files Each technique has special requirements but the most important difference between database and log files is that in the database the metadata need to be defined during application installation, whereas log file creation can be accomplished fully automatically by the framework. That is why we define the core data structure, which describes all spatial measurements to be stored. • Latitude • Longitude • Elevation • Datum • Grid • Measurement Time • Magnetic Variation (i.e. Orientation) • Textual Description (e.g. Name) This requirement is met if the framework can store a series of real-world locations, which have been visited by the user, or the position of POIs in real time. That information should be available for reuse at any time during system execution.

Functional Requirement 11

Identifier	**FR11**
Description	The CMS shall be able to derive orientation data
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if it can offer orientation information without the existence of a digital compass, in real time. This is required because employing a dedicated sensor may not always be possible or may be prohibited by external conditions. For a specific span of time, orientation can be measured by calculating former and present GPS location coordinates. Even if DR10 is not met, by implementing additional algorithms and by exploiting DR9, results roughly similar to those found in FR4 can be produced. This requirement is met if redundant context management techniques are

	utilised in order to support retrieval from secondary available sources as a fail-safe mechanism.

Functional Requirement 12

Identifier	**FR12**
Description	The CMS shall be able to derive temporal data
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if it can retrieve the current date and time and synchronise further interactions based on the new temporal parameters. There are several options about how this process can be accomplished. Each method is applicable if the previous is not available. The synchronisation methods are presented below, in order of preference. • Synchronise with the service provider clock o If the framework uses a custom server for exchanging other types of context, the clock of this server can be used to synchronise the client entities. • Synchronise with a specific Internet Time Server (i.e. NTP protocol) o If and when a client device establishes a connection to a network that offers Internet access, a specific time server (e.g. pool.ntp.org) can be queried for up-to-date information. • Synchronise with the clock of the device o If network connectivity is not available, the framework can synchronise with the clock of the device. This is the least preferred option because the device may not always hold up-to-date information. Clock synchronisation can take place either at the beginning of execution and/or at specific time intervals. This requirement is met if the framework clock is always synchronised and produces accurate measurements when necessary.

Performance Requirement 8

Identifier	**PR8**
Description	The CMS shall filter out-of-bound measurements
Type	Performance Requirement
Fit Criterion	The framework satisfies the requirement if it can identify and discard any measurements that are not precise or are considered unwanted for optimum system operation. There is a number of such cases, which the CMS must recognise and adapt its functionality. For instance, major differences between GPS coordinates, in very short intervals (i.e. 1 to 2 seconds), indicate that these measurements are not valid and should be discarded. In contrast, when the CMS is logging the user track, very small position adjustments (i.e. ≤10cm) may extremely increase the track log and pose negative implications due to storage requirements.

	This requirement is met if the framework stores all measurements for a certain time span and then compares them with the new ones, in order to discover and reject those that are found to be redundant.

4.2.2 Visualisation Requirements

The pattern that governs human-computer interactions on mobile devices follows a very deterministic path. The reason is that the interaction interface and the visualisation interface are tightly coupled to form a unique whole. Because of the special characteristics that rule client-side pervasive applications, the requirements are different when inputting and when presenting information.

Information visualisation on pervasive systems functionally depends on the task in progress and the management of acquired contextual information. By having different visualisation mechanisms that can simulate the users' spatiotemporal environment and social relations, immersion in the alternate environment becomes smoother and more effective. Immersion in a computer-generated environment is an important factor that needs special handling because just an abstract representation of the surroundings is not adequate. The user needs to maintain a complete cognitive map before acting, based on the appropriate feedback from the device. To match the frames of reference of the cognitive environment with the registration parameters of the user in the represented scene, the most recent and valid contextual information is required from the sensors. The process of self-localisation in an unknown place can be enhanced by the existence of natural landmarks (incl. urban constructions) and the various perspectives that are supported by the user interface (Liarokapis et al., 2006a). Perspectives ranging from egocentric to allocentric and computer vision see-through displays are capable of providing discreet solution that simulates and completes the cognitive map of a person.

By employing the aforementioned methods on a client-side service tier, we observe that traditional two-dimensional approaches pose drawbacks and the employment of specific VR characteristics are more suitable for ubiquitous presence and interaction. Currently, the first disadvantage of VR, which Weiser discussed, has been surpassed. GIS, geo-visualisation and computer graphics have produced the means to create 3D worlds of adequate verisimilitude and in a cost-sensible manner.

Figure 4-7: An Allocentric Perspective in Aura's VR interface

Figure 4-8: The Egocentric Perspective in Aura's VR interface

Research into pervasive systems operation has shown that the user needs to be minimally distracted from the current activity. In cases where user intervention is necessary, though, the pervasive system has to instruct rapid augmentation of the visualisation interface, in order to match the user's state. On the other hand, due to the large volume of information that may be processed, exactly the opposite result may occur. In terms of Information Science, we refer to this outcome as *information overload*. Controlled overlaying of objects (e.g. text or images) in a virtual world, allows more information to fit in the display real estate. Adapting the context-sensitive interface to a size that can fit on a personal mobile device can produce the necessary difference that crosses the usability threshold, as mentioned by Want and Pering (Want and Pering, 2005).

Adaptation to user, device, environment, temporal and other context adjustments has to be immediately reproduced on the interface of the service. Henricksen et al. argue that users must have a uniform mental model of the executing application, regardless of the type of interface that is presented to them. This is particularly complicated when the interaction paradigms are based on diverse input methodologies (Henricksen et al., 2001). Adapting to context variables does not have to make the interface more complicated due to the risk of concealing part of the advanced functionality. User context and personalisation features can customise the interface of the application according to predefined values, when specific events are triggered. Thus, the interface may operate unobtrusively, called or discarded only when specific decision points have been reached. Additionally, adaptation to physical world context is equally important. The actor has to relate with the environment and the objects that possess contextual information. Currently, several mobile devices are available that have embedded sensors for acquiring context, such as density of light, as well as other services such as traffic (TMC). These e-services have just started to be commercially explored in order to provide benefits for their customers.

Functional Requirement 13

Identifier	**FR13**
Description	The IPS shall offer egocentric perspectives of the environment
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if the physical environment, which surrounds the user, including selected objects or features of interest that exist in it, can be represented on the mobile device screen by adopting an egocentric viewpoint. Egocentric is defined as the perspective that describes the scene through the eyes of the beholder. The source of spatial information could be either generated in real time or loaded from persistent storage. Both ways should be transparent to the user, without the need to make complex decisions. The most efficient visualisation interfaces that are capable of supporting this

	requirement are Augmented Reality and Virtual Reality. This requirement is met if the represented, on the device display, scene matches the field of view of the local user.

Functional Requirement 14

Identifier	**FR14**
Description	The IPS shall offer allocentric oblique perspectives of the environment
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if the physical environment, which surrounds the user, including selected objects or features of interest that exist in it, can be represented on the mobile device screen by adopting an oblique viewpoint. Allocentric oblique is defined as the perspective that describes the scene by raising the viewpoint and producing a small inclination (e.g 45°) towards the ground. The source of spatial information could be either generated in real time or loaded from persistent storage. Both ways should be transparent to the user, without the need to make complex decisions. The most efficient visualisation interface that is capable of supporting this requirement is Virtual Reality. This requirement is met if the represented, on the device display, scene can be simulated by an elevated camera, which partially observes and illustrates the local user and the surrounding environmental features.

Functional Requirement 15

Identifier	**FR15**
Description	The IPS shall offer allocentric plan views of the environment
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if the physical environment, which surrounds the user, including selected objects or features of interest that exist in it, can be represented on the mobile device screen by adopting an allocentric viewpoint. Allocentric plan view is defined as the perspective that describes the scene by raising the viewpoint and producing a vertical inclination towards the ground. The source of spatial information could be either generated in real time or loaded from persistent storage. Both ways should be transparent to the user, without the need to make complex decisions. The most efficient visualisation interfaces that are capable of supporting this requirement are Virtual Reality and map-like illustrations. This requirement is met if the represented, on the device display, scene can be simulated by an elevated camera, which simulates a bird's eye field of view and illustrates the local user at the centre of the scene.

The only interface that can support all aforementioned user perspectives of the environment is Virtual Reality. The reason is that VR simulates an environment comparable to the real, which users may observe by adopting their preferred viewpoint. But as the results of the user survey depicted, every viewpoint does not equally assist the users in their current task. Question 10 of the main questionnaire tried to identify which perspective the participants would prefer while being involved in a naïve or primed search, as well as an exploration task in a virtual environment analogous to the real. Because a large proportion of the participants did not have extended experience with real-time first person (AR) and allocentric oblique (VR) visualisation engines, the allocentric plan view was found the most favoured viewpoint, as it resembles the widely recognised map representations. The results for each wayfinding task are presented in the following charts.

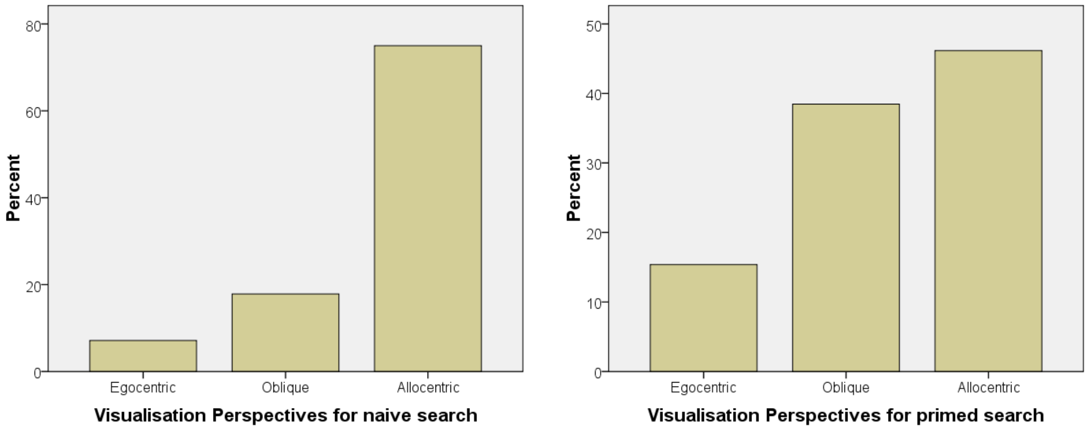


Figure 4-9: Preferred visualisation perspective for a) naïve and b) primed search tasks

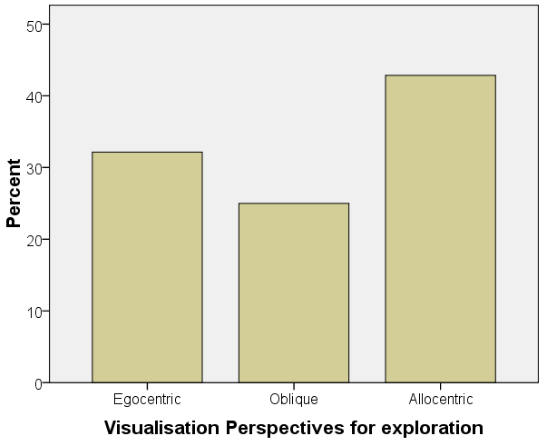


Figure 4-10: Preferred visualisation perspective for exploration tasks

In addition, Question 11 of the main part of the *Requirements Acquisition Survey* tried to explore if the transportation means that is being used and by extension the speed of travel contributes to the selection of a certain perspective in a virtual environment. Similarly, the bird's-eye perspective was found more useful in most cases for the participants, but the results have also shown that the speed of movement is a determining factor towards viewpoint selection. The egocentric perspective is found more efficient, when the actor velocity is low. Allocentric oblique and plan views have been deemed more supportive, when the speed is increasing respectively. The main reason that contributes to this phenomenon is that the abstract area described by a map requires less cognitive load from the user because it changes frequently. In contrast, when the user spends more time in a selected area, he or she needs a more detailed representation. The results of Question 11 are presented below. Our results agree with the findings and proposals presented by other researchers (Meng and Reichenbacher, 2005).

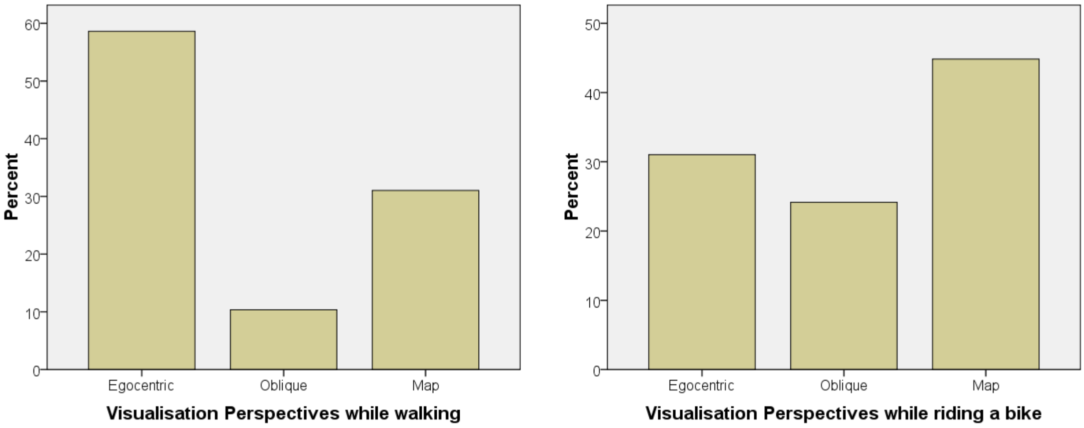


Figure 4-11: Preferred visualisation perspective while a) walking and b) riding a bike

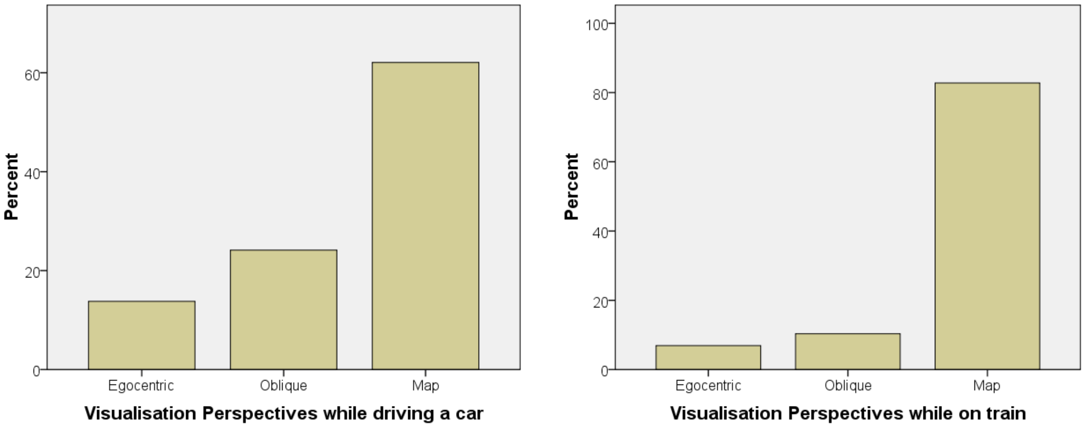


Figure 4-12: Preferred visualisation perspective while a) driving a car and b) on train

Functional Requirement 16

Identifier	**FR16**
Description	The IPS shall offer photorealistic representations of the environment
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if the rendering engine is capable of producing photorealistic images of the environment in real time. Presenting a visually realistic environment to the users contributes to their engagement and immersion, while it creates a sense of *being there* (i.e. presence). The most efficient visualisation interfaces that are capable of supporting this requirement are Augmented Reality and Virtual Reality. The source of environmental representations could

	be either generated in real time or loaded from persistent storage. In real time (AR) rendering, reconstruction of the visual feedback should be triggered by the on-board camera (DR11), whereas in the VR interface the modelled environment should be textured with recent photorealistic images, either automatically or semi-automatically, which resemble as accurately as possible the real world phenomena.

Functional Requirement 17

Identifier	**FR17**
Description	The IPS shall offer non-photorealistic representations of the environment
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if the rendering engine is capable of producing non-photorealistic images of the environment in real time. Applying Non-Photorealistic Rendering (NPR) techniques can produce an artificial world with an artistic background in cases where physical realism is not an issue. Greater abstraction engages the senses and imagination of the user to create the perception of being elsewhere (i.e. in the computer generated world). The most efficient visualisation interfaces that are capable of supporting this requirement are Augmented Reality and Virtual Reality. The source of environmental representations could be either generated in real time or loaded from persistent storage. In real-time (AR) rendering, reconstruction of the visual feedback should be triggered by the on-board camera (DR11) and the surfaces amended by custom textures. In contrast, the simulated environment (VR) could be textured, either automatically or semi-automatically or left without any textures. Some of the NPR styles, which are proposed for rendering texture images of large environments (i.e. urban), include the following. • Normal shading • Cartoon shading • Pen-and-ink with noise • Pen-and-ink without noise • Line rendering • Volume illustration

Even though photorealistic virtual environments offer greater verisimilitude to the real world, several people do not require such features when interacting with such interfaces. In several cases, it depends on the artistic preferences of each individual. In contrast, non-photorealistic virtual worlds are considerably more efficient, in terms of performance and graphic requirements of the underlying hardware. This happens because the quality of the applied textures is a contributing factor to the rendering performance. The results that have been retrieved from the survey analysis illustrate that a large proportion of the participants was in doubt about this issue either due to their lack of experience or because other factors, relevant to personal preferences and application-specific functionality, rendered this feature optional. The results of Question 12 are presented in the following bar chart.

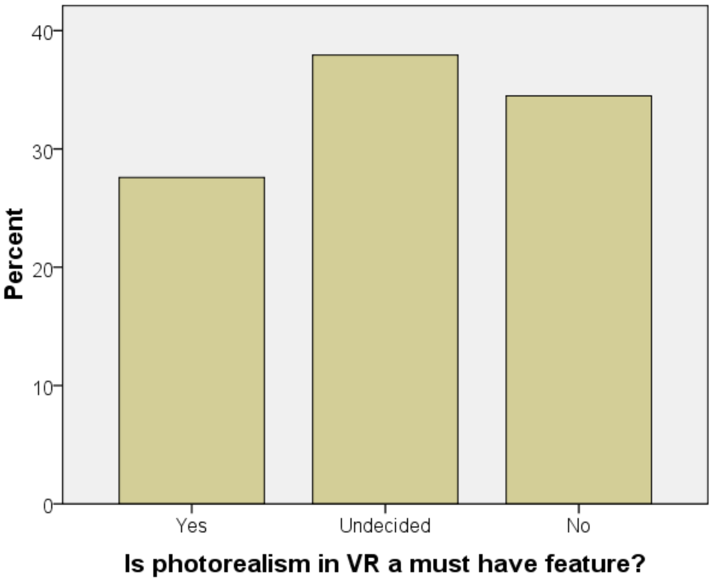


Figure 4-13: Importance of photorealism in VR

Functional Requirement 18

Identifier	**FR18**
Description	The IPS shall represent absolute spatial-context attributes on the user interfaces
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if it provides a pool of elements that can be used to identify and locate specific features of interest in the represented environment, including any relevant virtual and real objects or actors that exist in it. This requirement is applicable to Augmented Reality, Virtual Reality and 2D map visualisation interfaces. Each type of context variables should be represented by a distinct 2D symbol, 3D element and/or descriptive text annotations, which identify as accurately as possible the corresponding object type. The on-screen position of every annotation should be accurately located on the exact coordinates that each object occupies. In addition, the rotation parameters of the virtual object should correspond to its real world behaviour or, if this is not possible, its primary facet should be directed towards the user. This requirement is met if the represented scene accurately characterises any element useful to the user on the bound interface. The user shall be able to explicitly control the selection and duration that such illustrations or annotations are apparent on the interfaces.

Functional Requirement 19

Identifier	**FR19**
Description	The IPS shall represent provisional spatial-context attributes on the user interfaces
Type	Functional Requirement

Fit Criterion	The framework satisfies the requirement if it provides a pool of elements that can be used to identify and locate specific points of interest in the represented environment. An example of such context can be considered a series of GPS coordinates, which may form a track for the user to follow. This requirement is applicable to Augmented Reality, Virtual Reality and 2D map visualisation interfaces. These context variables should be represented by a distinct 2D symbol, 3D element and/or descriptive text annotations, which identify as accurately as possible the point or track. The on-screen position of every annotation should be accurately located on the exact coordinates that each objects occupies. In addition, the rotation parameters of the virtual object should correspond to the targeted direction. This requirement is met if the represented scene accurately characterises any useful, for the user, points on the bound interface. The user shall be able to explicitly control the selection and duration that such illustrations or annotations are apparent on the interfaces.

Usability Requirement 1

Identifier	**UR1**
Description	The IPS shall align the represented environment with the user's cognitive frames of reference
Type	Usability Requirement
Fit Criterion	The framework satisfies the requirement if it accumulates the 6 degrees of freedom that have been acquired by the sensors and forms a detailed representation of the immediate surroundings that matches the one held as a mental image by the user (i.e. cognitive map) in real time. This way, the user can rapidly access multiple frames of reference (cognitive or simulated). The environment representations must be drawn dynamically so that they match the observer's cognitive viewing scale.

Usability Requirement 2

Identifier	**UR2**
Description	The IPS shall prevent information overload
Type	Usability Requirement
Fit Criterion	The framework satisfies the requirement if it offers automatic and real-time techniques to minimise information overload on the presented output interface. Representation of context variables on the user interfaces, which comes in the form of text or graphical elements, should not obstruct the readability of other similar variables. Additionally, the selection of which variables to represent on the bound user interface can be made out of certain criteria, such as the distance from an object, its type (e.g. hospital) or importance (e.g. next objective). The visualisation interfaces that this requirement is applicable to are Augmented Reality, Virtual Reality and 2D maps. This requirement is met if controlled overlaying of non-environmental variables efficiently manages the intensity levels of the useful content that is represented for the user.

Usability Requirement 3

Identifier	**UR3**
Description	The IPS shall offer user-customisable field of view
Type	Usability Requirement
Fit Criterion	The framework satisfies the requirement if it provides the option to the user of the system to control the field of view that has been adopted while observing the virtual environment. This can be accomplished by explicitly managing the pitch angle of the horizontal plane. This requirement contributes positively to UR2, as well. The visualisation interface that this requirement is applicable to is Virtual Reality. This requirement is met if the system provides a default field of view, equivalent to the one found in the human eye (i.e. 60°) and provides an adjustable range of ±30°.

Usability Requirement 4

Identifier	**UR4**
Description	The IPS shall resize video content to fit on the interface
Type	Usability Requirement
Fit Criterion	The framework satisfies the requirement if the video stream that is generated by the on-board camera occupies the maximum available space on the device screen. This requirement is applicable to the Augmented Reality interface. Due to the differences between the camera and the screen resolutions and due to variable screen sizes, the video content may not fit exactly on the screen rectangle. There may be instances where the resolution proportions are not compatible (e.g. 4:3 vs. 2:2) with each other. In such cases, the framework must identify these incompatibilities and adjust the output, so that it occupies most of the screen. Very low camera resolutions must not exceed a specific threshold (e.g. x3) because the quality of the presented content is severely affected. This requirement is met if the framework provides the algorithms to scale down or up the video content in order to fit on the display of any compatible mobile device

4.2.3 Interaction Requirements

Interaction with the interface is the process whereby the user implicitly or explicitly acts in order to provide information to the framework. In pervasive services, the user has the option to act or not to act at all.

The second reason that made Weiser discourage the use of VR in ubiquitous systems is related to the immersion of the user in a VE by diverging from reality. However, in the case of context-aware interfaces, gaining information through effective utilisation of computing devices is a core requirement. In addition, great value would apply if the interaction lasted for the minimum amount of time and involved communication with an equally resourceful or even more elaborate environment. The answer to these requirements comes from a widely adopted product (i.e. mobile devices) and a quickly evolving technology (i.e. AR). In technical terms, AR is not a single technology but a collection of different technologies and algorithms that operate in conjunction, with the aim of enhancing the user's perception of the real world through computer-generated information (Azuma, 1997). This type of information is referred to as virtual, digital or synthetic information. Pervasive service consumers should work individually or collectively, experiment with computer-generated information and interact with a mixed environment in a natural way (Liarokapis, 2007). Natural interaction is a crucial usability requirement for pervasive interfaces. Novel types of interactions have to promote a rewarding for the user experience, in terms of information retrieval. The research has revealed that interactions with mobile devices are more frequent, but last shorter periods in comparison to static computers that are less frequent but more time consuming. This explains how the second drawback of VR (Weiser, 1993) can be surpassed - with minimal interactions, on a customised and highly immersive environment that can quickly transfer relevant and in-context information to the requesting party.

If the mobility and usability factors of current smartphones are taken into consideration, ubiquitous operation of pervasive systems comes closer to reality. Ubiquity does not relate only to location context and network availability. It includes additional variables that formulate complete understanding of the user's natural presence. Gesture-based interactions can be triggered by detecting changes on the orientation attributes. Advanced functionality can be triggered through that context source. In more detail, if the sensor (e.g. digital compass) is attached on the subject, it can reveal current user context (i.e. orientation) and may be used as a navigation aid, whereas if attached on the mobile device it can become an interaction means for the pervasive interface. Both ways are useful, depending on the functionality that is required from the service. Interaction interfaces residing on the client side of a pervasive application must provide means of communication for people with diverse background and habits. Once again, we refer to the example of current smartphones because they employ multiple standardised interfaces and their functionality is improved when adapting to real-time context. Common implementations offer a touch screen, a keyboard and a multi-direction navigation button. If naturally caused interactions produced by changes in position and orientation are also appended to that list, this results in five explicit or implicit ways to work with the interaction interface of the service. Thus, no assumption is made by the service about the preferred input method, because all of them may produce equivalent results.

Functional Requirement 20

Identifier	**FR20**
Description	The IPS shall be able to operate in sensor-controlled mode
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if user movement and interactions in the real world can be simulated in the virtual environment by accumulating sensor data. According to sensor availability, various interactions may be represented in the digitised environment. Altering the interface characteristics through manual input methods should be avoided, although this should not pose any negative operational issues. A set of predefined perspectives (i.e. egocentric, allocentric oblique and plan view) can be explicitly selected by the actor, according to personal preferences. The combination of the sensor-controlled mode with the adoption of customisable user viewpoints should be able to reveal as much information about the immediate surroundings and any relevant features as possible. The visualisation interfaces that this requirement is applicable to are Augmented Reality, Virtual Reality and 2D maps. This requirement is met if the interface provides the mechanism for the user to select this mode of operation. When it is enabled, continuous querying of sensor data should take place, which after interpretation should trigger virtual world interactions.

Functional Requirement 21

Identifier	**FR21**
Description	The IPS shall be able to operate in user-configurable mode
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if its users have explicit control of their movement and interactions in the digitised environment. No sensor data should be required to direct the user while exploring the virtual world. Exploration should take place manually by utilising the interface and mobile device input mechanisms. Additionally, any location of the synthetic environment should be available for examination. Multiple, user-acquired perspectives need to be supported, which offer the best visualisation, based on current user needs. Every interaction based on 6-DOF should be accomplished by utilising the device and interface input methods. The visualisation interfaces that this requirement is applicable to are Virtual Reality and 2D maps. This requirement is met if the interface provides the mechanism for the user to select this mode of operation. When it is enabled, the user should be free to manually visit any location of the

	virtual world, perform user-centred actions and retrieve additional information about remote objects.

Functional Requirement 22

Identifier	**FR22**
Description	The IPS shall be able to represent and simulate occupied user trajectories
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if it provides the option to the user to visualise, in sequence or not, any previously visited POIs. These POIs can be either collected by the local user (i.e. GPS tracks) or by any other user. The coordinates of these points should be located in the persistence storage of the device. The visualisation interfaces on which the trajectories can be represented are Virtual Reality and 2D maps. This requirement is met if the users can visualise their previous trajectories or any other spatial trajectory by selecting the appropriate option on the bound user interface and choosing the required source of information.

In real-world scenarios, several navigational aids may assist the users to reach their goal or help them locate an object or person. Several PND and LBS manufacturers have introduced their own aids with the aim of efficiently directing users. During the evaluation phases of the project, we discovered that not all navigation aids actually support user decisions. This happens because the elements that are employed and the time during which they are visible on the UI may confuse rather than assist the users. In order to discover which navigational aids are found useful for the participants, we introduced Question 15 in the survey's main questionnaire. The results of this question are presented below. By examining these results we identified the most valuable navigational aids for the participants.

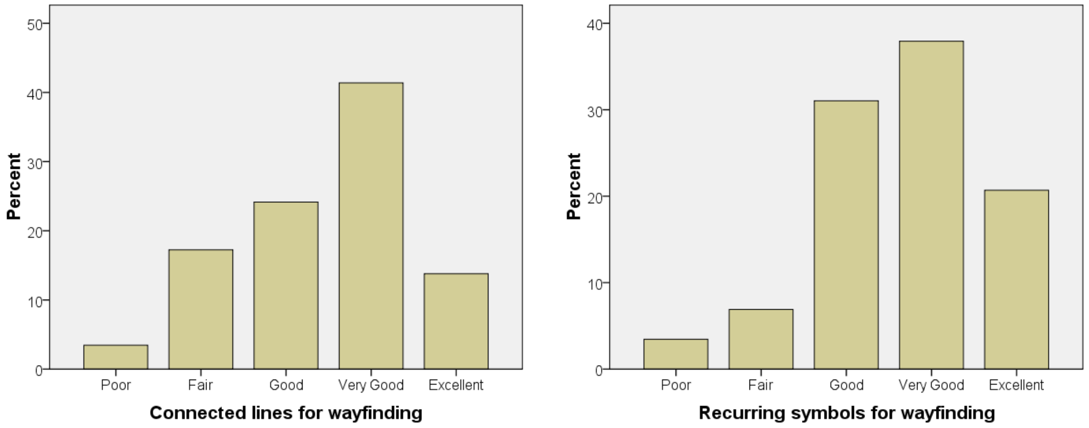


Figure 4-14: Effectiveness of a) connected lines and b) recurring navigational symbols overlaid on route

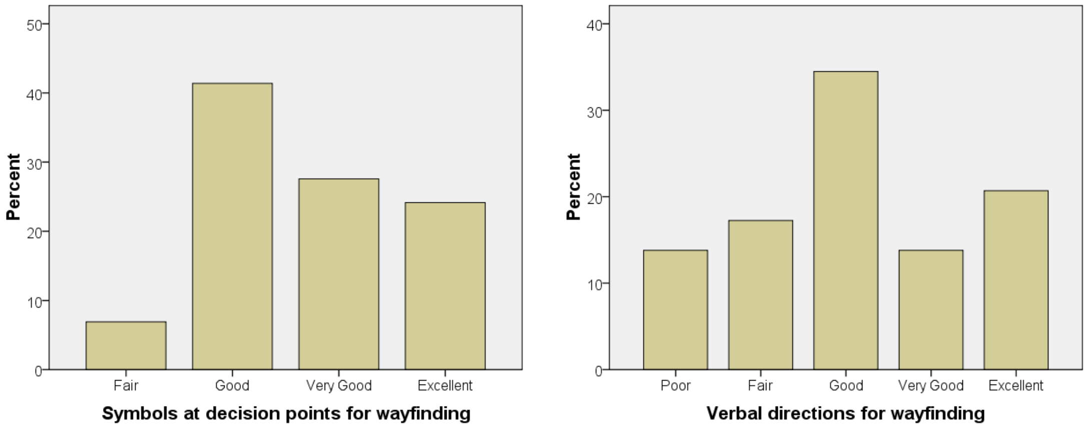


Figure 4-15: Effectiveness of a) signs placed at decision points and b) verbal instructions

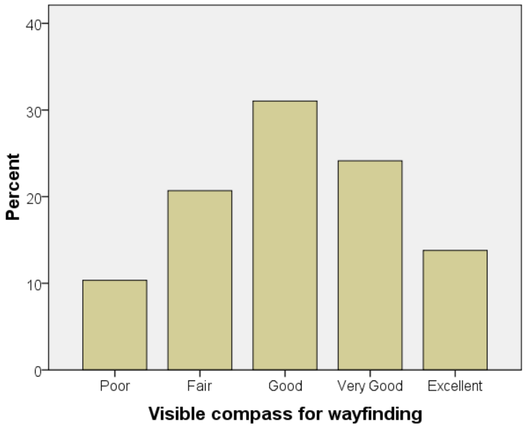


Figure 4-16: Effectiveness of a visible compass

Functional Requirement 23

Identifier	**FR23**
Description	The IPS shall offer gesture interactions
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if it can utilise the underlying hardware components (e.g. accelerometer) in order to provide advanced interactivity between the user and the environment. These interactions may trigger functionality in various layers of the framework. Specific functionality varies according to the bound user interface. In AR, the users may interrogate the physical scene by panning the device around them. VR offers more functionality for gesture interactions because it can accommodate more user perspectives. For instance, when the device points towards the ground an allocentric perspective could be presented. In contrast, when the device assumes vertical position, the perspective can be altered in a way that the environment is visualised from an egocentric viewpoint. Furthermore, the application can translate gestures to specific user actions in the simulated environment. This requirement is met if existing manual interactions (e.g. pressing a button) can be replaced by hand gestures performed by the user.

Hand gestures are a very promising feature and may enhance the functionality of several applications. Usability can be improved but there is no standardised framework on how to utilise such a feature. Specialised hardware such as 3D gloves and HMDs have only been used in limited research or industrial applications. The introduction of accelerometers and gyroscopes, besides digital compasses, allows the expansion of this feature to more applications that target end-users. During the development of our framework we found that there are several opportunities for this feature to be embedded in the existing functionality. That is why we wanted to question the participants of the survey about their preferences on this issue, especially how to translate a potential roll of the device. The results of Question 16 are presented in the following bar chart.

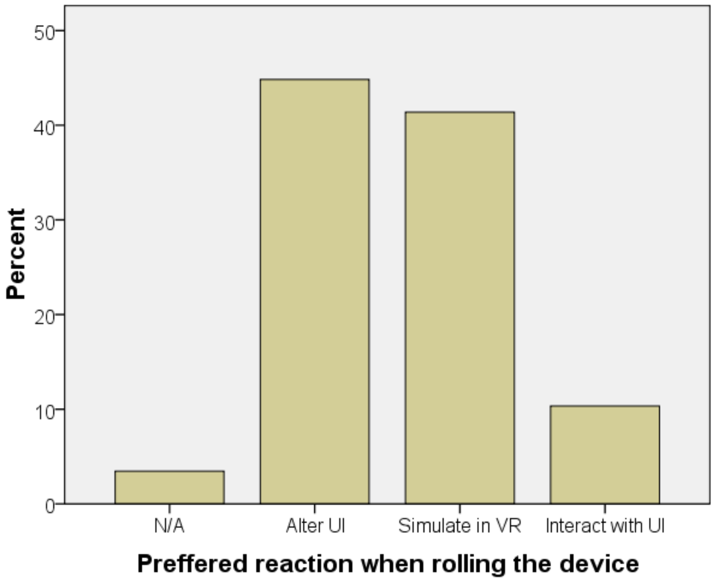


Figure 4-17: Preferred interaction while rolling the mobile device

As we have seen in a previous section, modern mobile devices embed several standardised mechanisms so that the users can interact with the application and the available interface options. Apart from the common, numerical buttons, these mechanisms include a touch screen, a 4-way navigation button and the sensors, which can offer real-time context reactivity. Furthermore, when the framework operates in manual mode, the user may need to explore the virtual environment. Thus, a decision needs to be made about which interaction mechanism to implement in order to support such a functionality. The preferred mechanism should satisfy the majority of users but an alternative should also be available. The participant’s preferences, which have been documented by the *Requirements Acquisition Survey* analysis and more precisely by Question 19, are presented in the following figure.

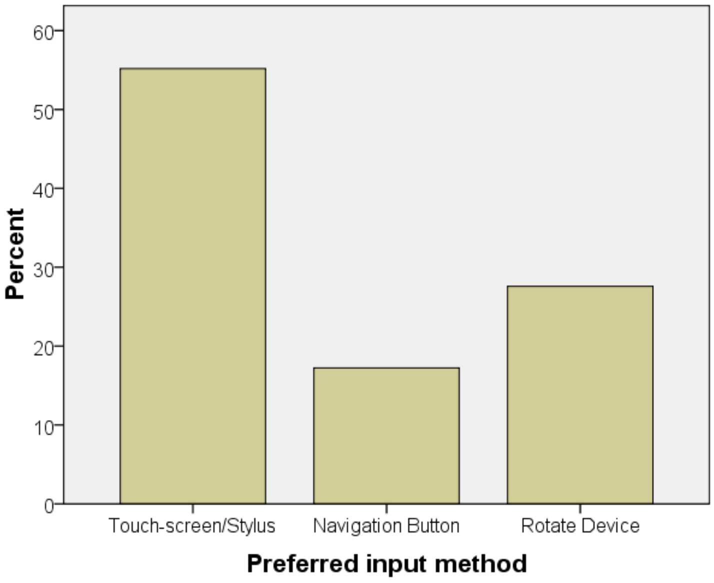


Figure 4-18: Preferred input method for VR interface

Functional Requirement 24

Identifier	**FR24**
Description	The IPS shall be able to transform between real-world coordinates, screen coordinates and virtual-world coordinates
Type	Functional Requirement
Fit Criterion	The framework satisfies the requirement if it is capable of translating any coordinates that exist in the implemented geometric location model. Translation should take place between real world, virtual world and screen coordinates or, alternatively, between world, camera and image coordinates. The direction of translation depends on the required functionality. Three-dimensional coordinate transformations are applicable to the Augmented Reality and Virtual Reality interfaces, whereas 2D transformations are applicable to the map-like interface. If specific camera attributes (e.g. lens distortion) are known, they should be catered for. Furthermore, the screen coordinates should be translated into pixels. This requirement is met if the IPS can make the relation between two or three coordinate systems, so that the user can visualise or interact with the objects of the environment in an accurate and straightforward way.

4.3 Proposed System Design

The core structure of *Aura* is formed out of two principal entities (Papakonstantinou and Brujic-Okretic, 2009a). Their primary functionality involves receiving input from the external sources and offering relevant information to the users with the aim of assisting in the completion of the task that they are currently involved in. These subsystems are capable of rapidly controlling the information flow between each other, as well as providing interaction (i.e. input and output) services. The first entity is the Context Management System (CMS) and the second one is the Information Presentation System (IPS), as indicated by the following figure. Apart from the core system, which is composed of the CMS and IPS entities, the user is also considered as an active source of context. This occurs because potential interactions are interpreted and influenced by the actual situation that the user is experiencing, and his or hers behaviour.

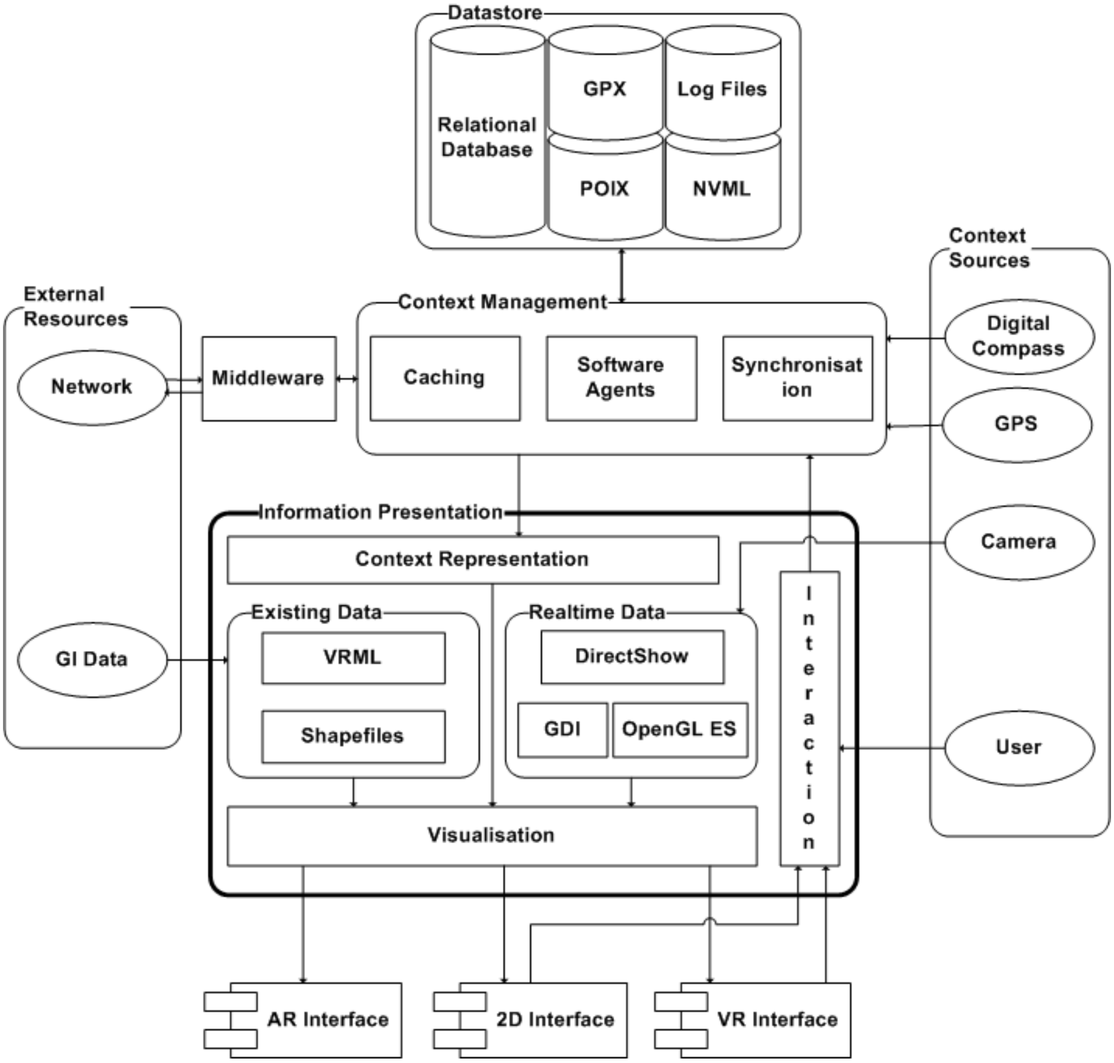


Figure 4-19: The design blueprint of the prototype platform

4.3.1 Context Management System (CMS)

The CMS unit is a low-level subsystem, which receives input from the peripheral resources and is responsible for keeping the location model, which governs the application, updated with relevant information, either from local or remote entities (Papakonstantinou and Brujic-Okretic, 2009a). There are a number of autonomous, adjacent entities that push data in continuously or randomly. The data sources can be software or hardware based. The substantial existence of these data sources signifies them as independent subsystems. Input is considered the real-time data generated by the sensors or network updates and any relevant, locally stored information. Furthermore, the CMS is responsible for accepting and processing certain user-generated actions, after they have been interpreted into contextual information by the IPS entity. This subsystem is also responsible for implementing all geometric transformations in order to keep the location model coherent and to exchange information with the available remote entities. Additionally, movement of non-natural objects (e.g. Non-Player Characters) that may exist in the scope of specialised applications is accommodated by this structure. The synchronisation of the available information generated by any source is a responsibility of the CMS. During configuration and debugging, this entity generated and simulated artificial movement of remote objects. In this subsystem, algorithms, which enhance the accuracy, performance, prediction and interpretation requirements of the application, are executed. Furthermore, the CMS is the last entity, which checks for conformance with the user-specified privacy rules before materialising any exchange of information between remote parties.

4.3.2 Information Presentation System (IPS)

Next in sequence comes the IPS unit, which is another co-operative subsystem, used for reflecting any changes caused by the CMS and for accepting explicit user input (Papakonstantinou and Brujic-Okretic, 2009a). Its main functionality is to visualise and interact with the environment and any relevant objects that exist in it. This subsystem manages input from three sources in order to accomplish accurate visualisation functions on every interface (i.e. 2D, VR and AR). Two of them are used to simulate the real surroundings and the other one aims to interconnect objects from the real world with objects in the virtual world. The latter is an interface to the CMS, which receives numerical and textual descriptions (i.e. metadata) of objects and creates their virtual representations. These can be interrogated through all user interfaces and can vary depending on the type of interrogated entity (e.g. avatar for human in 3D or planar icon in 2D). For the 2D and VR interfaces, the environment is modelled out of existing data. The supported formats are either shapefiles (.shp) for 2D, or VRML (.wrl) models for 3D elements, both generated out of the same spatial dataset, by using ESRI's ArcGIS software suite (ESRI, 2011). In contrast, the AR interface processes real-time data from the camera. The level of interaction is meant to be controlled by the user, to enable different levels of familiarity with such systems to be catered for, as discussed by Liarokapis et al. (Liarokapis et al., 2006b). The level of immersion should also be user-selectable for the reasons described by Koda et al. (Koda et al., 2005). More information about available interactions can be found in the preceding sections of the report.

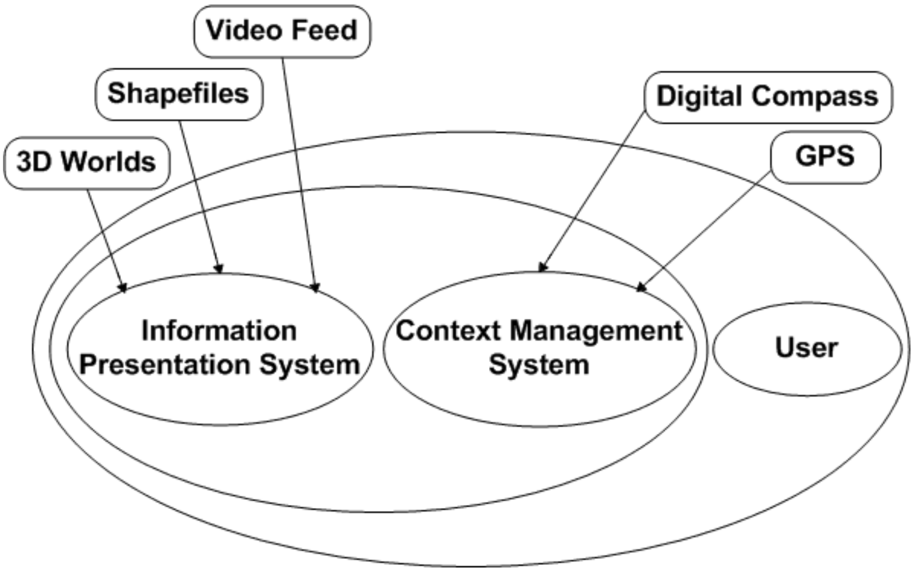


Figure 4-20: Simple Context Diagram of all adjacent systems

4.3.3 Supported Interface Paradigms

Visualisation and interaction in mobile Mixed Reality applications, which offer ubiquitous operation and environmental representation, functionally depend on the precise registration of the subject (i.e. user, sensors and mobile device) on the available interfaces, in relation to real-world conditions (Papakonstantinou and Brujic-Okretic, 2009a). The described framework analogously registers the user on the map, VR and AR interface by examining 6 degrees of freedom, three of them generated by the GPS and the rest from the compass. Reactivity of the application is triggered after detecting changes on retrieved context. Presenting information about objects in the real world takes place by querying the application pool for location information (i.e. longitude, latitude, sea-level height, type and description). Demonstrating synthetic information on each interface can occur by comparing context from the viewpoint of the subject and the remote resource. As a result, descriptive information can be visualised in relation to the actor and the real-world elements. The benefit of this methodology is that it can simulate and enhance the cognitive map of a person. This is accomplished by matching the user's cognitive frames of reference with the registration parameters of the represented scene (Klatzky, 1998). Having multiple, concurrent perspectives of the real world is invaluable for functional pervasive entertainment applications. Additionally, binding physical interactions with representations in advanced user interfaces encourages the transition from traditional applications to pervasive applications (Roecker et al., 2007). Each interface has its own special characteristics and offers certain advantages, in terms of user functionalities, over the rest.

4.3.4 Modes of Operation

Fundamentally, interaction in the 2D and 3D environments is supported by two operational modes: the *sensor-controlled* and the *user-configurable* mode (Burigat and Chittaro, 2005). This way, the system has the potential to meet a variety of user needs, such as naïve search, primed search and exploration (Darken, 1995). These modes of operation provide the means to form a multiple level-of-immersion application - visible from absolute egocentric to any allocentric perspective. In the sensor-controlled mode, interaction takes place by considering context input and placing the user in the appropriate position, with analogous orientation that corresponds to natural behaviour. This mode is designed for simulating real-time interactions, whereas the manual mode assists in the exploration of remote locations and the enhancement of the decision-making process (Papakonstantinou and Brujic-Okretic, 2009a). The user triggers manual interactions explicitly. This way, any place in the virtual world can be rapidly examined and its surroundings evaluated. In the user-configurable mode, any observation viewpoint is supported. In contrast, the sensor-controlled mode supports first-person, oblique and bird-eye views of the scene. Each perspective complements the other and it is up to the user to select the one that he or she is more familiar and comfortable with. Additional elements that could enhance the user experience depend on user personalisation preferences and involve further technical issues. For instance, the user perspective and the current orientation may be identical to the natural surrounding scene, or may vary, depending on whether the user is interested in some remote site features. In terms of pervasive operation, sensor and manual control of the interfaces can reflect the active and passive context reactivity of the system (Chen and Kotz, 2000). The following figure presents the menu item (i.e. Go Live) in *Aura's* VR interface which is used to switch between the sensor-controlled and the manual mode of operation.

Figure 4-21: The Context Menu in the VR Interface

The following Use Case diagram illustrates the system-level functions that can be performed for assisting the local user when engaged in any type of wayfinding scenario. Due to the design of the system architecture, a unifying solution can be applied for every available user interface. The Use Case diagram could be extended to other operational scenarios by replacing the three top-level goals with those required by the pervasive alternative.

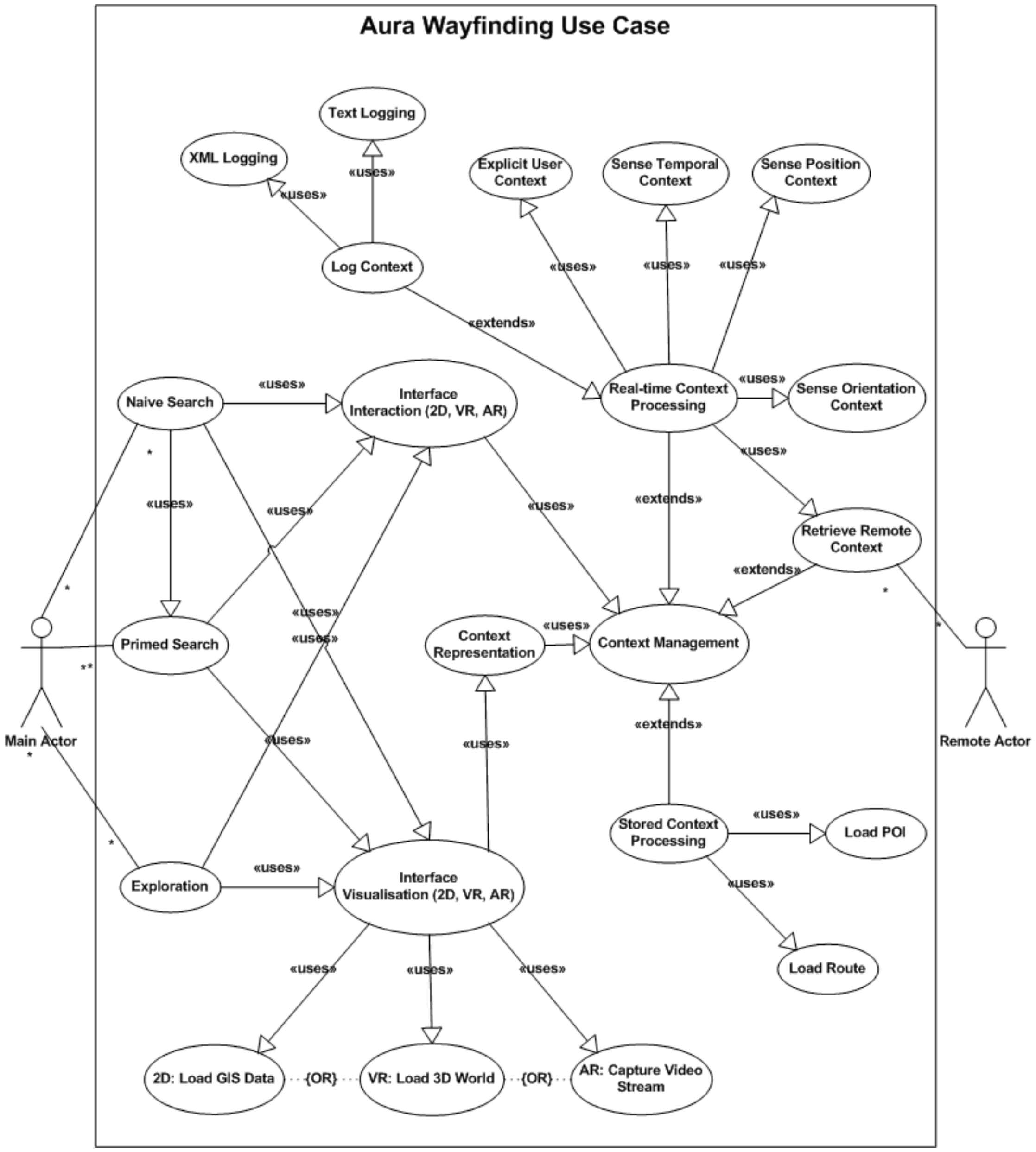


Figure 4-22: Use Case diagram for wayfinding in Aura

4.3.5 Applied System-Engineering Techniques

During the implementation phases of the client-side system, we have extensively employed software design patterns. The range spans from simple (e.g. strategy pattern for UI selection) to more complicated (e.g. observer pattern for context changes) patterns. Following this approach, we have been able to schematically describe recurring issues in the system development process. Knowledge gained from a single problem has been applied to other problems, whose solutions have been pursued with growing confidence through the use of these proven methods. Additionally, a common model for system extensibility has been established, which enhances further scalability of the system. Using patterns helps to reduce coupling between system elements and supports better design quality through flexible programming techniques. Furthermore, the quality of the system design can be determined by evaluating common collaboration between its entities. The aforementioned techniques could render the architecture of *Aura* lighter, simpler and easier to understand and, consequently, expand (Gamma et al., 1994).

A novel process for producing and distributing the visualisation and interaction interfaces was found in the development of reusable components. Initially, only the AR interface was developed by following this technique, because it needed to be evaluated in other applications as well (i.e. LOCUS project). However, assembling the client application from components actually results in faster, cheaper and less error-prone solutions. Interaction with components takes place through programming interfaces that are implemented in the Interface Definition Language (IDL). These include the description of available methods and conversions, as well as support for polymorphism. More precisely, the AR interface was created in the form of an ActiveX control (ARIE). ActiveX controls are a set of COM technologies, which let the functionality to be built from binary precompiled (e.g. .dll) files that can share information between applications, regardless the programming language, which the components have been created with.

5 Applying Context-Awareness on Mobile VR & AR Interfaces

In this chapter, we introduce a context-aware application, *Aura*, designed and implemented on a mobile device platform. *Aura* can adapt its functionality according to context changes related to the user and the environment in real time. The ability to visualise contextual information through a variety of interfaces is the main feature that promotes interactivity. The implemented solution includes a scalable 2D map-based environment, a detailed virtual 3D engine and a photorealistic image-based augmented reality interface. The application queries the coupled sensors to identify modifications in context, integrates the output and adjusts the mode of interface to be employed, as requested by the user. The sequence of operating modes can vary, depending on the context and/or user's preferences. Use cases describing navigation models have been applied and more complex pervasive scenarios have been explored. The proposed framework aims for truly ubiquitous operation that will enable novel collaboration patterns to evolve, which in sequence may trigger social interaction based on proximity and user preferences. By implementing the requirements that are presented in the previous chapter, we can influence the development of the technical specification which is required for achieving the necessary results for the 4th Research Question, illustrated in Chapter 1.1. The approach that was selected to implement the requirements in the proposed framework is presented in this Chapter.

5.1 Context Acquisition

This is the most essential feature, which must be embedded in the framework. The main reason is that advanced visualisation, interactivity and other functionalities of the application are dependent on and triggered by the CMS module, either after examining the data produced by the sensors or by taking into consideration the user-specified personalisation rules. There are several classes, which implement the requested functionality and they will be presented in separate paragraphs. The class, though, which integrates and synchronises sensory input, and passes the updates to the available user interfaces is *AuraContextManager*. The class description containing a full list of attributes and procedures can be found in Appendix V. *AuraContextManager* is a high-level class, which acquires real-time data either from the sensors or the network, transforms it according to predefined data structures by adding metadata and stores it to non-volatile memory. In addition, according to the requested functionality, this class disseminates and publishes the results to the selected user interface. *AuraContextManager* was designed in order to conform to the *Singleton* design pattern because it can only have one global instance while the framework is operational. When *Aura* is launched, *AuraContextManager* is created and the highest-level thread is initialised. This thread has several modes of execution, according to the required output. The following section describes the available modes of operation in terms of functionality.

- **`COMMAND_NOCOMMAND_CM`**: This command is executed when there is no interface selected as an output component. It is applicable when the user requires only the acquisition, and possibly the transmission, of sensory data. In most cases, it is selected to store acquired context for further processing. It is the default functionality executed whenever *AuraContextManager's* main thread is up and running.
- **`COMMAND_LOCATION_TO_VR`**: This command is executed when there is only GPS input. Additionally, network context can be utilised for presenting the position of a remote entity. Visualisation of the environment and of any significant entities that exist in it takes place in the VR interface. This command can also be used to store the acquired context for further processing. Accumulating position updates triggers orientation information.
- **`COMMAND_POS_OR_TO_VR`**: This command is executed when there is position and orientation input. Additionally, network context can be utilised for presenting the position of a remote entity. Visualisation of the environment and of any significant entities takes place in the VR interface. This command can be used to store the acquired context for further processing. If for some reason compass input becomes unavailable, then the previous command, as a fallback mechanism, replaces this one.
- **`COMMAND_POS_OR_TO_AR`**: In order for this command to be executed, position and orientation input from the sensors must be available. Additionally, network-generated context can be utilised for presenting the position of a remote

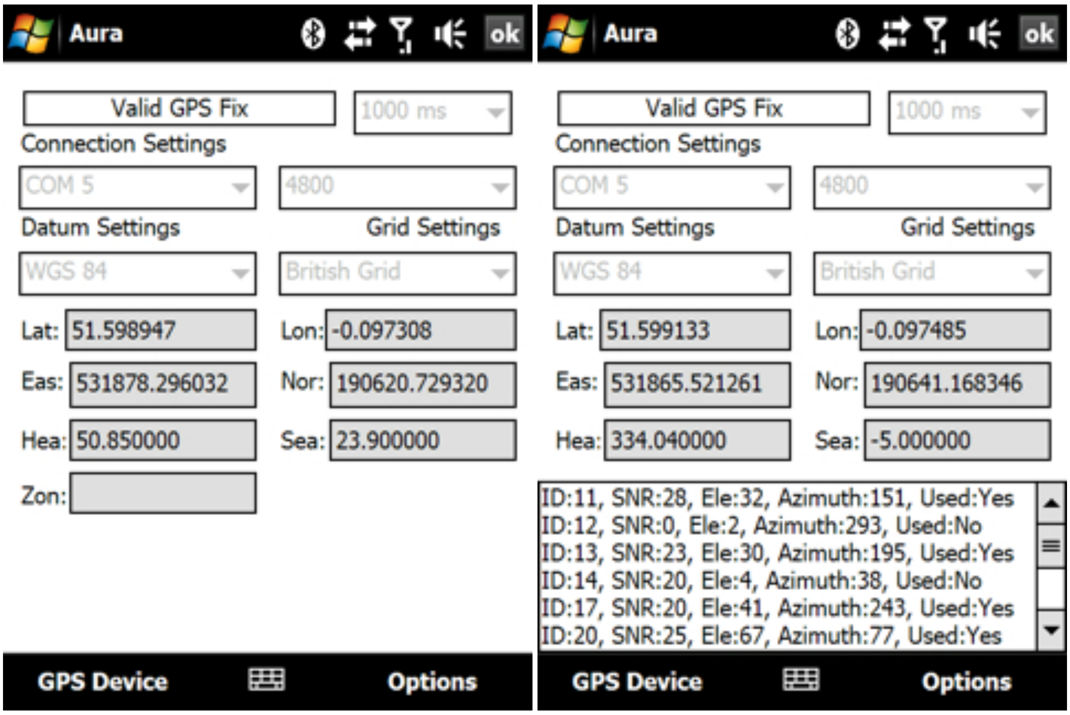
entity. Visualisation of the environment and of any significant entities that exist in it takes place on the AR interface. This command is also used to store the acquired context for further processing.

Since *AuraContextManager* is a fundamental class of the system, we are going to refer to it several times in this chapter. The following paragraphs describe the technical details of how *Aura* acquires real-time context, and the engineering specifics that have been applied.

5.1.1 Location Context

In this section, the reader can find information about the acquisition of location context by *Aura*. It is considered to be the most important contextual attribute of the framework, as most functionality is built upon it. Currently, there are two classes that implement the acquisition of position updates. These are the *GpsController* and the *GpsDlg* classes. The description of these classes, illustrating a full list of attributes and procedures can be found in Appendix V. *GpsController* is essentially the GPS input parser and *GpsDlg* is the interface, which the user can call to adjust the settings of the underlying parser. Although *GpsDlg* depends on *GpsController*, both classes are highly cohesive and decoupled. This means that *GpsController* may be used in other implementations that require GPS functionality.

GpsController is based on the *Singleton* design pattern. The reason is that it is governed by events, which are triggered when new system data becomes available and that it provides several methods, which can be called by the container class in order to either alter the functionality or retrieve the latest data. The class has several groups of internal variables, which are updated after the occurrence of specific events. As a result, whenever new data becomes available, it is stored to the object's internal memory. The internal variables are updated according to the changes in context. The update interval is synchronous and it depends solely on the hardware sensor and its operating frequency. The content of the internal variables must be accessible by other classes of the framework and, specifically, by the *GpsDlg* at any time. This functionality is accommodated by the higher-level class methods, which are concerned with the acquisition of the internal values. Calling these higher-level methods is asynchronous and does not restrict concurrent updates of the internal variables. This is accomplished by placing the internal variable updates in *critical sections*, which are thread safety mechanisms.



Valid GPS Fix
1000 ms
Connection Settings
COM 5
4800
Datum Settings
Grid Settings
WGS 84
British Grid
Lat: 51.598947
Lon: -0.097308
Eas: 531878.296032
Nor: 190620.729320
Hea: 50.850000
Sea: 23.900000
Zon:

Valid GPS Fix
1000 ms
Connection Settings
COM 5
4800
Datum Settings
Grid Settings
WGS 84
British Grid
Lat: 51.599133
Lon: -0.097485
Eas: 531865.521261
Nor: 190641.168346
Hea: 334.040000
Sea: -5.000000

```
ID:11, SNR:28, Ele:32, Azimuth:151, Used:Yes
ID:12, SNR:0, Ele:2, Azimuth:293, Used:No
ID:13, SNR:23, Ele:30, Azimuth:195, Used:Yes
ID:14, SNR:20, Ele:4, Azimuth:38, Used:No
ID:17, SNR:20, Ele:41, Azimuth:243, Used:Yes
ID:20, SNR:25, Ele:67, Azimuth:77, Used:Yes
```

Figure 5-1: Aura's interface for GPS connectivity a) Normal view b) Satellite view

Aura senses and derives a large number of variables as location context. These attributes can be grouped in distinct categories according to their purpose and importance. *GpsController* is the class, which manages these groups, because a GPS receiver generates them. The following tables present these variables according to the group that they reside in. They can be retrieved by any application component and in turn by the user at any time during system operation. These attributes are essentially the main output of the of the *GpsController* class.

Position Information			
Latitude	Easting	Datum	Altitude
Longitude	Northing	Grid	Zone

Table 5-1: Aura's Position Information Output

Movement Information		
Heading	Magnetic Variation	Speed

Table 5-2: Aura's Movement Information Output

Furthermore, the *GpsController* processes more output variables, which are directly associated with the functionality of the system and the quality of the produced location context (i.e. *Satellite Information*). This way, *Aura* has the potential to be mildly proactive when location context is required. The input attributes (i.e. *GPS Communication Information*) do not need to be explicitly managed by the user of the system because they require further technical expertise to disseminate. That is why we have developed high-level functions, which let the user initialise and terminate location context acquisition, without having any prior knowledge about the technical details. These functions either load the default variables or automatically search for valid GPS sensor input. The following tables present these system-level attributes.

GPS Communication Information			
COM Port	BAUD Rate	Timeout	Fix Type

Table 5-3: Aura's GPS Communication Information Input

Satellite Information			
Satellite ID (PRN)	Signal-to-Noise-Ratio (SNR)	Elevation	Azimuth

Table 5-4: Aura's Satellite Information Output

Aura natively supports a large number of geographic coordinate systems. The first reason to implement this functionality is that *Aura* needs to fully support ubiquitous operation. In addition, users of the system may reside at distant places, maybe even in different countries, and they may be used to working with different coordinate systems. Implementing all of these options, which offer real-time coordinate transformation between 27 datums and 51 grids, the users of the system can communicate location information, without the need to manually adjust any system properties. This way, geographic expertise is not required to operate the system. When initiating position context acquisition, the user must select a specific datum and grid, in order to visualise information in a supported format. If the user does not want to use a custom datum or if he or she is not interested in such detail, the default settings are loaded. In this case, the default settings are the *WGS 84 Datum* and the *British Grid*. The reason is that the framework has been developed in the U.K. and these are the traditional coordinate systems utilised by geographers in this country. The default settings can be altered for operation under different conditions. Geographic coordinate transformations take place several times during system operation and it was found particularly useful to offer all of these options, especially when context is being exchanged over the network.

Supported Datum(s)		
WGS 84	ETRS 89	OSGB 36
CH 1903+	RT 90	IRELAND 65
FINLAND HAYFORD	LUREF	WGS 72
AGD 84	GDA 94	MGI
NZGD 49	NZGD 2000	NTF
BD 72	ED 50	POTSDAM
NAD 83	NAD 27 Alaska	NAD 27 East
NAD 27 Conus	NAD 27 West	AMERSFOORT
ROME 40	NGO 48	Teknisk 50

Table 5-5: Datums supported by Aura

Supported Grid(s)		
UTM North	UTM South	British Grid
Irish Grid	Irish Grid ITM	Swedish Grid
Swiss Grid	Swiss Grid Lv95	Finnish Grid Zone1
Finnish Grid Zone2	Finnish Grid Zone3	Finnish Grid Zone4
UTM ETRS 89	Luxembourg (LUREF)	AMG 84
Map Grid of Australia	Austrain Grid M28	Austrain Grid M31
Austrain Grid M34	New Zealand 2000	New Zealand 1972
Belgium Grid	French Grid Zone 1	French Grid Zone 2
French Grid Zone 3	French Grid Zone 4	French Grid Lambert 93
UTM ED 50	DHDN (Germany)	SPCS 27
SPCS 83	UTM NAD 27	UTM NAD 83
RD / Amersfoort	Italian Grid Zone 1	Italian Grid Zone 2
Norwegian Grid Zone 1	Norwegian Grid Zone 2	Norwegian Grid Zone 3
Norwegian Grid Zone 4	Norwegian Grid Zone 5	Norwegian Grid Zone 6

Norwegian Grid Zone 7	Norwegian Grid Zone 8	France Lambert 2 Entendu
France Grand Champ	Denmark UTM 32	Denmark Mainland
Denmark Bornholm	Sweden Linkoping	Swedish 5gonV

Table 5-6: Grids supported by Aura

A context updating issue relevant to position that has been examined, is the ability to alter between the available visualisation interfaces based on the speed of user movement. It is impractical and performance prohibitive to present the egocentric VR perspective when the actor is moving over a predefined speed limit. Therefore, automatic visualisation of the allocentric plan VR perspective takes place, which relieves bound resources.

5.1.2 Orientation Context

This section presents the details regarding the acquisition of orientation information by *Aura*. It is considered to be an important contextual attribute of the framework as it provides details that trigger the advanced functionality offered by *Aura*. Currently, there are two classes that implement the acquisition and presentation of orientation updates. These are the *CompassController* and the *CompassDlg* classes. The description of these classes containing a full list of attributes and procedures can be found in Appendix V. *CompassController* is essentially the orientation input parser and *CompassDlg* is the interface, which the user can call to adjust the settings of the underlying parser. Although *CompassDlg* depends on *CompassController*, both classes are highly cohesive and decoupled. This means that *CompassController* may be used in other systems, which require orientation functionality.

CompassController is a class, which is based on the *Singleton* design pattern. This class is governed by events, which are triggered when new system data becomes available and it also provides several methods, which can be called by the container class in order to either alter functionality or retrieve the latest information. The class has a group of internal variables, which are updated after the occurrence of specific events. As a result, whenever new data is available, it is stored on the object’s internal memory. The internal variables are updated according to the changes in context. The update interval is synchronous and depends solely on the hardware sensor and its operating frequency. The content of the internal variables must be accessible by other classes of the framework and, specifically, by the *CompassDlg* at any time. This functionality is accommodated by the higher-level class methods, which are concerned with the acquisition of the internal values. Calling these higher-level methods is asynchronous and does not restrict concurrent updates of the internal variables. This is accomplished by utilising thread safety procedures.

Aura senses 3 variables as orientation context. *CompassController* is the class, which manages these attributes, due to the fact that a compass sensor generates them. The following table presents these variables. They can be retrieved by any application component and in turn by the user at any time during system operation. These attributes are essentially the main output of the of the *CompassController* class.

Orientation Information		
Yaw	Pitch	Roll

Table 5-7: Aura Orientation Information Output

The input attributes (i.e. *Compass Communication Information*) do not need to be explicitly managed by the user of the system because they require further technical expertise to control. That is why *CompassDlg* and *CompassController* have certain functions embedded, which allow the users to initialise and terminate orientation context acquisition, without having any prior knowledge about the technical details. These functions either load the default variables or automatically search for valid compass sensor input. The following table presents these attributes.

Compass Communication Information		
COM Port	BAUD Rate	Timeout

Table 5-8: Aura Compass Communication Information Input

Gesture recognition is obtained by examining orientation context. By taking advantage of the digital compass, which has been attached to the mobile device, the application can recognise current heading. Thus, informed exchange between the available interfaces can occur. The implemented functionality supports transparency between the VR and the AR environment. When the translated context verifies that the device is kept vertical to the ground, it enables the AR interface. On the other hand, when the device is inclined and parallel to the ground, the VR interface is presented to the user. Therefore, the users can alter the visualisation perspective of the surrounding environment, by changing the pitch or just moving their wrist. This functionality was not applied on the prototype that was used in the Extensive Evaluation of the system because the aim was to compare the differences of specific AR and VR features in more detail.

5.1.3 Temporal Context

Another collection of very important environmental variables, which can be sensed by the framework, is temporal context. Documenting accurate date and time is essential in various components of *Aura*, because it is considered to be a real-time system. Although a single class, which handles time management, was not found vital to implement, most classes accomplish the acquisition of this contextual attribute individually, and particularly those which are responsible for sensing and managing other contextual attributes. In essence, these classes are *AuraContextManager*, *AuraGraphManager*, *GpsController* and *CompassController*. Furthermore, several other classes of the framework need date and time information. These classes include *AuraPositionX*, *AuraRemoteProfile, GpxParser, SocketController* and *VrController*. A description of these classes will be provided in following sections of the document, as this part is dedicated to the acquisition of temporal context.

Aura assumes that the clock and the calendar of the mobile device keep valid and up-to-date date and time information. As a result, *Aura* can query the device any time that it needs to retrieve the current values. The framework records these variables in low-level C++ *SYSTEMTIME* structures. *SYSTEMTIME* specifies information with up to millisecond accuracy, which is very useful during system operation. Although the main engine of *Aura* processes date and time based on the Coordinated Universal Time (UTC) format, the presentation of these variables to the user takes place according to the local time format. This way, the system functionality can be synchronised based on one format, while the information that is presented to the user, according to another format, which can be more practical.

There are several processes in the framework, which require temporal context. More precisely, every time that *Aura* senses a new attribute change, which may be location, orientation, a remote user or a new camera frame, it attaches the acquired temporal signature to the metadata structure of this entry. As a result, system intelligence is improved and the system becomes capable of recalling prior measurements more accurately. Apart from the CMS, time is also used in the IPS entity of the framework. Apart from synchronising core system functionalities, not every temporal-context modification is presented to the visualisation interfaces at the exact time when it takes place. A descriptive example could be found in the course of sun. Its position in the sky is considered as a quantifiable environmental variable and is represented in the VR interface for helping the user register more effectively with the environment. A spotlight can be positioned in one of the six predefined locations on the virtual sky and it can simulate the sun's actual movement around the earth based on temporal information. The spotlight can be observed in five distinct locations over an imaginary arc during the day and in one during the night, with an additional change in the colour of the background sky. We did not implement advance algorithms that could accurately simulate the position of the sun around the globe; instead abstract representations of the course of the sun are presented. Such software functionality would pose performance issues in the application and it was out of the scope of the project. During night, the spotlight that is presented on the virtual sky represents the moon and it is helpful for illuminating the scene, in addition to the headlight of the camera

5.1.4 User Context

Because *Aura* is a socio-technical system, an essential aspect is the acquisition of personal user context. Although user context is considered to be a group of attributes that may trigger advanced functionality of the framework, it poses a major difference compared to the other contextual attributes, which *Aura* controls. The difference is that user context is explicitly defined by the users according to predefined personalisation rules in contrast to other contextual attributes, which are sensed or derived by the sensors and in turn by the CMS. User context is particularly valuable during interconnected sessions, when there is the need to exchange vital personal information about the participating users. In this implementation of *Aura*, we have introduced only a few basic attributes, which can identify a user. In a commercial solution, more personal attributes could be established, which would offer additional functionalities, not suitable for a research-focused prototype. Apart from the expansion of the personalisation options, user context also defines the privacy settings, which have been applied. The privacy settings may either allow or restrict the distribution of specific information, which is regarded as sensitive data. This way, the user is allowed to decide whether to establish a communication channel with a remote party and what kind of personal information to make available to that party.

Two classes are responsible for implementing the user-defined functionality in *Aura*. These are *AuraLocalProfile* and *AuraRemoteProfile*. The description of these classes presenting a full list of attributes and procedures can be found in Appendix V. *AuraLocalProfile* is responsible for storing and recalling information relevant to the local user, who is in charge of the system operation and seeks services. In contrast, *AuraRemoteProfile* objects are created and initialised only when a new connection has been established with a remote client. In essence, these two classes realise the user profiles, which need to be exchanged in order to support informal communications between users during chance or scheduled encounters. The actual profile exchange can only take place if the user consents to transmitting any type of user context. Consequently, after completing the profile exchange, the users may decide to initiate further interactions based on specific attributes, preferences or even proximity. In the current implementation, the privacy policy and the recorded preferences have been established but are limited. We did not expand on these options because the primary objective was to promote awareness and informal communication between people.

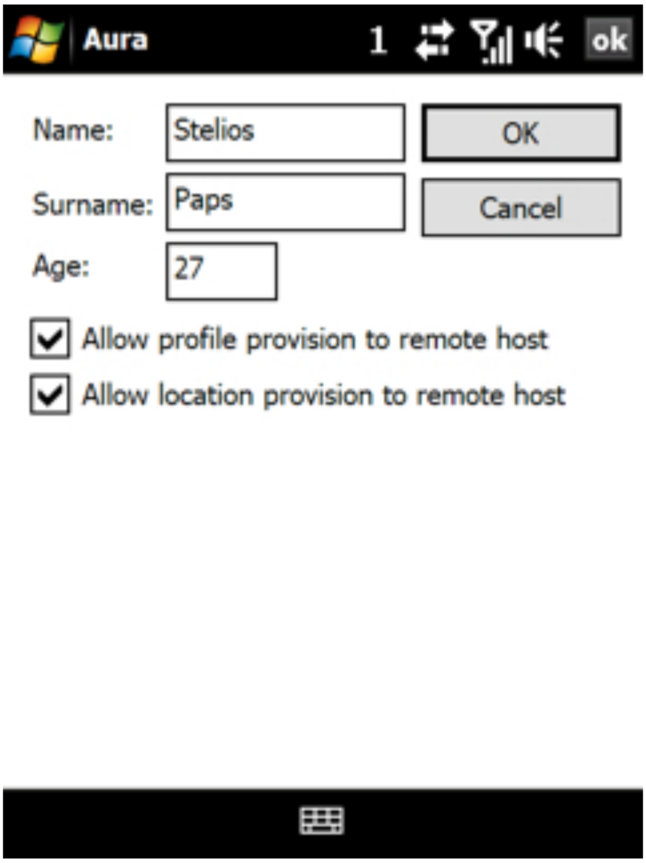
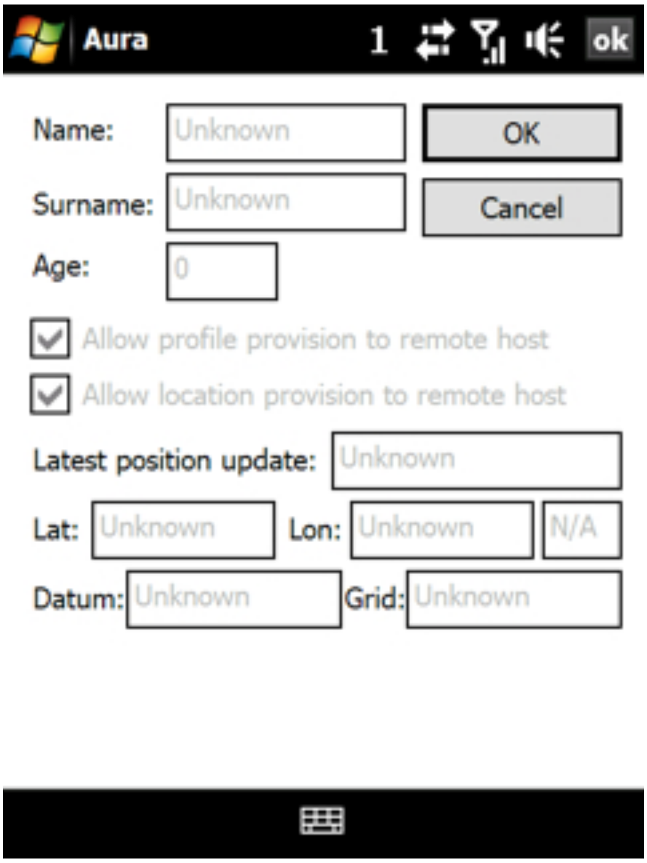

Name: Stelios
Surname: Paps
Age: 27
☑ Allow profile provision to remote host
☑ Allow location provision to remote host

Figure 5-2: Aura's interface for accessing the local user profile

The system defines and either allows or restricts further interaction between the clients through both user profile classes. Furthermore, any other class of the system and especially *CSocketController* can call these two classes. *CSocketController* is responsible for network communications. This is accomplished by designing both profile classes to conform to the *Singleton* design pattern due to the fact that there can be only one user of each type. Thus, the options provided by *AuraLocalProfile* and *AuraRemoteProfile* are decisive regarding the framework's authorised functionalities.



Name: Unknown
Surname: Unknown
Age: 0
☑ Allow profile provision to remote host
☑ Allow location provision to remote host
Latest position update: Unknown
Lat: Unknown
Lon: Unknown N/A
Datum: Unknown
Grid: Unknown

Figure 5-3: Aura's interface for accessing the remote user profile

Aura's users can be characterised by a number of context variables, which formulate their personal profile. These attributes can be grouped in categories according to their purpose and significance. *AuraLocalProfile* and *AuraRemoteProfile* manage these groups, because the users generate the required information. Although it was found that four high-level attribute categories are required for new encounters (i.e. people who don't know each other*)*, the solution which has been implemented keeps information about three categories. Alternatively, it was found that three categories are adequate when there is the need to contact familiar persons. Namely, the four high-level user context categories are listed below.

- Personal Information
- Privacy Settings
- Location Information
- Personal Preferences

The attribute categories, which have been introduced in *Aura,* are *Personal Information*, *Privacy Settings* and *Location Information*. We decided not to introduce the *Personal Preferences* category because there was no plan to over expand on this part of the project, as it would become application-dependent. Despite that, specific attributes of *Personal Information*, such as age, can be considered as a determining factor of whether a user would like to continue communicating with a remote party or not. The following tables present the user variables according to the category that they reside in. They can be queried by the application and altered by the user at any time during system operation. These attributes are essentially the main output of *AuraLocalProfile* and *AuraRemoteProfile* classes.

Personal User Profile		
Name	Surname	Age

Table 5-9: Personal Information processed by Aura

The following table lists the 2 *Boolean* variables, which are required by the system in order to enable or disable the communication of relevant information. The user may decide whether to publish personal and location context. In terms of operational sequence, directly after establishing a connection with a remote entity, the user profiles are exchanged. The user then has the chance to identify and evaluate the remote user details and, if the profile is found attractive enough, he or she may decide to keep the interaction channel open. Allowing the remote user to access the latest position information triggers the advanced functionalities of the framework and that particular type of information can be visualised on the bound UI.

Privacy Information	
User Profile Provision	User Location Provision

Table 5-10: Privacy Information processed by Aura

Although both classes are similar regarding the core elements that they interrogate, *AuraRemoteProfile* enables supplementary functionality. A vector object has been introduced, which records the latest position trajectories of the remote user in order to make it available to the local user. This feature offers several options to the local user as he or she can locate, navigate or even replay the path, which has been followed by the remote user. The information that is recorded in the *PositionVector* is presented in the following table.

Position Information		
Latitude	Longitude	Altitude
Time	Datum	Grid

Table 5-11: Remote Position Information processed by Aura

In this section of the chapter, we described the two classes, which are used to record personal user information. We can regard *AuraLocalProfile* as the class, which handles the personalisation and customisation settings defined by the local user, and similarly *AuraRemoteProfile* for the remote users. The complementary functionality (i.e. continuous location recording) offered by the latter class may be found useful in several operational features of the framework. For instance, we can record remote user trajectories for any kind of post-processing or even surveillance purposes. Furthermore, the following paragraphs describe how this functionality can be presented on the developed visualisation interfaces in real time.

5.2 Context Management

The following paragraphs document some crucial issues for effectively distributing contextual information to every layer of the system. The format of this information needs to be managed and normalised for compatibility reasons. The processing on different coordinate systems, in terms of the real environment, the 3D model, the camera, the device and the captured images, needs to conform to the specified requirements and models. The following sections describe these models, as well as the metadata that identifies them.

AuraPositionX

The single most important contextual attribute that is managed by *Aura* is position information. The most recent position of the user or any significant remote entity needs to be processed by the framework and fused into the relevant layer. That is why a custom mediator mechanism is required, which will be able to interconnect and transfer information between the framework components. The previous section describes the means that have been developed in order to calculate in real time the location of the user or a device. In this section, the results produced by the described process are presented. These results come in the form of a custom software class. This class is called *AuraPositionX* and is a vital entity of the system. When the framework acquires a new position update, either from the sensors or the network, it creates an object of this class, which holds and processes the relevant attributes of the location. Furthermore, *AuraPositionX* assists in the processing of the position of POIs or remote users engaged in the operation of the system. As a result, in every case where information about a specific location point is required, *Aura* examines and alters an instance of *AuraPositionX*. A description of this class containing a full list of attributes and procedures can be found in Appendix V.

In effect, *AuraPositionX* is the class, which represents the position of any active entity managed by the framework. Although the class is extensively utilised, its structure and functionality is mildly complicated because it is highly cohesive. An *AuraPositionX* object contains information about a specific position on earth. The position can be represented as latitude, longitude and altitude on a selected datum or as easting and northing coordinates on a selected grid (e.g. UTM). When an object of this class is required, one of the available constructors is called. Several constructors have been implemented because of the various potential uses of the class. In most cases, the *GpsController* class returns an *AuraPositionX* object. The object is produced after a fixed position has been acquired from the GPS sensor. This happens after a NMEA0183 RMC, GLL or GGA command has been processed which holds a valid position measurement. *AuraPositionX* provides the methods to define, to alter and to retrieve the properties of a position object, either in numerical form or as text. The properties managed by this class are presented in the following table.

Internal Position Information		
Latitude	Datum	Altitude
Longitude	Grid	Measurement Time

Table 5-12: Aura's Internal Position Information

The vital information required for the creation of an *AuraPositionX* object is *latitude*, *longitude* and *datum*. The omission of one of these attributes would produce inefficient results. That is why such inaccuracies are programmatically restricted. Furthermore, the value of *altitude* decides whether elevation information is available (i.e. a 3D fix was obtained) or only latitude and longitude information can be retrieved (i.e. a 2D fix was obtained, Altitude = 0). *Measurement Time* defines the time based on a *SYSTEMTIME* structure. It can represent the time when the position information was calculated or it can change due to custom required functionality, such as recent movement.

This class can also be used to accommodate the conversion of a position to a different datum than the default one. In addition, it can be used to convert the position to a different grid. The datums and grids supported by *Aura* have been presented in Chapter 5.5.1. Furthermore, *AuraPositionX* can be used to calculate the distance between two points. *Aura* has three distinct ways to calculate the distance between two objects.

- **Great Circle Distance (in Metres)**: This method produces accurate results over short and long distances. The radius of the Earth is assumed to be 6.366.710 metres, which results in a nautical mile of 1852 metres. This method is typically used at sea and air.
- **Great Circle Distance (in Radians)**: The value should be multiplied by the Earth's radius of choice. For instance, the former FAI standard for aviation records defined a FAI sphere with radius of 6.371.000 metres.
- **Pythagoras Theorem (in Metres)**: This method is used to calculate the distance on a grid. Typically used on land maps and in *Aura's* 2D interface. It is preferred for short distances.

AuraPositionX object is also used to calculate the bearing between two points (i.e. objects of *AuraPositionX*). *Aura* has two distinct ways to calculate the bearing between objects. The heading is specified in degrees. 0° or 360° represent North, 90° represent East, 180° represent South and 270° represent West.

- **Great Circle Bearing**: This method produces accurate results over short and long distances. The radius of the Earth is assumed to be 6.366.710 metres, which results in a nautical mile of 1852 metres. This method is typically used at sea and air.
- **Grid Bearing (in Metres)**: This method is used to calculate the heading on a grid. Typically used on land maps and in *Aura's* 2D interface.

5.2.1 Storing Context

Some useful functionality that has been introduced in *Aura* is the real-time documentation of user trajectories in logs. Two independent text parsers have been developed; a GPX parser (.gpx) and a human-readable log file parser (.log). The objective of these two components is to store on demand the position tracks, which the user has occupied while exploring the real world. These files need to conform to a certain format, which will render them flexible for archiving and interrogation. This is an important feature if the users want to keep records of their movement over a period of time. In addition, apart from recording user trajectories, the GPX files and the developed parser can describe the location of available POIs. These records can then be used to provide content for additional functionalities of the framework or they may even be fused to external applications. GPX is an XML derivative for storing and exchanging GPS data, such as waypoints, routes and tracks. The latest official XML-schema can be found by following the link attached to the Bibliography section of this report. Briefly, some common uses of the log files include the offline visualisation of previous explorations and waypoints within the VR interface (without utilising any sensors) and their distribution to other parties (i.e. remote users) who require information about the user or certain POIs held by the user. A sample of each log type is presented in the following tables.

```xml
<?xml version="1.0" encoding="UTF-16"?>
<gpx version="1.0" creator="Aura"
xsi:schemaLocation="http://www.topografix.com/GPX/1/0/gpx.xs
d xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.topografix.com/GPX/1/0"
xmlns:topografix="http://www.topografix.com/GPX/Private/Topo
Grafix/0/1">
      <email>stelios@soi.city.ac.uk</email>
      <time>2010-10-20T14:21:05Z</time>
      <trk>
            <name>Aura Track</name>
            <trkseg>
                  <trkpt lat="51.5324849999" lon="-0.103425">
                        <ele>2</ele>
                        <magvar>10.69651</magvar>
                        <time>2010-10-20T14:21:20Z</time>
                  </trkpt>
                  <trkpt lat="51.53251" lon="-0.1034217">
                        <ele>2</ele>
                        <magvar>17.295</magvar>
                        <time>2010-10-20T14:21:23Z</time>
                  </trkpt>
                  <trkpt lat="51.532599" lon="-0.1034167">
                        <ele>2</ele>
                        <magvar>30.697</magvar>
                        <time>2010-10-20T14:21:36Z</time>
                  </trkpt>
            </trkseg>
      </trk>
</gpx>
```

Table 5-13: Sample GPX file produced by Aura

```
Aura GPS Data Log file
Start Date: Saturday, 08 July 2006
GMT Time: 23:14:51
Communication Port: COM 8
BAUD Rate: Auto
Polling Rate: 1000 ms
Datum: WGS 84
Grid: British Grid

---Measurements---
M1:
Lon: -0.119135
Lat: 51.562630
186543.108069
530470.609174
Zone:
90.200000
Ell: 137.200000
0.2503870
275.590000
MV:0.000000


M2:
Lon: -0.119137
Lat: 51.562632
```

```
186543.290439
530470.488880
Zone:
90.800000
Ell: 137.800000
0.250387
275.590000
MV:0.000000
```

Table 5-14: Sample text log file produced by Aura

The previous paragraph explained that text log files are capable of storing data about previously occupied user tracks. In addition, GPX log files are capable of storing data about user tracks and information about POIs (TopoGrafix, 2011). Further description about the text log file will not be provided because its functionality is only a subset of the functionality provided by the GPX parser. In the following paragraphs, a description of the specifications of *Aura's* GPX parser is presented.

While the GPX-schema defines several elements and types, *Aura* makes use of a subset according to the requirements defined in the previous chapter. *Aura* needs to record in persistent storage the tracks of a certain user and the description of certain points of interest. That is why the GPX documents produced by *Aura* have been classified into two categories; those that describe user tracks and those that describe POIs. Although a single GPX file can hold both types of information, we are going to describe them as if they were individual documents.

A common GPX file consists of the header and either a collection of tracks (TRK) or a collection of POIs, which can be considered as waypoints (WPT). The header describes information about each GPX document. This information includes the GPX version (i.e. v1.0), the name of the application, which created the document, the location of the XML-Schema, a contact email address and the date on which the document was created.

A TRK represents an ordered list of points describing a path, which has been created in real time by a user. In that context, a TRK can hold information about the track itself (i.e. track name and track description) and one or more track segments (TRKSEG). A TRKSEG represents a list of track points (TRKPT), which are connected in order. One track is supported per file, while more track segments may be present. A new TRKSEG, which includes several TRKPT with data, is created every time that there may be a disruption in the process of retrieving data (e.g. GPS unavailability). *Aura* can distinguish between the following states of the GPS sensor.

- GPS parser not created;
- GPS parser not advised;
- Parser not started;
- GPS device not found;
- Lost connection with GPS device;
- No valid GPS fix obtained;
- GPS fix was lost;
- Valid GPX fix obtained.

The previous list shows that Aura is quite resourceful and offers exact information about the system status to the user. These status messages influence other functionalities of the framework as well and are not restricted only to logging services. Following next, every TRKPT has two core properties. These are the latitude and longitude of that point on earth, presented in decimal degrees and conforming to the WGS84 datum. Moreover, every track point can have additional elements, which provide a better description of that point. These elements include:

- Elevation, if a 3D fix has been obtained;
- The date and time that the measurement has been obtained;
- The magnetic variation, by a compass or by accumulating GPS measurements;
- The fix type, either 2D or 3D;
- The number of satellites that have been used to calculate the GPX fix.

Apart from logging the user tracks, the GPX parser can be used to retrieve and save information about certain POIs and their features. This can occur in a GPX document the same way as it did with tracks. The WPT element, which is used to describe a real-world feature, is a child node of the root element (i.e. gpx). The available metadata, which describes the feature of interest, is similar to the metadata that describes a track point. As a result, a geographic point in the trajectory of a user and a point of interest can be described by using the same details, but in the case of a waypoint, it will obviously need to hold supplementary context.

The software mechanisms, which are used to record position and orientation updates on both file types (i.e. GPX and text logs), have been implemented by two distinct classes. These classes are called *GpxParser* and *TextParser* respectively. Although the initialisation of the logging procedure takes place explicitly through the user interface, the actual implementation of the files is accommodated by another thread of the framework. During initialisation, the header of each file is saved and then the file handler is passed to the *AuraContextManager* class, which accommodates the main thread that manages the acquired context and the presentation of the results to the selected user interface. At this point, the *AuraContextManager's* thread examines the status of the sensors to verify that they are transmitting valid data. If there has not been any previously recorded data, a new TRK is created. Alternatively, if there is a problem with the device connection (e.g. Bluetooth disconnection) or if a valid fix has not been obtained, a new TRKSEG is produced. Finally, in case everything is working well, a new TRKPT and its attributes are used to populate the file.

5.2.2 Normalising & Fusing Context

Filtering Position Context

An important requirement that was identified and presented in Chapter 4 is the filtering of unwanted measurements generated by the sensors. Filtering data is a crucial functionality of *Aura* because it offers additional accuracy and increased performance of the system. There are several reasons, which can render a sensor measurement redundant. First of all, a measurement can be considered surplus to the existing pool of data. For instance, consider the retrieval of heading information. Although the framework can calculate the heading based on two independent GPS measurements, it would be inappropriate to record and process this information type if a digital compass has been connected and has already generated valid data. The main reason is that the accuracy of the data generated by the GPS is not precise enough and because the system cannot retrieve orientation information when the device is stationary. Furthermore, *Aura* has been designed to recognise out-of-bound position measurements. We define out-of-bound measurements as those that produce great disparity compared to the preceding and following. It was found critical to implement such a functionality because, even though position determination technologies are rapidly evolving, there can be several external issues that affect their precision. For instance, consider the initial time, which is required by a GPS chipset to provide accurate positioning (i.e. cold boot) and the movement in urban canyons, which evidently affects measurement accuracy due to signal multipath interference. The third reason that calls for filtering position context is the huge volume of data that can be generated during *Aura's* operation. Although the user has the ability to select the polling rate through the user interface, even in that case the sensors may generate similar or identical measurements. This may happen while the user is standing still for a specific time span. During that period the sensor should be generating identical measurements, which are stored either in volatile memory or in persistent storage (i.e. GPX log files). With optimal conditions, two or more consequent measurements would be identical. With nominal and worst-case conditions, these measurements would have minor or even major differences. That is why *Aura* must be prepared to verify and recognise such discrepancies. As a result, the positional measurements that are very similar to the former (e.g. distance smaller than 0.5 meter) and occur in a very small time frame (e.g. below 1 second) are discarded by the framework. By verifying the validity and by processing each measurement before it reaches the target pool, the framework is capable of improving its performance. Evident performance enhancement, due to the aforementioned reasons, is especially noticeable when the logging mechanisms and the information visualisation interfaces are concurrently executed. The data logging procedure becomes faster because continuous conversions of numbers (i.e. integer, float and double types) to strings of characters required by the target files (e.g. GPX documents) are effectively minimised. Likewise, continuous positional updates proportionally increase the processing requirements of the visualisation interfaces. Especially the VR and AR interfaces become quite faster because excessive measurements are not being considered and therefore decrease the supplementary calculations, which need to be accomplished.

Filtering position context in *Aura* takes place in a thread of the *AuraContextManager* class. Although the main class used for retrieving the latest occupied coordinates is *GpsController*, it does not include any of the filtering functionality. Instead, it was found more practical and efficient to implement the filtering functionality in *AuraContextManager* because it becomes easier for the user to enable or disable it according to his preferences and objectives. While any thread of *AuraContextManager* is activated there is a number of different ways in which the system can discard the redundant measurements. Initially, the refresh rate of the sensors affects the input frequency. The user is responsible for adjusting the polling rate when he initiates system operation. The offered options are 250ms, 500ms, 1000ms, 1500ms, 2000ms, 3000ms, 4000ms and 5000ms. The second way, which is used to filter position context, is by calculating the difference between the current and previous measurement. If the difference is less than half a meter, then the measurement is discarded. At this point we have to consider the fact that a user may be moving at a very low speed. That is why *Aura* does not only examine the last couple of measurements, but also takes into consideration a small pool of the preceding ones. As a result, the framework is capable of verifying whether a user is actually moving very slowly or if there are any sensor-generated inefficiencies. The last solution for filtering position context is based on the quality of the measurements produced by the GPS sensor. In Chapter 5.1.1, we mentioned that for every GPS measurement, *Aura* identifies the satellites, which generated data for the measurement and also the quality of the input based on the Signal-to-Noise Ratio (SNR). This functionality was found particularly useful in this occasion because any data that has been accompanied by very low SNR can be effectively rejected.

Location Model

Users need to acquire information about their position in the real world as well as about the position of other objects and actors. There is a possibility that not all users retrieve and manage their geographic position based on the same coordinate systems and it becomes extremely inefficient if the stored coordinates conform to different formats. This is one of the technical reasons for which a location model has been applied to the system architecture. It was found that the introduction of a location model would have a major impact on the quality of the provided functionality. Therefore, the architecture of the framework is bound to depend on a specific location model. Almost every location-sensitive system that has been examined in the Literature Review was found to be working on top of a specific location model, which stores representations of static or mobile objects for further interrogation. A very good description of available location model types is provided by Becker and Durr (Becker and Durr, 2005). They divide location models into categories and describe the advantages and disadvantages of each one. These categories are presented in the following list, including more specific classifications.

1. Geometric model;
2. Symbolic model;
    a. Set-based model;
    b. Hierarchical model;
    c. Graph-based model;
3. Hybrid model;
    a. Subspaces model;
    b. Partial Subspaces model;

The sensors that have been employed to retrieve location context offer geometric coordinates and it seemed appropriate to rely on a purely geometric model. The geographic coordinates that are retrieved by the GPS receiver can effectively provide simple spatial reasoning. Moreover, because one of the objectives of the framework is to provide adequate navigation assistance "*the topological relation 'connected to' has to be modelled, which describes interconnections between neighbouring locations*" (Becker and Durr, 2005). This is why the possibility to work on a hybrid model has also been examined. Any chosen model must support specific functions, which -according to Becker and Durr - are finding *object positions*, *distance functions*, *topological relations* and *orientation* of mobile objects. Additionally, "*simple models of location ignore the richness that arises from humans' perception of location which, if leveraged, can greatly improve a system's ability to reason with location information*" (Dobson, 2005). This means that different reasoning paths may affect the result of a given function, due to the erratic answers that people provide. The introduction of a location model will allow the users to interact with the framework based on predefined structures, implemented on technical specifications, which can provide basic pervasive reasoning. The development of an abstract geometric location model and the architecture that lies on top of it has taken place after considering the design requirements. This way, flawless information communication can be achieved between the participating entities. A specific structure has been proposed by Hightower et al. (Hightower et al., 2002) and its evaluation in the real world (Graumann et al., 2003) has directed further improvements of our system. A comprehensive set of current location technologies and their connection to the underlying location models have been studied by Hightower and Borriello (Hightower and Borriello, 2001).

Management of User Context

Chen and Kotz stated that apart from location there is a lot more information to record in a context-sensitive system. Apart from low-level context, high-level context has to be retrieved, including changes in context over time. Activity, which is considered as high-level context, can be acquired by combining several low-level context values, using any of the following methods; machine vision, user calendar and AI techniques (Chen and Kotz, 2000). User context is very important for personalisation and collaboration because similar preferences between the users might trigger potential interaction. Additionally, by recording changes we can observe alterations in user behaviour over a timeline. "*The use of user context in ambient computing is needed for several reasons: users are increasingly mobile and require ambient computing with context-aware applications; and they need personalized information services to help them in their tasks and needs*" (Göker and Myrhaug, 2002). Likewise, Göker and Myrhaug defined the optimal contextual information that forms user context for targeting personalisation functionalities. It is composed out of *environmental*, *personal*, *task*, *social* and *spatiotemporal* context. By introducing case-based reasoning between user context and other context variables activity can be recorded, which will enable advanced supervision of each mobile user solely, or in cooperation with others. This requires the introduction of explicit scenarios in order to operate effectively for the users. Profile-based management of user context is implemented based on intelligent operations, as described in Chapter 5.1.4. In addition, further collaboration may be enhanced if the visualisation of remote users takes place in a MR environment, which enables virtual interactions between the entities. Due to the fact that user context is considered as sensitive information, well-formulated privacy constraints have been applied based on each actor's personal preferences and the system managed privacy control mechanism. This way, the probability of sensitive information reaching non-authorised parties is effectively minimised.

5.3 Map (2D) Interface

Figure 5-4: Northampton Sq. represented in the Map Interface

From a Preliminary Evaluation (Papakonstantinou, 2005) and the results of other research projects described in the Literature Review, it was found that multiple concurrent perspectives of the illustrated world should be implemented in any functional context-aware, and more specifically, location-aware application. It could supply sufficient representations of the environment to the users, so that they can compare and match them with the cognitive map stored in their mind. Two-dimensional, three-dimensional and augmented representations of the user's surrounding areas are available in the current client implementation. When the research commenced, planar illustrations of the environment (i.e. digital maps) received special focus due to their low complexity and fast development cycles. From the point that the VR interface was established, though, further development of the map interface was postponed. The reason was that the research focused on more immersive solutions to represent context and due to the fact that the allocentric plan view of VR offered a better alternative to the map approach. Therefore, this section presents some core functionalities of the 2D approach, but it does not get too deep in examining specific implementation issues.

Although traditional map representations are becoming dated, commercial applications are reluctant to discard them and developers try to boost their functionality by coupling them with supplementary enhancements (Papakonstantinou and Brujic-Okretic, 2009a). The reason for which this kind of interface is still invaluable springs from the fact that it can satisfy certain user needs. The map interface has proven extremely beneficial, especially for the cognitive understanding of the distance between two objects and the topological relation of the objects in the environment. Furthermore, scalable maps have been a productive tool for the GIS community because they can hold overlaid phenomena and features of the real world in a controlled fashion (Papakonstantinou and Brujic-Okretic, 2009a). The *Requirements Acquisition Survey* analysis showed that most users feel more comfortable working with a map interface, especially when they quickly need to retrieve information without requiring extensive interactions. Furthermore, new mobile device platforms natively support map illustrations generated by online services such as Google Maps. Thus, a sensible solution for a commercial application would be to employ these already available APIs. In contrast, the reduced solution developed for this project can be considered as proof of concept and be used for specialised applications that may require geographically intense features.

Two-dimensional representations in *Aura* depend on *Franson's GpsView* API. The developed application can present vector visualisations of the environment by reproducing geo-referenced shapefiles (Papakonstantinou and Brujic-Okretic, 2009a). This is the standard format for exchanging geographic data supported by ESRI. The user obtains a vertical perspective of the represented space and every interaction takes place in abstract 2D scenes. Real-time context is classified into categories and each type is placed on a specific layer. The depth (i.e. Z-order) of each layer and the icons that are presented depend on user preferences and the current situation. Visualising the movement of the subject requires the context-controlled mode of the system to be enabled. Interaction is triggered when the user moves in the real world and consequent position updates occur. Zooming on the digital map is accomplished explicitly by the user, either with the stylus or the navigation button of the device. Automatic zooming takes place when the distance between the user and the target reaches a specific threshold. Furthermore, rotation can occur, either explicitly or via compass input. Actors can create their own POIs by clicking on any part of the map and by providing relevant metadata. These POIs are appended to new or existing GPX documents and can be used to promote further interactions if shared with other users.

5.4 Virtual Reality Interface

The proposed framework employs VRML to establish the VR environment, which is an extension of XML dedicated on the visualisation of 3D worlds. Although it does not provide advanced functionalities, such as those found in *Mobile Direct3D*, *OpenGL ES* and *M3G*, this format can hold geo-referenced information (i.e. GeoVRML) and is still a widespread standard for exchanging 3D content over the web as well as other networking applications. Moreover, VRML is a platform-independent mark-up language, which improves compatibility between diverse platforms. As a result, there are several VRML APIs, which can be attached to a variety of applications, ranging from native code applications, like *Aura*, to higher-level applications, such as interpreted code in a web browser. There are several VRML SDKs available to developers but for the implementation of *Aura,* ParallelGraphics *PocketCortona* was selected. The reason is that this software development kit offered the best support for windows-based mobile devices. *PocketCortona* has been very popular, and applications that exploit this software can be found in many vertical business models such as the mining or e-learning industry. The 3D models have been developed and enhanced with semi-automatic modelling techniques out of spatial datasets originating from geo-referenced shapefiles. As a result, the detailed 2D and 3D environments conform to the same location model and coordinate systems, which makes interaction design for both interfaces easier to capitalise (Papakonstantinou and Brujic-Okretic, 2009a). The VR interface is considered as a core component of *Aura's* Information Presentation System (IPS) and it has sustained a lot of optimisation and refactoring during the course of the project. At this moment, it is capable of supporting the advanced requirements of the proposed framework. The main functionality of the VR interface is to visualise and enable interaction with the user. Synchronisation of the virtual environment with the location data takes place under the explicit control of the user. The frequency of context reception varies according to the user preferences, but some options are not suitable for all situations. Further details about this issue can be found in the Evaluation chapter of the report.

The visualisation of specific information that describes existing entities of the environment is a core component of the framework because it is the output on which the user must rely in order to make informed decisions. Currently, there are two classes that implement the visualisation mechanisms of the virtual reality interface. These are the

VrController and the *VrDlg* classes. A description of these classes containing a full list of attributes and procedures can be found in the 5TH Appendix. *VrController* is the class, which can be considered as the low-level VR engine of the framework. In contrast, *VrDlg* is a higher-level class, which presents to the user translated information, the available options and also receives potential interactions. Although *VrDlg* depends on *VrController*, their functionality is decoupled. Consequently, *VrController* can be used in other implementations, which take advantage of the *Pocket Cortona* API.

VrController is a wrapper class, which offers several functionalities. *VrController* is based on the *Singleton* design pattern, because it can be instantiated only once during framework operation. This way, *VrDlg* or any other class may concurrently call commands, which are interpreted and rapidly presented on the virtual environment. In a normal use case scenario the VRML file, which holds the geometry and possibly the textures of the represented scene, is initially loaded in the graphics engine. The user can select which file to load through *VrDlg*. After selecting the VRML world, a series of intrinsic *VrController* operations are executed in sequence. A great advantage of this class is that it can load a variety of VRML extensions in a straightforward manner for the user without experiencing any compatibility issues. Currently, *VrController* has been trained to recognise the following VRML file extensions.

1. VRML97 files (original format, ISO/IEC 14772:1997)
2. VRML files produced by ESRI’s ArcGIS ArcScene (3D Analyst);
3. VRML files produced by Autodesk’s 3D Studio Max;
4. Geo-referenced VRML files (GeoVRML extension).

Although it may seem trivial, processing these extensions is a core functionality, which has been introduced to the framework. Each VRML extension has specific particularities, which need to be accommodated. *ArcGIS* output files have their geometry reversed. In essence, they translate latitude to longitude and vice versa. *3ds Max* (Autodesk, 2011) output files support specific VRML node extensions, which are not compatible with *Pocket Cortona* and need to be translated to primitive nodes (i.e. VRML97). Finally, GeoVRML file types have higher precision (i.e. digits after the decimal point), which is not natively supported by *Pocket Cortona*. Therefore, *VrController* has implemented all required options for efficient visualisation of a 3D scene, regardless of the underlying constraints.

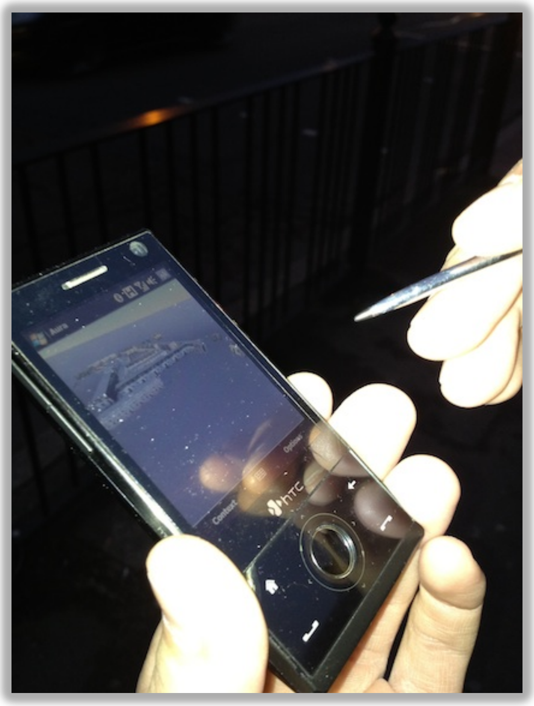


Figure 5-5: A user explores the Virtual Environment, in the manual mode of operation

Another important functionality that takes place after loading the selected 3D scene is the count and indexing the initial VRML nodes that have been described in the source file. This functionality is imperative because several additional nodes will be introduced during execution that will describe the dynamic entities of the environment. Thus, when there is the need to access any of the new nodes for moving or even deleting them from the scene, the framework has an indexed account of which node to query at any given time. Further essential functionality incorporated in the *VrController* class includes the initial setup of the 3D environment, which consists of introducing a headlight that always points in the direction of the camera, setting up the collision detection options, as well as managing the transition method that will be used to move between viewpoints. In the current implementation, the viewer moves to the new viewpoint with a transition effect, and not instantaneously. This method offers a more natural look-and-feel movement between geographic points, by removing the *jump* effect. Finally, *VrController* offers several functions, which actively manage the performance of the VR interface. A dedicated mechanism that identifies whether VR is the currently bound interface has been developed because it is heavily dependent on the hardware specifications of the mobile device. This way, whenever the virtual environment is not visible, its thread priority is lowered for promoting efficient use of other components. Moreover, the frequency of updates (i.e. Frames Per Second - FPS) that reach the screen of the device is also controllable. For instance, when several entities need to be loaded, the scene is not updated after the introduction of every single entity. In that case, the framework waits until every entity has been loaded to memory and then the screen is updated once, to reflect all changes.

VrDlg is also responsible for accommodating the user's input and for passing the required information to *VrController* for further representation. *VrDlg* is formed out of two main components; a menu, which offers external options to the user, and the actual VR interface that accepts user input and presents the output of the internal processes. Interaction with the VR interface is directly handled by the *VrController*. In contrast, the *VrDlg* menu offers several options to the user. These options are documented in the following list and can be seen in several figures of this chapter.

- Switch between the *Manual* and *Sensor-controlled* mode of the framework;
- Load or unload tracks from files (e.g. GPX) by using the *GpxParser* class;
- Load or unload waypoints from files (e.g. GPX) by using the *GpxParser* class;
- Hide or reveal any remote users from the virtual environment;
- Alter the field of view;
- Visit the previous or following waypoint;
- Animate between viewpoints;
- Change the perspective of the environment between egocentric view, allocentric oblique view and bird's eye view.

The users may tweak further options of the VR engine by accessing the context menu (i.e. click and hold) of the VR interface. These options will not be explored in detail due to their exclusive relation to the user preferences.

5.4.1 Three-Dimensional Content Creation

Semi-Automatic Urban Modelling

Implicit objectives of the research include tasks such as modelling urban environments and using visualisation concepts and techniques to aid navigation with a mobile device (Liarokapis et al., 2006a). Currently, the surrounding scene is manually modelled and the output is applied on scenarios that require exploration of a location. It is important for a user to smoothly navigate in a VE and every effort has been made to accommodate that need. The interaction and collaboration potential of the VR interface is increased by introducing multiple collaborative actors. A company based in Cambridge called the GeoInformation Group (GeoInformation Group, 2010), which distributes geographic information services to Google (e.g. Earth and Maps products) as well, provided a comprehensive spatial dataset containing building heights and footprint data for the city of London. In the case of our system, the proprietary software applications that have been used to assemble the 3D models include *ArcGIS* Desktop (ESRI, 2011), shown in the next figure, and Ordnance Survey mapping products (Ordnance Survey, 2011). The generation of a 3D model describing a specific scene of the world is accomplished manually by using the aforementioned products. The process involves loading vector shapefiles into ArcGIS, extruding building footprints and heights out of aerial photography data and converting them to VRML files. Still, none of the mobile device platforms offer native support for GeoVRML. Furthermore, the produced 3D world is enhanced with textures of buildings and special landmarks. Textures are photographs of the real environment that have been manually captured by using high definition digital cameras. "*The steps in the semi-automated technique for preparing and texturing the 3D meshes include: detaching the objects in the scene; un-flipping the mesh normals; unifying the mesh normals; collapsing mesh faces into polygons and texturing the faces*" (Liarokapis et al., 2006a). In addition, further options on improving the level-of-detail of the generated 3D worlds have been explored (Gatzidis et al., 2008). This has been accomplished by evaluating and applying Non-Photorealistic Rendering (NPR) techniques, which may not appear very realistic to the user, but are a valuable substitute for certain cases.

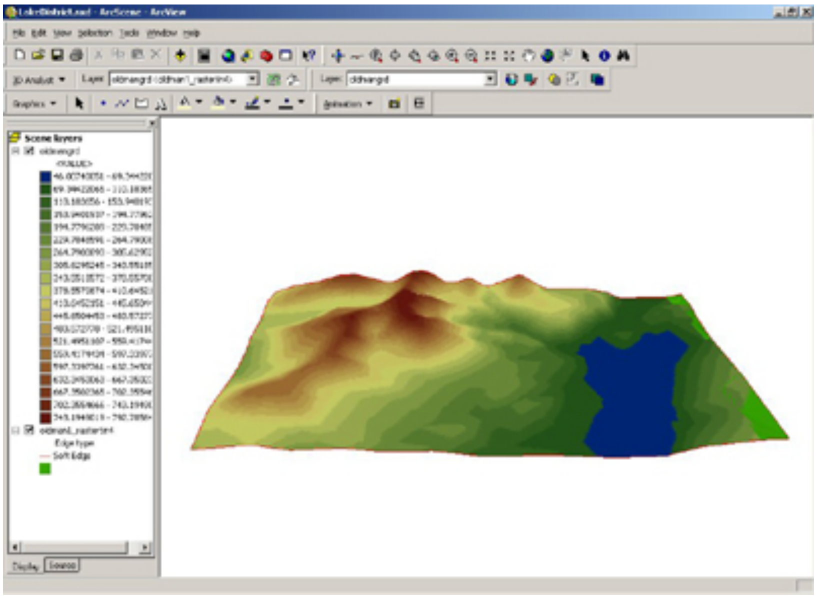


Figure 5-6: Modelling the environment with ESRI ArcScene

An identified issue that affects the modelling of an urban environment is that the size of the 3D product might be quite large, and as a result, it severely affects performance when loaded on a mobile device. There are 2 types of files that must be downloaded to the device, in order to flawlessly visualise the 3D world. These files consist of the actual 3D model that conforms to the *.wrl* extension and the textures, which come in the form of supported computer images (e.g. *.gif*, *.jpg*, *.bmp*). Early Windows Mobile devices did not support the advanced graphics specifications, which current devices do. Furthermore, in most cases, it is good practice to minimise the size of the output models without severely affecting their quality because the utilisation of hardware resources (i.e. RAM) is effectively minimised. Following certain methodologies can reduce the size of the resulting product in a way that is not recognisable by the end-users. This process is called data model *generalisation* and it can severely decrease the size of 3D scenes, from several megabytes to a few hundred kilobytes.

There are several techniques for data generalisation in order to minimise the size of the resulting file. The analytical documentation of the process, which has been applied to generate 3D scenes from the *Cities Revealed* spatial dataset (GeoInformation Group, 2010), is described below. The description conforms to the terms provided by the software tools (e.g. ArcGIS).

1. [Optional] *Merge* [ARCTOOLS] two or more shapefile tiles from GIG dataset if required region is on join of N tiles;
2. *Create new shapefile* with area geometry [ARCCAT] and *edit* to create mask of area of interest [ARCMAP];
3. *Clip* the required model from the parent set using the mask. There are three ways of doing it: *Intersect*, *Clip* or *Select* / *Copy* [ARCTOOLS / ARCVIEW];
4. Convert shapefile to coverage [ARCINFO]. WARNING: The attributes will be lost;
5. Use the dissolve tool on the coverage file, to remove internal building geometry [ARCTOOLS];
6. Use the *BuildingSimplify* command [ARCINFO] to reduce building geometric complexity. WARNING: It will convert polygon to polyline geometry;

7. Convert polyline coverage to polyline shapefile [ARCINFO];
8. Convert polyline shapefile to polygon shapefile [ET GEOWIZARDS];
9. Transfer attributes to new polygon shapefile: either do a *Join* based upon spatial location or use the transfer attributes method [ET GEOWIZARDS] to transfer building heights data from original shapefile to simplified model;
10. [Optional] *Select by Attributes* [ARCMAP] to select only buildings above a certain height and export selection to a new shapefile [ARCMAP]. This will reduce the size of the scene.
11. Open shapefile in ARCSCENE and extrude to height attribute;
12. [Optional] Add aerial photography from the GIG as a backdrop to buildings [ARCSCENE];
13. [Optional] Add Digital Elevation Model (DEM) surface from OS (Profile / Panorama) for base heights [ARCSCENE];
14. Export as VRML scene [ARCSCENE];
15. [Optional] Use the CHISEL software to further reduce the scene size.

Automatic 3D Scene Reconstruction

The content for the VE of *Aura* is created out of geo-referenced data. It would be ideal if the process became fully automated in order to ubiquitously cope with 3D scene reconstructions, at any time and by any client. The conceived scenario consists of the following tasks; transmission of the occupied geographic coordinates from a mobile client to a server running a custom application, in turn the server selects the appropriate shapefile, crops a region around the user, extrudes the building heights, converts the result to VRML, saves it in indexed storage and distributes it to the mobile device that requested it. The region that will be represented depends on several parameters, such as the user's speed (i.e. a larger area must be modelled if the user is travelling fast) and the hardware capabilities of the client device, for making generalisations on the produced model. Obviously, the quality of the model will not be high because texturing and other techniques for enhancing the level of detail cannot be implemented due to the fact that there are not any available pictures from the environment. Although the produced 3D scene will be an abstract reproduction of the real world, additional client-side functionality can be attached, in order to enhance immersion. For instance, the use of fog could provide realistic results and enhance the performance of the graphic subsystem of the mobile device. Based on the spatial vector data that was available (i.e. shapefiles), the author made an attempt to provide the aforementioned server-side functionality. Many software packages and tools were investigated in order to find the one that provides the API to implement such functionality. Currently, there is not any software tool available that supports automatic generation of VRML models originating from shapefiles. Some applications supported only manual modelling, which is the process that we have already put to work. Moreover, it seemed unreasonable to avoid ArcGIS (i.e. 3D Analyst extension) and prefer other software packages, because of the advanced functionalities that are offered by this group of GIS tools. Development in ArcGIS is supported on almost any platform through COM, .NET and Java implementations. ArcGIS APIs come into three distinct editions, which are *Desktop*, *Engine* and *Server* (ESRI, 2011). Development on the *Desktop* edition provides customisation and extension for the main user interface of ArcGIS, which makes the semi-automatic procedure for urban modelling faster, but not fully automated, because it requires explicit user input. In the *Server* edition, web application development is supported and, finally, *Engine* is the package required to build completely new standalone solutions, which utilise abstract ArcGIS functionalities. The version that has been found suitable for fully automated 3D scene reconstruction is *Engine*, because a brand new application can be developed based on custom input (i.e. GPS coordinates and shapefiles) and specific output (i.e. VRML scenes). The problem in completing this procedure is that the author managed to obtain only the *Desktop* version, which does not completely satisfy the requirements. Although it would be ideal if 3D scenes could automatically be reconstructed in real time out of existing data, customised and distributed to the appropriate client device, this effort has been postponed due to the fact that we selected to focus the development efforts on the client side of the framework.

5.4.2 Information Intensity

This section describes how *Aura* translates specific contextual variables into information and how this information can be visualised in the dedicated virtual environment. The accuracy of this procedure is very important in terms of functionality because it offers to the users of the system a visual feedback, which aims to enhance their decision-making process and inform them in real time about the available options or about further interactions that may be required to achieve their goal. *Aura's* VR interface can visualise the following absolute entities.

- The environment surrounding the user;
- Any potentially remarkable elements that exist in the environment (e.g. POIs);
- Any available remote users that have been connected;
- The local user operating the system.

Furthermore, higher-level entities can be visualised in the VR interface to provide assistance and to complement the de facto view of the user about the world. These entities are described below.

- Track points in space, which the user has already occupied until now;
- Track paths in space, which the user must follow to reach an objective;
- Specific views of the environment, which illustrate a target.

Figure 5-7: The Context Menu in the VR Interface used for A) loading additional Context B) representing a Remote User

Every element type is expressed by a specific representation. Although it would be ideal to demonstrate each element type as faithfully as possible, this could not be achieved because of the diversity of existing categories. For example, there is no difference, in terms of the representing entity, between a blonde and a brunette user or between a bank and a grocery shop. As a result, the classification that has been used to illustrate any interesting entities of the environment is comprised out of the alternatives that have been declared in this paragraph. *Aura* is using a dedicated bank of custom 3D elements, which are called whenever there is a need to depict a particular entity or attribute.

During trial, we found that various operational purposes needed to dynamically introduce 3D elements in the virtual environment – and, in many cases, change their location – or, subsequently, remove them. These elements, amongst all others, which we have referred to, are geographically referenced and positioned according to their real-world coordinates. Therefore, it is possible to adjust the location of every element according to real-time information. This can be done through the menu items depicted in Figure 5-7. Furthermore, the Context Menu can be used to show or hide the remote user that is currently connected with *Aura*.

The process, which enables the loading of selected features (e.g. tracks, POIs, waypoints) from persistent storage, translating them to VRML nodes and presenting them to the interface, needs to take advantage of three framework classes. These classes are *VrDlg, GpxParser, VrController*. Initially the user must select which kind of features should be loaded in the virtual environment. These can be either points or tracks (i.e. a series of interconnected points). The next step is to locate and select the file that holds the appropriate content. *Aura* can load data from *.gpx*, *.xml* or any other text file that describes the features in a compatible way. The first two steps are accomplished by inputting information to the *VrDlg* class. In turn, *VrDlg* calls the appropriate function from *GpxParser*. The *GpxParser* functions, which query the source file and translate the input, are:

```
•  bool CGpxParser::LoadGpxTracks(CString thePath);
•  bool CGpxParser::LoadGpxWayPoints(CString thePath);
```

Table 5-15: Aura’s functions for transforming GPX to 3D content

Both functions make extensive use of processor time because they manipulate strings of characters and translate the required attributes to new variables, which require a lot of operating memory. Furthermore, both functions include optimisation and error-checking commands because the input needs to be verified according to the requirements of the framework. There may be cases when the information obtained from the source file is not complete or may not be valid, for instance, when there is no altitude information attached to a specific entity or when certain values do not conform to the specifications (e.g. orientation > 360°). Consequently, it is up to the *GpxParser* class to recognise such discrepancies and correct them on-the-fly. As a result, execution of these functions is a time-consuming and complicated procedure.

For every valid object described in the source document, the *GpxParser* calls a different method of *VrController* in order to draw the 3D node describing that feature. Earlier, we listed the various features, which can be visualised in *Aura*. Consequently, a different method is called for drawing each feature, based on the entity type. The following section describes how each feature is presented in the VR environment according to its type.

The following VRML code adds a 3D cone (i.e. *AuraCones*) in the rendered scene and represents a waypoint in the environment. It is considered as a decision point, which the users must reach and, subsequently, alter their behaviour.

```
VRMLbuf.Format(_T("DEF AuraCone Transform { rotation 1.0 0.0
0.0 3.14 center 0.0 0.0 0.0 translation %.6f %.6f %.6f
children Shape { appearance DEF Brown Appearance { material
Material { diffuseColor 0.6 0.4 0.0 shininess 0.2 } } geometry
Cone { height 10.0 bottomRadius 2.5 side TRUE bottom TRUE } }
}"), xEasting, yNorthing, zAltitude);
```

Table 5-16: On-the-fly development of 3D AuraCone objects

AuraSpheres are small 3D spheres that are placed along the path that the user must follow in order to reach a decision point. They can be considered as GPS track points. They are grouped and coloured according to the track segment to which they belong.

```
VRMLbuf.Format(_T("DEF AuraSphere Transform { rotation 1.0 0.0
0.0 3.14 center 0.0 0.0 0.0 translation %.6f %.6f %.6f
children Shape { appearance DEF Brown Appearance { material
Material { diffuseColor 0.6 0.4 0.0 emissiveColor 0.6 0.4 0.0
shininess 0.2 } } geometry Sphere { radius 1 } } }"),
xEasting, yNorthing, zAltitude);
```

Table 5-17: On-the-fly development of 3D AuraSphere objects

AuraViewPoints represent self-explanatory views of the environment that have been explicitly recorded by a user and assist the registration process.

```
VRMLbuf.Format(_T("DEF AuraViewPoint Viewpoint { jump FALSE
fieldOfView 1.0 orientation 1 0 0 -0.785398 position %.6f %.6f
%.6f description \"%s\" }"), xEasting, yNorthing,
zAltitude+10, vpName);
```

Table 5-18: On-the-fly development of 3D AuraViewPoint objects

AuraLines show a track segment, which connects track points, and must be followed by the user in order to reach a new decision point. As input, they require the coordinates of the two track points that should be connected.

```
VRMLbuf.Format(_T("DEF AuraLine Shape { geometry
IndexedLineSet { coord Coordinate { point [ %.6f %.6f %.6f,
%.6f %.6f %.6f]} coordIndex [ 0 1 ] color Color { color [ 0 1
0, 0 1 0] } } }"), xEasting1, yNorthing1, zAltitude1,
xEasting2, yNorthing2, zAltitude2);
```

Table 5-19: On-the-fly development of 3D AuraLine objects

AuraExaLines are 3D polygons, comparable to *AuraLines* that connect two waypoints in the VE. The result looks like a rectangular prism of variable length.

```
VRMLbuf.Format(_T("DEF AuraExaLine Shape{ geometry
IndexedFaceSet { coord Coordinate { point [%.6f %.6f %.6f,
%.6f %.6f %.6f, %.6f %.6f %.6f, %.6f %.6f %.6f, %.6f %.6f
%.6f, %.6f %.6f %.6f, %.6f %.6f %.6f, %.6f %.6f %.6f] }
coordIndex [ 0, 1, 2, 3, -1, 7, 6, 5, 4, -1, 0, 4, 5, 1, -1,
1, 5, 6, 2, -1, 2, 6, 7, 3, -1, 3, 7, 4, 0 ] color Color {
color [ 0 0 1, 0 0 1, 1 0 0, 0 1 0, 1 0 0, 0 1 0 ]}
colorPerVertex FALSE } }"), xEasting1-0.5, yNorthing1,
Altitude1+0.5, xEasting1+0.5, yNorthing1, zAltitude1+0.5,
xEasting1+0.5, yNorthing1, zAltitude1-0.5, xEasting1-0.5,
yNorthing1, zAltitude1-0.5, xEasting2-0.5, yNorthing2,
zAltitude2+0.5, xEasting2+0.5, yNorthing2, zAltitude2+0.5,
xEasting2+0.5, yNorthing2, zAltitude2-0.5, xEasting2-0.5,
yNorthing2, zAltitude2-0.5);
```

Table 5-20: On-the-fly development of 3D AuraExaLine objects

Fog supports two purposes, to simulate real-world behaviour and to enhance system performance by hiding 3D content, which is far from the user and does not need to be rendered. This is managed by the VRML *visibilityRange* parameter.

```
VRMLbuf.Format(_T("DEF AuraFog Fog {color 1.0 1.0 1.0 fogType
\"EXPONENTIAL\" visibilityRange %i }"), visibilityRange);
```

Table 5-21: On-the-fly development of 3D AuraFog

AuraSignPosts are 3D signs that are placed over specific POIs and present descriptive information about them. They are also used to easily locate these objects in the VE.

```
VRMLbuf.Format(_T("DEF AuraSignPost Transform { translation
%.6f %.6f %.6f children [ Billboard { children [ Group {
children [ Shape { geometry Text { string [ \"%s\" ] fontStyle
FontStyle { justify \"MIDDLE\" style \"BOLD\" } } appearance
Appearance { material Material { emissiveColor 1 1 1 } } }
Transform { rotation 1 0 0 -1.57 translation 0 0.3 -0.1
children [ Shape { geometry Cylinder { height 0.1 radius 1.3 }
appearance Appearance { material Material { diffuseColor 0 0.3
0.8 } } } ] } Transform { rotation 1 0 0 -1.57 translation 0
0.3 -0.1 children [ Shape { geometry Cylinder { height 0.08
radius 1.4 } appearance Appearance { material Material {
diffuseColor 0 0 0 emissiveColor 1 1 1 } } } ] } Transform {
translation 0 0.3 -0.1 children [ Shape { geometry Box { size
4 1.2 0.06 } appearance Appearance { material Material {
diffuseColor 0.8 0 0 } } } Shape { geometry Box { size 4.2 1.4
0.04 } appearance Appearance { material Material {
diffuseColor 0 0 0 emissiveColor 1 1 1 } } } ] } ] } ] }] }"),
xEasting, yNorthing, zAltitude, theText);
```

Table 5-22: On-the-fly development of 3D AuraSignpost objects

5.4.3 Immersion

Customisable Field of View

Having an easily customisable Field of View (FOV) is very practical when describing the world from an egocentric point of view (Papakonstantinou, 2005). In a virtual environment, it may happen that the user's viewpoint is sufficiently close to an object to make it lose its identity. An example that could describe this situation is found when a user navigates between tall buildings, with low visibility conditions. Equally, there may be occasions in which too much information is presented in 3D space, which may confuse the user (i.e. information overload). These problems can be resolved by introducing manageable FOV in the virtual environment, provided that it is intelligently accessed. Although that there can be a fully automatic way to implicitly control the FOV angle, by examining the distance between the device and the object that lies immediately in front of it, we did not test this feature in any evaluation of our system. The reason was that it might have overloaded the users with unfamiliar interactions that could distract them from the other goals of the survey. That is why we decided to embed this feature as a component that can be managed explicitly by the user through the user interface. Figure 5-10 presents the menu items of the VR interface which are used to adjust the VR camera's FOV. Figures 5-8 and 5-9 present two screenshots of *Aura's*

VR interface captured from an identical viewpoint with the same position and orientation parameters, but with different pitch in the viewing angles. In Figure 5-8, pitch is equal to 30°, whereas in Figure 5-9 the obtained pitch is equal to 90°. These figures present the difference in the volume of information that is contained in the virtual environment and how FOV is influenced by the different pitch angles. A viewpoint slightly wider than 60° (i.e. human eye) was found to provide ideal results. Once the user obtains the preferred position, no additional rotations are required in order to evaluate the surroundings lying in front of him or her.

Figure 5-8: Narrow-angle Field of View

Figure 5-9: Wide-angle Field of View

Interchangeable Perspectives of the Environment

A set of requirements, which have been described in Chapter 4.2.2, instructed the implementation of multiple easily interchangeable perspectives of the environment. It was particularly important to implement this functionality especially for the VR interface. The reason was that this interface is capable of supporting all required views of the environment in order to provide valuable visualisation perspectives, which can complement the cognitive map of the user and affect the user performance while executing a task. Furthermore, it was found that information overload and occlusion could be effectively minimised by implementing every view. Namely, the three perspectives, which describe the surrounding environment of the user, are the egocentric, the allocentric oblique and the allocentric plan views. The user may change the selected viewpoint at any time during system operation according to his preference, or according to the current task. Figure 5-10 presents the menu item of the VR interface which is used to iterate between the three available perspectives. In addition, users can set a default perspective, which will be loaded every time that the VR interface is selected. When the *egocentric* perspective is selected, the viewpoint is parallel to the ground and the height is accumulated from GPS data. The *allocentric oblique* perspective raises the viewpoint by 50 metres and produces an inclination of 45° towards the ground. *Birds-eye* (i.e. allocentric plan) view is particularly useful if the actor moves in urban canyons where 3D positioning (i.e. ≥3 GPS satellites are in line-of-sight) is not available (Papakonstantinou and Brujic-Okretic, 2009a). The class that enables the selection of an alternative perspective is *VrDlg* whereas *AuraContextManager* processes the administration of input variables. The source class implementing the presentation of various perspectives is *VrController* and, more specifically, its following functions.

```
•  short CVrController::MoveCamera(float xEasting, float
   yNorthing, float zAltitude);

•  short CVrController::RotateCamera(float inPitch, float
   inHeading, float inRoll, bool realHeight);

•  short CVrController::AddSubject(float xEasting, float
   yNorthing, float zAltitude, short subjectNo);

•  short CVrController::MoveSubject(float xEasting, float
   yNorthing, float zAltitude, short subjectNo);

•  short CVrController::RemoveSubject(short subjectNo);
```

Table 5-23: Aura's functions for altering the viewpoint and repositioning a subject

A subset of the aforementioned functions is called, according to the selected user view. Furthermore, in cases where the required information is not available, the fallback mechanisms are executed for presenting the best visualisations. For instance, when there is not a dedicated orientation sensor attached, only one parameter (i.e. heading) of *CVrController::RotateCamera()* is processed. The rest of the parameters are filled with constants which alter according to the selected perspective. Although this technique does not offer the highest level of immersion, because it draws information from backup sources (i.e. GPS), it is quite useful for the registration purposes. The following lists describe in sequence the process of presenting the view of the main actor according to the perspective that has been selected. The major difference between the three perspectives is the presentation of an avatar-like 3D object in the VR scene. When the egocentric perspective is selected, there is no need to illustrate an avatar because it is supposed to describe a first-person's point of view of the environment. In contrast, in the allocentric oblique perspective and the allocentric plan view, the VR engine must present an avatar, register it according to the latest location information, and move and/or rotate the camera so that it focuses on the subject. The presentation of the following views to the user takes place when the system operates in the *sensor-controlled* mode. In this mode, the user's context from the physical world is acquired by the sensors and fused to the system so that it is represented in the VR interface. The position of the VR camera in every perspective of the virtual environment follows the position of the user in the real world, whereas the heading of the VR camera in every perspective of the virtual environment is aligned with the heading of the user in the physical environment (i.e forward-up) (Aretz and Wickens, 1992).

Sensor-controlled mode - describing the *Egocentic* user perspective.

1. Retrieve location parameters from sensors (Output: Latitude, Longitude and Altitude);
2. Retrieve orientation parameters from sensors (Output: Heading, Pitch and Roll);
3. Move VR camera to new location (Input: Latitude, Longitude and Altitude);
4. Adjust orientation of VR camera (Input: Heading, Pitch and Roll).

Sensor-controlled mode - describing the *Allocentric Oblique* user perspective.

1. Retrieve location parameters from sensors (Output: Latitude, Longitude and Altitude);

2. Retrieve orientation parameters from sensors (Output: Heading, Pitch and Roll);
3. Move avatar to new location (Input: Latitude, Longitude and Altitude);
4. Move VR camera to new location (Input: Latitude, Longitude and Altitude+50m);
5. Adjust orientation of VR camera (Input: Heading, Pitch=45° and Roll).

Sensor-controlled mode - describing the *Allocentric Plan View* user perspective.

1. Retrieve location parameters from sensors (Output: Latitude, Longitude and Altitude);
2. Retrieve orientation parameters from sensors (Output: Heading, Pitch and Roll);
3. Move avatar to new location (Input: Latitude, Longitude and Altitude);
4. Move VR camera to new location (Input: Latitude, Longitude and Altitude+200m);
5. Adjust orientation of VR camera (Input: Heading, Pitch=90° and Roll).

5.4.4 Interactivity

Movement Functions

A technical characteristic that has been implemented in order to examine the performance and visualisation issues, which are related to movement in the virtual environment, reveals various ways of changing the position and orientation parameters. Three methods have been established and put to work. The first implementation relocates the camera to a precise position with the applicable heading, pitch and roll information, according to the acquired input coordinates (i.e. 6-DOF). Repositioning of the camera is instant. The second way is to create a new viewpoint and alter the perspective from the bound viewpoint to the new one. Viewpoints are VRML-specific nodes that represent a specific outlook of the VE. In this case, VRML code that contains the new coordinates for position and orientation must be accumulated and executed on the fly. The advantage of this method is that movement becomes smoother because a countdown timer specifies the duration of the viewer's motion from the current position to the new position in the scene. The last method for automatic interaction comes by moving all elements of the virtual world in order to conform to the new vista of the observer. Transition is more rewarding by utilising the second method, while the first one enhances the performance of the engine. Currently, the third method is not found ideal for any condition and poses a dramatic performance snag on the system operation.

The aforementioned interaction methods imply transformation of coordinates from a geographic format to a VRML format and vice versa. In VRML, the position of the viewer in the 3D scene is retrieved in the form of an array consisting of 12 elements. "*This array can be subdivided into four sections: origin, ortx, orty, ortz; each of this sections includes three elements (x, y, z). Origin is the position of the viewer; ortx, orty, and ortz are the projections of the viewer's direction on the x, y, and z axis respectively*" (ParallelGraphics, 2003). In order to specify a new value for a property, developers have to create a new VRML matrix object, set the values of the beholder's position and orientation according to the values of the translation and rotation properties of this object, and assign the value of the *Position* property of the VRML matrix object to the target property.

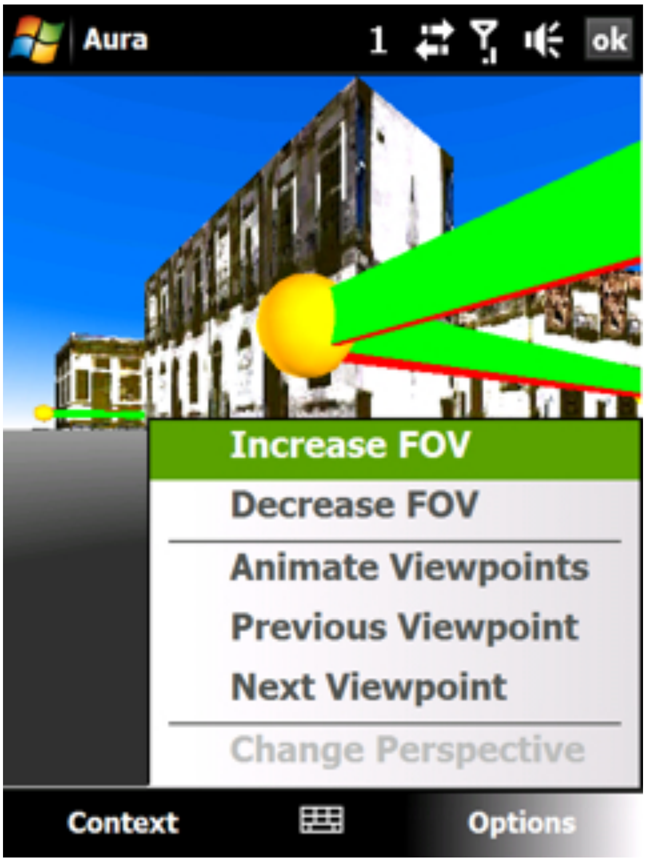


Figure 5-10: The Options Menu in the VR Interface

Orientation Functions

Real-time orientation is fused by two different sources. Primarily, continuous GPS coordinate retrieval can reveal the current heading of the actor. This functionality takes place by calculating former and current position updates, which expose the followed direction. We have observed that even though there was a minor latency, due to the fact that the system expects the completion of two or more iterations, the accuracy of this reading was sufficiently high. This is important because digital compasses, which are the other means to calculate the three orientation parameters, are still very rarely found in commercial products and are quite expensive to obtain for average consumers. Lately though, several mobile device manufacturers have started to embed sensors (e.g. 3-axis gyroscopes, accelerometers) that provide orientation information as a standard feature of their product. As a result, the accuracy of all orientation measurements has been improved and the polling rate is minimised to fully support the concurrency requirements of the framework.

In the VE, and more specifically in the sensor-controlled mode of operation, two mechanisms have been developed to make the user proficient to select the desired orientation of the camera. The first option presents the current user heading, which is constantly altered while moving, in order to simulate the natural behaviour. This way, the actors can observe through the VE the objects that are actually lying in front of them, which simplifies the registration process. The second option keeps the camera targeted on a specific direction that has been explicitly selected by the user and, while moving the camera, focus is maintained on that object. This helps the user to observe a specific feature of the environment, possibly another actor, while moving. This functionality can be manually enabled or disabled by the user through the menu of the VR interface. Figure 5-10 presents the menu of the VR interface, where the user can select the desired *FOV angle*, mentioned in Chapter 5.4.3, or select on which *Viewpoint* (i.e. POI) to lock the camera by iterating between the available ones. The *Animate Viewpoint* menu item starts an automatic iteration of all viewpoints making it possible to follow a virtual route, if a sequence of POIs has been loaded from a file.

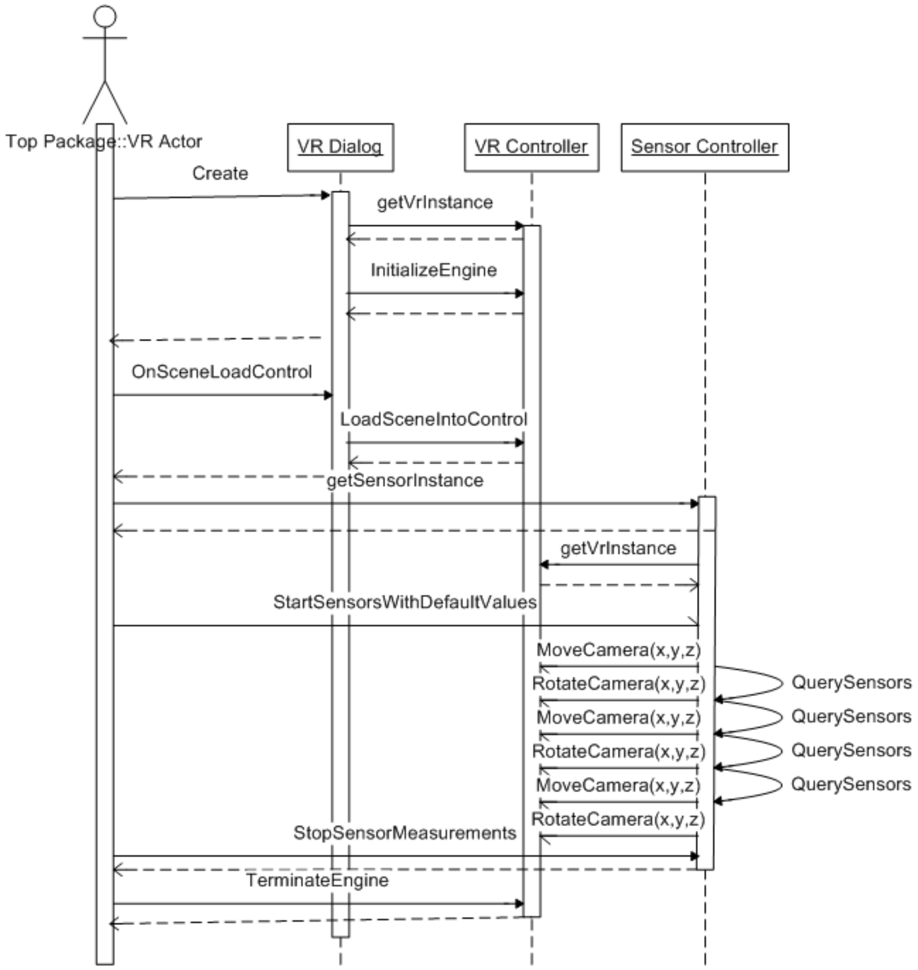


Figure 5-11: Sequence Diagram for sensor-controlled VR operation

The following code demonstrates how the user's orientation is represented in the VE, according to real-time data retrieved from the sensor (i.e. pitch, heading and roll).

```
// Get sin & cos values
float cosPitch = (float) cos(pitch);
float sinPitch = (float) sin(pitch);
float cosHeading = (float) cos(heading);
float sinHeading = (float) sin(heading);
float cosRoll = (float) cos(roll);
float sinRoll = (float) sin(roll);

// Set angles for pitch, heading and roll
pos.ortx.x = cosHeading * cosRoll;
pos.ortx.y = (-cosHeading) * sinRoll;
pos.ortx.z = sinHeading;
pos.orty.x = (sinPitch * sinHeading * cosRoll) + (cosPitch *
sinRoll);
pos.orty.y = ((-sinPitch) * sinHeading * sinRoll) + (cosPitch
```

```
* cosRoll);
pos.orty.z = (-sinPitch) * cosHeading;
pos.ortz.x = ((-cosPitch) * sinHeading * cosRoll) + (sinPitch
* sinRoll);
pos.ortz.y = (cosPitch * sinHeading * sinRoll) + (sinPitch *
cosRoll);
pos.ortz.z = ( cosPitch * cosHeading);
```

Table 5-24: Converting from orientation coordinates to VRML matrix coordinates

5.5 Augmented Reality Interface

In the Literature Review (see Chapter 2.4.2.1), we examined the most common approaches in order to retrieve the 6-DOF, which are required for registering the device both in virtual and real space. Each technique has certain advantages and disadvantages, and is considered optimal for different scenarios. Even in the first decade of 2000, the processing power of mobile devices had not reached consumer products for rapidly executing expensive calculations. AR needs very frequent updates in terms of context, as well as extensive processing times for transforming coordinate matrices, scene rendering and content authoring.

Having stored remotely, or even locally, a large number of patterns, which must be compared against natural features of interest, renders the process even lengthier. Moreover, there are not many physical elements that can be used as markers for AR implementations, therefore making traditional AR inefficient for truly ubiquitous operation. The marker-less solution that we have implemented needs accurate position and orientation context of the subject and the geographic coordinates of the remote entities, which are investigated. The perspective of this interface simulates the eye of the beholder and is presented on a live video stream.

5.5.1 Context-Sensitive AR

For ubiquitous service operation of such technology, a new approach is required. The Augmented Reality visualisation interface that was developed during the course of this project is capable of amalgamating the scene and selected surroundings that are currently observed by an actor with spatially referenced context descriptions, by utilising a mobile device and a number of context-sensitive sensors. Similar to the

Virtual Reality interface, for ubiquitous AR, the location and orientation parameters of the device constitute the essential information required by the context management layer (i.e. CMS). Furthermore, the proposed AR implementation assumes that the underlying engine holds knowledge about the spatiotemporal coordinates of specific objects or POIs in the real world. Being aware of the local spatial context and remote spatiotemporal context makes the system capable of introducing to the scene spatially referenced context descriptions by superimposing digital information on precise locations on the device display (Papakonstantinou and Brujic-Okretic, 2009a).

Apart from the camera, a GPS receiver and a digital compass must be present for achieving the required functionality. The retrieved measurements have proven to be sufficiently accurate for representing features of interest of the real environment, in an estimated location relative to the device's actual orientation and position. The frequency of updates and the small derivation from the genuine location poses limitations, when the requirements instruct for advanced functionality, such as the alignment of an alternative texture on the façade of a building in the real-world scene. That functionality was explored, by Reitmayr and Drummond, by querying additional sources of context, which in our case are not available because a goal of this research has been to operate the application with minimum standardised configurations available for commercial exploitation (Reitmayr and Drummond, 2006).

Figure 5-12: The AR interface, while detecting a Point of Interest

The aim of *Aura's* AR interface is to augment a live camera feed of the real world scene with virtual information in real time, achieving registration between the virtual and real, while providing a convincing Mixed Reality experience for the users of the system. Most virtual information has been confined to textual annotations and descriptive 2D symbols providing information about the user's surroundings or further navigational assistance. Chapter 5.5.6.3 examines how virtual 3D objects can be rendered in the real world scene. The combination of sensor context with this video see-through solution is efficient, which shows that there is no need for HMD utilisation to achieve a working see-through AR prototype.

Hand gestures are used to interact in the real world. The user can pan the device around in order to interrelate with the environment and retrieve information about objects that exist in it. The type of feedback that is gathered depends solely on the applicable scenario. For locating subsequent objectives in navigational scenarios, directional aids are applied. These include textual descriptions about the target, distance calculations, and arrow representations that point towards the next objective. For exploration scenarios, information about entities in close proximity is offered. Self-explanatory symbols and textual descriptions allow users to comprehend and decide which is going to be the next place that they will visit, depending upon their preferences (e.g. hotel, casino). In any pervasive scenario, directed guidance is provided towards the next goal of the scenario. This may be either locating an object or person, reaching a checkpoint or unlocking the next level of interaction and visualisation functionalities.

It was found beneficial for the decision-making process of an actor to combine real and artificial information about the environment. Observing the surroundings from an egocentric perspective allows the actor to make better informed choices, taking under consideration measurable phenomena and quantifiable criteria that may not have been considered before. For instance, consider a user who is located in an unfamiliar location and seeks a train station. Two relevant candidates appear on the screen but in opposite directions. It is then up to the user to decide which one to visit, based upon their proximity and other information that is included on file about this entry. Subsequently, if there is a relevant web link stored in the entry about the candidate, the user may wish to visit the electronic address directly through the 3G connection of the device in order to gather additional information, such as the list of underground lines, which are served by that station. We have observed that several users of our system require information about their position in the real world as well as about the position of other objects and actors (*Position* queries). Additional required functionalities include: finding the *Nearest* POIs, *Range* queries and *Visualisation* and *Interaction* methods with the virtual content.

5.5.2 Tracking the User's Pose

There are several requirements that need to be fulfilled in order to provide mobile and ubiquitous AR services. The functional goal, though, is the estimation of the pose of the device according to a point in the real world, by processing the 6-DOF. In order to achieve that goal, the system has to react rapidly to all context changes and compute the new data that is generated by the sensors. The process, which includes a large number of transformations between *World*, *Device* and *Image* coordinate systems, should work with the minimum amount of resources. In the case of mobile device operation, this may prove challenging because earlier mobile graphic processors did not include native floating point processing and all measurements must have been made through fixed point arithmetics. Furthermore, continuously querying the storage layer of the system for location information about relevant POIs may prove a real burden, in terms of performance. Synchronisation of all sensory input, as well as real-time retrieval of data generated by either the local or remote entities, need to be efficiently managed, which occurs under the explicit control of the higher application layers. Finally, a live video stream is required in order to be enhanced and presented to the user, in a concurrent sequence. The previous observations show that in order to achieve the functional goal several technical constraints must be accommodated in advance.

The process that we describe in this section, which is applied in order to accomplish pose tracking, was found to be competent in terms of performance, resource-efficiency and reliability. It can be used to represent features of interest, of the real environment, in an estimated location relative to the device's actual orientation and position. Although calculating the pose in the real world by examining a single point in space is not efficient for advanced AR operation, as described in a previous paragraph, it has proven to be adequate for the objectives of this project. The reason is that camera resolutions, environmental conditions and quality of natural features or markers do not directly affect the efficiency of the interface when working outdoors (Papakonstantinou and

Brujic-Okretic, 2009a). In contrast, up-to-date context distribution of local and remote entities is vital. This makes our solution robust and ready to operate in unknown environments. Ultimately, we aim to examine pose tracking based on a sum of points. The next figure shows how location context of remote objects is utilised by the AR interface.

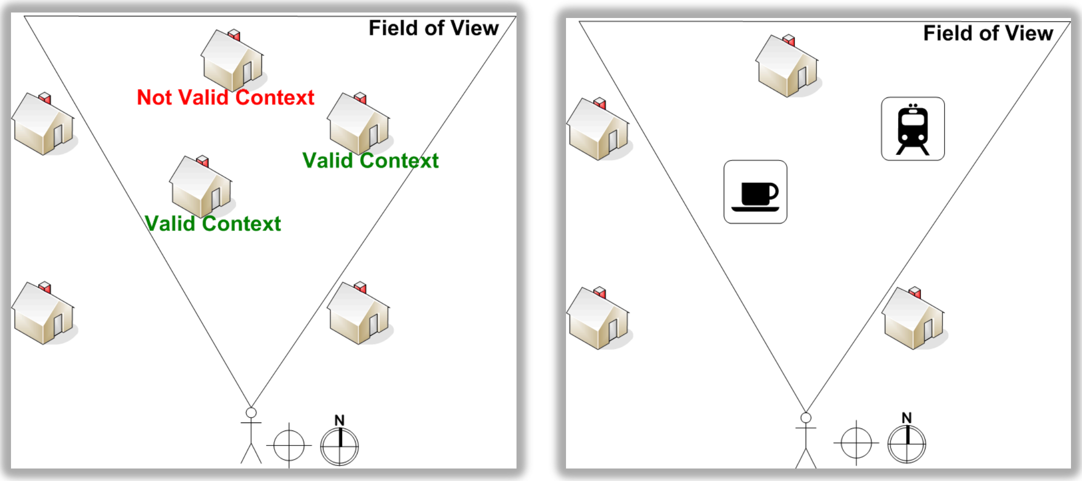


Figure 5-13: The use of AR - Contextualising location & orientation information

In the following sections the reader can find more about the components of the process that enables the estimation of the actual physical pose in the environment, and the mathematic formulas, which have been implemented in order to achieve the required functionality. The specified implementation can be easily ported on a variety of mobile devices because the interface is decoupled from the functionality and the only hardware requirement is the physical existence of the sensors (i.e. GPS, digital compass and camera). To prove it, we implemented the AR interface in two versions. The first was embedded in a standalone application, *Aura*, and the second has the form of a reusable ActiveX component, *ARIE* (Augmented Reality for Internet Explorer), which can be integrated to diverse host applications (e.g. mobile web browser). *ARIE* has been embedded and extensively tested as the AR interface of LOCUS. Both solutions enable the incorporation of graphical information with the media-streaming layer. The valid use case flow for delivering mobile and ubiquitous AR functionality is presented below and subsequently analysed:

1. Retrieve position and orientation for local entity. (Tracking)
2. Retrieve position (and orientation) for remote entity. (Tracking)
3. Calculate point coordinates of Field-of-View polygon.

4. Check if remote entity lies inside the FOV boundaries.
5. Calculate distance between local and remote entities.
6. Calculate coordinates for tangent side of FOV polygon.
7. Map the tangent side on the device display.
8. Calculate the image coordinates for the remote entity. (Registering)
9. Make the transformations from world to camera and from camera to image coordinate systems. (Camera Modelling)
10. Superimpose text, image, or a 3D model over the specified pixels on the device display. (Rendering)

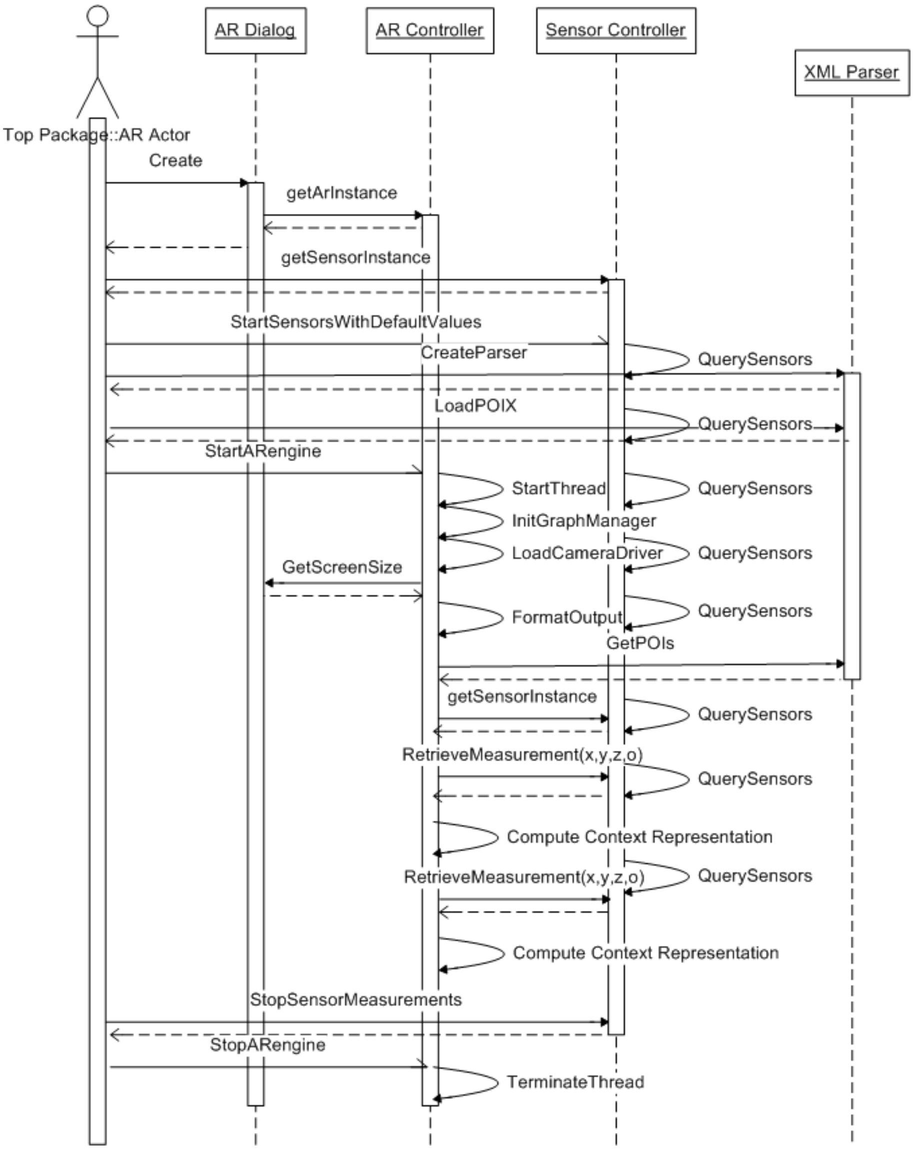


Figure 5-14: Sequence diagram describing the operation of the AR Interface

With nominal positional and directional accuracy, digital content can be overlaid in the real-world scene in real time. On the contrary, with almost absolute accuracy, synthetic content can be superimposed precisely over a physical object (e.g. door, person) on the device screen (Papakonstantinou and Brujic-Okretic, 2009a). There are certain cases in which the user may not want to point the camera directly over the target, but still needs to retrieve the descriptive information about the entity. In such instances, only the *yaw* variable is calculated from the compass while pitch and roll are discarded. Altitude from GPS readings is similarly not processed.

The spatiotemporal coordinates of a POI in the real world may be retrieved either through a network resource or the persistent storage. Extraction from the latter requires querying the persistent storage of the device about the remote entity or parsing a datafile (e.g. GPX) from non-volatile memory. Utilising a network resource is mostly concerned with moving objects such as people, whereas local information retrieval examines motionless entities such as shops or buildings.

5.5.3 Field-of-View Polygon Calculation

Synchronising GPS and compass measurements is a crucial factor for smooth operation. Knowing the device position and the current user's viewpoint allows the framework to calculate every element of the virtual polygon, which simulates the actor's field of view. The coordinates of points *B* and *C* can be retrieved in real time after applying certain trigonometric algorithms, as illustrated in the following figure. The mobile device position is represented by point *A* and its direction by vector *AD*. Subsequently, we have to map the left-most pixel of the video stream with point *B* and, similarly, the right-most pixel with point *C*. In the current implementation, the size of *AB* and *AC* is equal to a predefined value (i.e. 50m), which can be easily altered from the UI. Changing this constant will provide greater accuracy when reduced, and least when maximised. The difference, in terms of functionality, is that the framework can query entities lying at variable distances according to the user need.

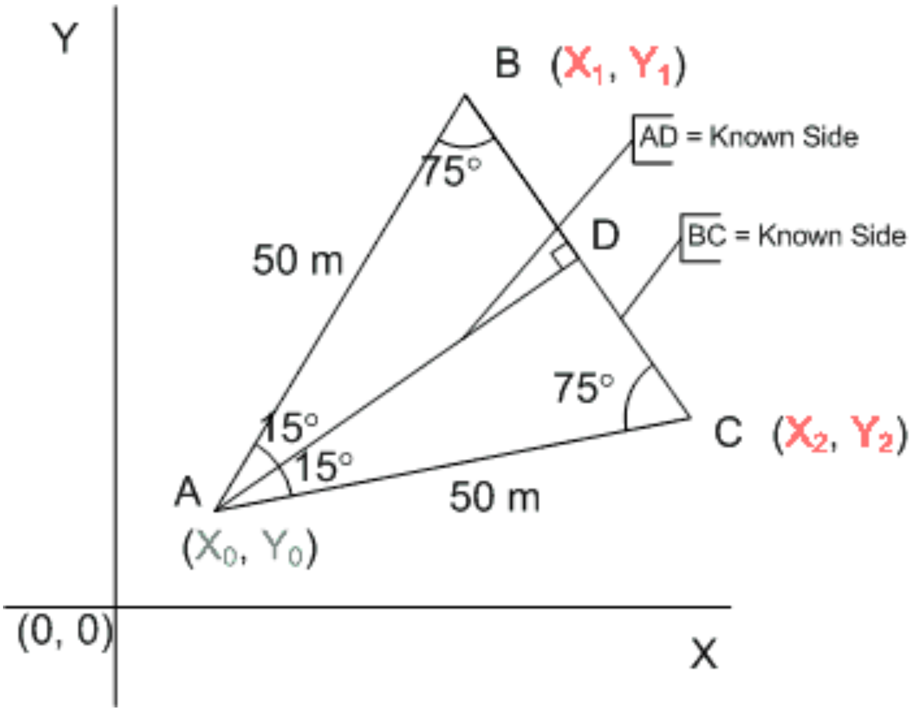


Figure 5-15: Field-of-View polygon under investigation

There are many different conceptual ways to implement this process and each way has various implementations in terms of the required algorithms. The author developed two different algorithms. The first algorithm was used in LOCUS, whereas the second and more detailed was used in the implementation of Aura. Each solution offered certain advantages. The calculation speed was higher in LOCUS version, whereas accuracy was better in the latter version due to the requirements of each application. In LOCUS, when ARIE was used, an angle deviation from North was being calculated. This approach operated for infinite distance, which means that exact pinpointing of an object on the screen of the device was not possible. The LOCUS solution could only identify an object and present information about it on the device screen. This presentation did not include any spatial relation between the actual object and the one represented on the device screen. That approach was developed by the author for the requirements of the LOCUS project and was considered as the initial step towards future research in this field. Furthermore, the algebraic approach did not prove efficient because there are several cases in which the calculated polygon lay in different quadrants and user orientation did not enable calculation through a unified solution.

The use of calculus in the Cartesian coordinate system provided the solution for calculating the geographic coordinates of the two unknown points (i.e. *B* and *C*), which are needed in order to accumulate all parameters of the polygon. This is the second alternative which was developed for satisfying the requirements of the project.

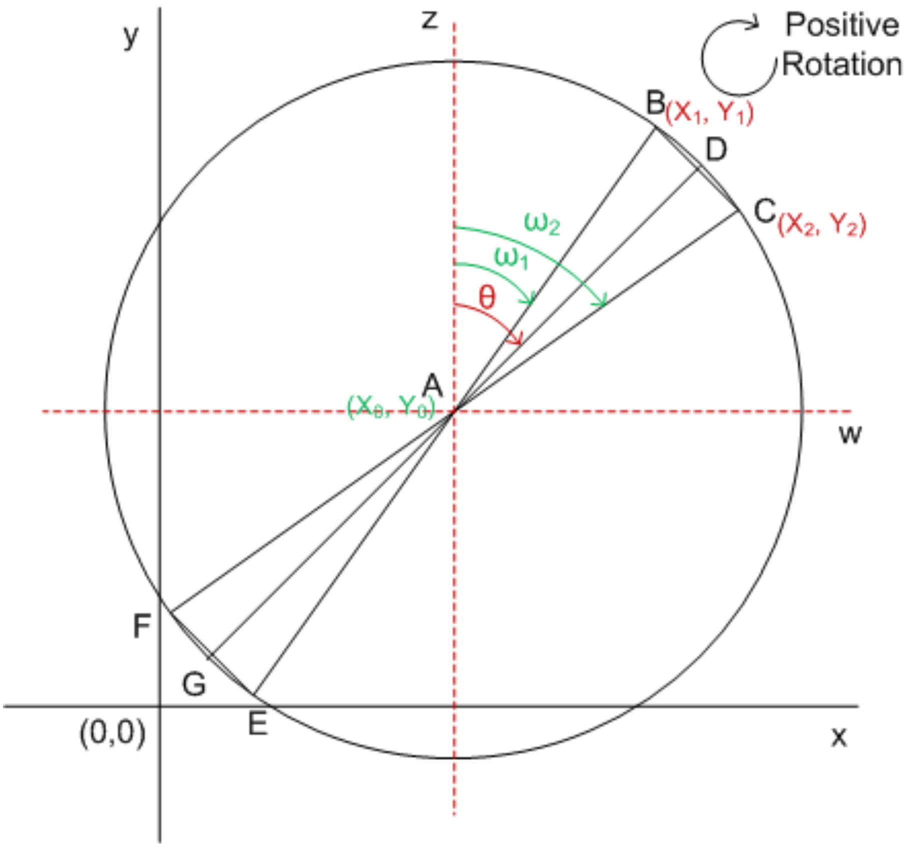


Figure 5-16: Line & Circle intersection sketch

We define angle θ of a given direction AD on the Cartesian coordinate system, the one that emerges by positively rotating the *y-axis* until it is absolutely aligned with line AD, which is the given yaw/heading of the device. Note that θ should be more than 0° and less than 180°.

We define the coefficient of direction λ of a given line AD, as the tangent of angle θ.

($\lambda = tan\theta$). Note that:

- for $0° < \theta < 90°$, $\lambda > 0$
- for $90° < \theta < 180°$, $\lambda < 0$
- for $\theta = 90°$, $\lambda \rightarrow \infty$

The equation of a line for which we know its λ and a point $A(x_0, y_0)$ that overlaps with it, is given by the following formula:

$$y = \lambda x + c$$

Equation 1

and intercept *c* is a point on the line, which intersects with the *y-axis* and can be found by solving the equation of a line:

$$c = y_0 - \lambda x_0$$

Equation 2

The equation of a line for which we know the coefficient of direction λ and two points $A(x_0, y_0)$ and $D(x, y)$, is given by the following formula:

$$y = \lambda(x - x_0) + y_0$$

Equation 3

The equation of a circle with a known centre $A(x_0, y_0)$ and a given radius *r* is given by the equation:

$$x^2 + y^2 = r^2$$

Equation 4

Similarly, the equation of a circle with a known centre $A(x_0, y_0)$, an overlapping point $D(x, y)$ and a given radius *r*, is given by the equation:

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$

Equation 5

The problem that we face is defined as follows.

- Direction of *AD*, the line of sight, or angle θ, is known by retrieving orientation context from the digital compass.
- The coordinates of a point $A(x_0, y_0)$, in the world coordinate system, is known by retrieving position context from the GPS receiver.
- The length of *AB*, *AC*, *AD* and *r* are equal to a manageable constant (e.g. 50m).
- The angles of BAD and DAC are equal to a predefined constant (i.e. camera's focal length divided by 2).

To calculate the field of view polygon, the required outputs are the geographic coordinates of points $B(x_1, y_1)$, $C(x_2, y_2)$ and $D(x_3, y_3)$. To find the intersection between lines *AB*, *AC* and *AD* the following process has to work out.

The substitution of y in Equation 5, with $\lambda x+c$ from Equation 1 leads to:

$$(x - x_0)^2 + (\lambda x + c - y_0)^2 = r^2$$

Equation 6

After expanding and grouping the terms we get:

$$(1 + \lambda^2)x^2 + (2\lambda c - 2\lambda y_0 - 2x_0)x + (x_0 + y_0 + c^2 - 2cy_0 - r^2) = 0$$

Equation 7

In Equation 7, if we set:

- $a = 1 + \lambda^2$
- $b = 2\lambda c - 2\lambda y_0 - 2x_0$
- $c = x_0 + y_0 + c^2 - 2cy_0 - r^2$

We observe that equation 7 is a quadratic formula conforming to the following format:

$$ax^2 + bx + c = 0$$

Equation 8

As a result, we get 2 solutions for obtaining x:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Equation 9

And

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Equation 10

Similarly, there are two results for y, when x is substituted in Equation 3 with the results of Equations 9 and 10. This process has to be implemented three times in order to get the coordinates of points B, C and D. This can be achieved by replacing θ with ω_1 and ω_2, which are equal to $\pm \frac{\text{focal length}}{2}$.

Apart from those points, we also get the coordinates of points E, F and G, which we need to automatically discriminate and leave out of the final calculations. This is accomplished with the following comparisons:

- if $0° < \theta < 180°$, for all substitutions equation 9 is used
- if $180° < \theta < 360°$, for all substitutions equation 10 is used

The source code, which implements the previous series of equations and calculates the FOV polygon, is provided in Appendix VII.

5.5.4 Point in Polygon & Polygon in Polygon Algorithms

The next step is to calculate if a point, which represents the coordinates of a remote entity, lies inside the boundaries of a given polygon, by implementing a Point-in-Polygon or Polygon-in-Polygon theorem. Following this procedure, the system can superimpose text annotations over real objects. In the AR interface, interaction is triggered by the sensors when the actor is moving and/or performing gestures in the physical world.

There are numerous open source algorithms available for determining if a point lies inside the boundaries of a polygon. Their functionality can be found in GIS, web and, in this case, in computer graphics applications. The most widely used implementation was developed by Eric Haines (Haines, 1994) and is being used in the Apache web server (Apache, 2011b) for checking if the mouse pointer is inside the boundaries of a picture on a web page, so that additional functionality may be reproduced (e.g. displaying the *ALTernative* text of the image). There are many conceptual approaches that support this functionality. Namely, a few of them include MacMartin's solution, Crossings-count, Triangle Fan (w/o Edge Sort), Barycentric and Angle summation. An extensive comparison of the efficiency between these solutions can be found in (Haines, 1994).

For the purpose of this research, the author decided to develop a custom algorithm, which implemented the Crossings-count method. The simplest way to achieve that functionality was by applying the *Jordan Curve Theorem*. This technique evaluates that a point exists inside a polygon by sending a ray, starting from the origins of the point and extending towards infinity. If the number of times that the ray crosses the polygon edges is an odd integer, then the point lies inside the boundaries. If the number is even, then the point is considered to be outside of the polygon. There are a few cases in more complex polygons in which this algorithm has proven not to be suitable. In contrast, for a simple pyramid-like polygon, our case, calculation has proven quite efficient.

	Galacticomm inpoly()	**Apache pointinpoly()**	**CERN/ W3C insidePoly()**	**GE CRD MTL inPolygon()**	**Woods Hole inside()**	**Aura PIP()**
Authors	Bob Stein, Craig Yap	Eric Haines, Rob McCool	Ari Luotonen	Kevin Kenny	Rich Pawlowicz	Stelios Papakonstantinou
Language	C	C	C	Tcl	Matlab	C++
Data type	unsigned int	double	int	(native)	(native)	double
Computation type	long	double	float, double	(native)	(native)	double
Method	crossings	crossings	crossings	crossings	angle sum	crossings
Lines of code	44	72	91	29	89	33
Loops	1	4	1	1	0?	1
If-statements	3	10	19	2	5	5
Else-statements	1	4	11	1	1	3

Multiplies	2	3	1	1	20	2
Divides	0	3	1	1	2	1
Arctangent	0	0	0	0	1	0

Table 5-25: Comparison of Point in Polygon algorithms (Stein, 1997)

The source code, which implements the point in polygon algorithm used in *Aura*, is provided in Appendix VIII.

5.5.5 Camera Modelling Algorithm

This section examines how we can relate the coordinates of pixels on the device screen with the geographic coordinates of a Frame of Reference (FOR) from the real world. This procedure is known as *camera calibration* and is separated into two phases. The first step is called camera modelling and copes with the mathematical approximation of the physical and optical behaviour of the sensor by using a set of parameters (Salvi et al., 2002). This step tries to relate the camera coordinate frame to an arbitrary coordinate frame, fixed to a given point in the real world. The second step tries to estimate the values of the parameters used in the first step. This way, we can establish a relation between the image coordinates and a coordinate frame aligned to the camera. These transformations depend on two sets of parameters. *Extrinsic* parameters are used to calculate the position and orientation of the camera with respect to the world coordinate system. In contrast, *intrinsic* parameters relate the camera coordinate system to the image coordinate system, by determining how light is projected through the lens onto the image plane of the sensor. A good reference to these points has been provided by Morris (Morris, 2004).

Research in *Computer Vision* and *Pattern Recognition* has produced various calibration methods, which can be classified according to diverse factors.

- Linear and non-linear
- Intrinsic and extrinsic
- Implicit and explicit

The framework requires the internal parameters of the camera lens because we need to get the screen coordinates of a point that exists in the real world, in relation to the actual orientation and position of the camera that captures it. Camera modelling is concerned with the approximation of the internal geometry along with the orientation and position of the camera in the real world scene. *Non-linear* models accomplish accurate modelling of the lens parameters, whereas *linear* transformations do not model the distortion of the lens. Thus non-linear approaches are far more accurate and applied to applications with the requirement of maximum precision (Salvi et al., 2002). The simplest approach is considered to be Hall's (Hall et al., 1982) but generally all linear transformations, including Tsai's (Tsai, 1987), follow the same four-phase procedure.

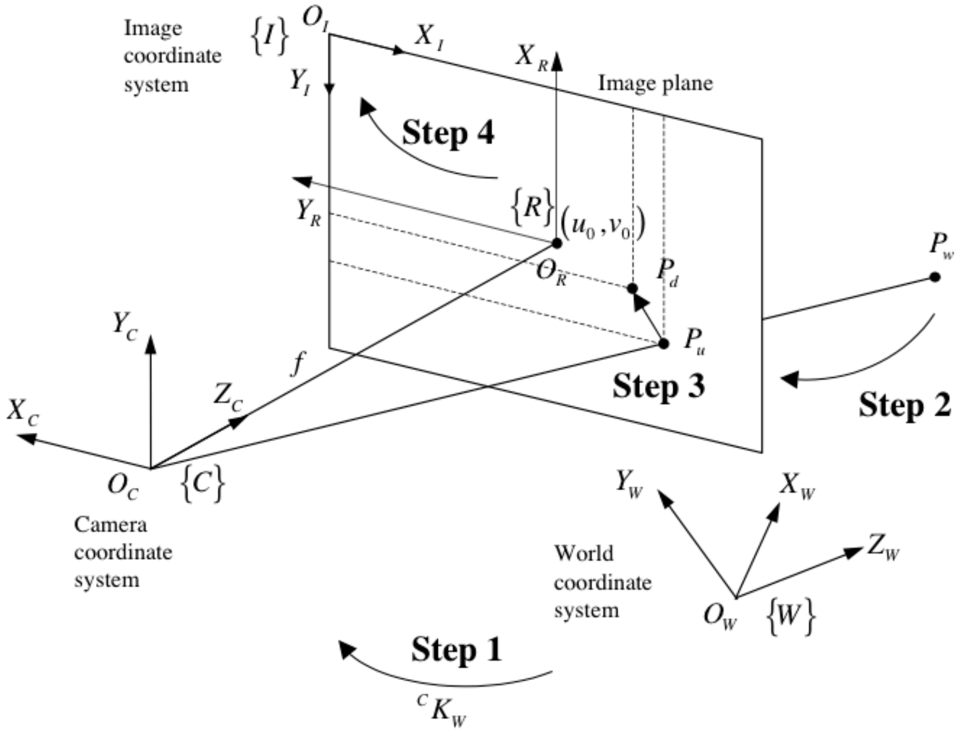


Figure 5-17: Geometric relation between 3D point and its 2D image projection (Salvi et al., 2002)

In the first step we have to retrieve point ${}^{C}P_{W}$ in the camera coordinate system, by relating it to point ${}^{W}P_{W}$ from the world coordinate system. To accomplish that, we have used a rotation matrix and a translation vector. Following next, we need to calculate the projection of point ${}^{C}P_{W}$ on the image plane and find point ${}^{C}P_{U}$, by using a projective transformation. In the third step, point ${}^{C}P_{U}$ is transformed to the real projection of ${}^{C}P_{D}$. In this step, radial distortion of the lens is calculated. Finally, in the fourth step we have to do another coordinate system transformation for changing from the camera coordinate system to the image coordinate system. This way, we can retrieve point ${}^{I}P_{D}$, which represents the coordinates of pixels on the device screen. The following process has been published, in the context of the whole AR solution, in the *International Journal of Computer Graphics* (Papakonstantinou and Brujic-Okretic, 2009a) and intends to be used in ubiquitous real-time systems that require context reactivity.

Step 1

Inputs for the first step are the coordinates (*x, y, z*) of a POI from the real world. The transformation from the world coordinate system of the given point to the 3D camera coordinate system is accomplished by executing the following equation. The transformation is modelled by utilising the rotation matrix $\mathbf{R}_{\mathbf{W}}^{\mathbf{C}}$ and the translation vector $\mathbf{T}_{\mathbf{W}}^{\mathbf{C}}$.

$$\begin{pmatrix} \mathbf{X}_{\mathbf{W}}^{\mathbf{C}} \\ \mathbf{Y}_{\mathbf{W}}^{\mathbf{C}} \\ \mathbf{Z}_{\mathbf{W}}^{\mathbf{C}} \end{pmatrix} = \mathbf{R}_{\mathbf{W}}^{\mathbf{C}} \begin{pmatrix} \mathbf{X}_{\mathbf{W}}^{\mathbf{W}} \\ \mathbf{Y}_{\mathbf{W}}^{\mathbf{W}} \\ \mathbf{Z}_{\mathbf{W}}^{\mathbf{W}} \end{pmatrix} + \mathbf{T}_{\mathbf{W}}^{\mathbf{C}}$$

Equation 11

Step 2

For the projection of the 3D point on the image plane, we have to consider the optical sensor as a pinhole camera. This means that the image plane is located at a distance *f* from the optical centre O_C and is parallel to the plane defined by the coordinate axis X_C and Y_C. If an object point $\mathbf{P}_{\mathbf{W}}^{\mathbf{C}}$ in the camera coordinate system is projected through the focal point O_{C}, the optical ray intercepts the image plane at the 2D image point $\mathbf{P}_{\mathbf{U}}^{\mathbf{C}}$. This is presented in the following equations. Note that commercial camera manufacturers provide the focal length with the product specifications, which is not the case for mobile phone cameras. This means that it has to be calculated by employing supplemental software (Vezhnevets and Velizhev, 2005), which examines pictures captured by the on-board camera.

$$X_U^C = f\frac{X_W^C}{Z_W^C}$$

Equation 12

And

$$X_U^C = f\frac{X_W^C}{Z_W^C}$$

Equation 13

Step 3

The third step is applied for modelling the distortion of the lens. The primary reason that causes radial distortion is the potentially flawed radial curvature of the camera lens. In the following equation we can see the transformation from the undistorted point $\mathbf{P_U^C}$ to the distorted point $\mathbf{P_D^C}$, where δ_x and δ_y represent the involved distortion.

$$X_U^C = X_D^C + \delta_x$$

Equation 14

And

$$Y_U^C = Y_D^C + \delta_Y$$

Equation 15

Although there are two kinds of potential distortion factors, *radial* and *tangential*, it has been noted (Tsai, 1987) (Salvi et al., 1998) that only radial has to be taken under consideration because modelling both may produce numerical instabilities during the calculations. The displacement given by radial distortion δ_r can be modelled in the following equation, which considers only k_1.

$$\delta_{xr} = k_{\mathbf{1}} X_D^C \left(cX_D^{\mathbf{2}} + cY_D^{\mathbf{2}} \right)$$

Equation 16

And

$$\delta_{yr} = k_{\mathbf{1}} Y_D^C \left(cX_D^{\mathbf{2}} + cY_D^{\mathbf{2}} \right)$$

Equation 17

Step 4

The last step deals with the change from the camera image to the screen image coordinate system. This is accomplished by conveying point $\mathbf{P_D^C}$ with respect to the screen image plane, which is constituted by pixels. The next two equations explain how to accomplish this transformation.

$$X_D^I = -k_u X_D^C + u_{\mathbf{0}}$$

Equation 18

And

$$Y_D^I = -k_v X_D^C + v_{\mathbf{0}}$$

Equation 19

Parameters k_u and k_v make the transformation from metric measures in the camera coordinate system, to pixel in the screen image coordinate system. Parameters u_0 and v_0 are those that define the projection of the focal point in the plane image in pixels. The translation between the two coordinates systems depends on their value

5.5.6 Scene Rendering

After finding the screen coordinates of a real-world point, the system must overlay distinct visual effects, on top of the video stream in relation to the identified point. Thus, the AR component of *Aura* is separated in two parts. The first one is concerned with tracking the 6-DOF of the user's device and it was described in the previous section. The second part, which is equally important because it presents the results of the aforementioned process to the user, is applied for rendering the video stream and for superimposing the additional context representations.

During the design phase, the technological framework was selected according to compatibility and performance issues. The functionality required for establishing the AR interface is presented below.

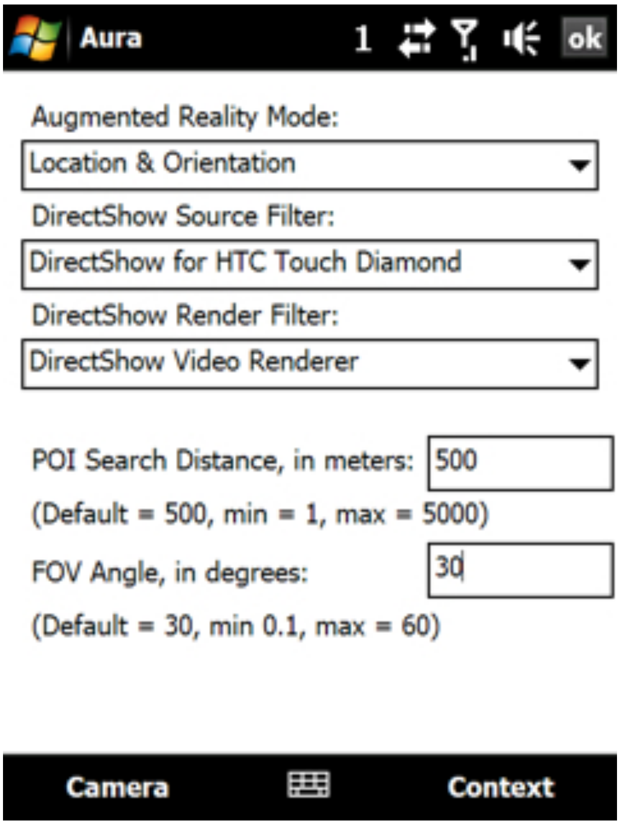
1. Creating an information visualisation platform that runs on a mobile device;
2. Displaying live video feed from the mobile device camera;
3. Superimposing/Rendering text and/or graphics over the video feed;
4. Updating text and/or graphics periodically, according to real-time context.

The requirements directed smooth operation of the application in an environment that is based on mobile devices for the hardware layer and on the Windows Mobile for the software layer. The selection of this platform enabled further development with a variety of options and methodology considerations. The theoretical research produced five distinct ways that have the potential to achieve the required functionality. Each method has its own special characteristics and drawbacks. The potential architecture options include the following.

1. A local media server component that can handle the camera input and publish the result on a local IP (e.g. 127.0.0.1);
2. A remote server that would receive the camera feed, enhance it with additional information and push it back to the device;
3. A J2ME low-level camera driver that would take advantage of the Mobile Media API (MMAPI, JSR-135) and Wireless Messaging API 2.0 (WMA, JSR-205) and present the result on a Java Applet;
4. A standalone application, which would handle the camera feed through DirectShow and use standardised libraries to visualise supplementary elements;
5. A reusable component (i.e. ActiveX) that may be embedded in any host application, like a mobile web browser (e.g. Pocket IE), on top of the OS-delivered, DirectShow libraries.

An additional requirement, which has been introduced in the system, instructs the development of the rendering component in such a way that it can be used to implement the required functionality in standalone solutions, as well as to be compatible with the currently evolving web technologies and possible other applications. That is why the author developed two solutions, which share some functionality, but also have distinct differences. These solutions are options 4 and 5, which were described in the previous list. The other options were discarded, mainly because of the performance drawbacks that they could present during system operation and because of the compatibility issues that would probably emerge with the technologies that had already been utilised for the development of the other framework components. The standalone application has been integrated to *Aura* and offers more advanced functionalities compared to the other one. The second implemented option produced a reusable component, which is called ARIE. For maximum compatibility and web accessibility, the underlying host application has been selected based on standard mobile browser applications (e.g. Pocket Internet Explorer, Opera) but it can also work with any applications that support ActiveX controls. ARIE was integrated to the LOCUS system in order to handle all visualisation and interaction requirements of its AR interface.

The previous paragraph mentioned that both solutions have similarities and differences in terms of development methodologies and utilised technologies. The major similarity of both solutions is the introduction of *DirectShow Mobile* API (Microsoft, 2010), which is capable of handling the input stream of the camera and present it on the device screen. In contrast, the major difference between the two rendering solutions is the technology that has been put to work in order to represent descriptions regarding context entities, which exist in the real world. Namely, both solutions use low-level Windows Mobile's Graphics Device Interface (GDI) functions to draw text or simple 2D representative graphics on the video stream, but in the case of the standalone solution worthwhile attempts have been made to introduce 3D graphic illustrations as well. GDI has proven compatible with all tested devices, efficient and very fast, so it was chosen as the default native drawing API for the implementation of the prototype and during the debugging process of both developed solutions.



Augmented Reality Mode:
Location & Orientation
DirectShow Source Filter:
DirectShow for HTC Touch Diamond
DirectShow Render Filter:
DirectShow Video Renderer

POI Search Distance, in meters: 500
(Default = 500, min = 1, max = 5000)
FOV Angle, in degrees: 30
(Default = 30, min 0.1, max = 60)

Figure 5-18: Aura's interface for accessing context-sensitive AR

The selected development environment offers two ways to advance with the implementation of the video-handling component: through the use of Microsoft Foundation Classes (MFC) or by using the Active Template Library (ATL). ATL has been selected for the development of ARIE because it produces components with small memory footprint that run faster on mobile devices with limited resources. To use an ActiveX control in an application, the run-time control (i.e. UUID) must be registered on the mobile device either manually or automatically. The security model of Windows Mobile 5.*x* onwards, instructs that dynamically linked libraries (i.e. DLL) have to be signed and validated against certificates in the *privileged* or *unprivileged* certificate store of the device. This is accomplished by building the output with the selected certificate on every post-compilation step. If the provision is set to a non-default certificate store, the specific certificate is downloaded to the device on every deployment. In contrast to ARIE, the development of the standalone solution was based on the MFC library, because it offered advanced options and the library had been already loaded on the application repository, therefore constituting ATL redundant.

5.5.6.1 Handling the Video Stream

The process of acquiring and presenting video content on the mobile device display is the same for both the standalone solution and the ARIE ActiveX component. Using the portable drivers associated with DirectShow, rather than lower-level drivers associated with specific graphic chipsets, provides a higher degree of portability between devices.

Chapter 4 described the advantages of using the DirectShow libraries for retrieving video content. Although, it is the best solution for the Windows Mobile platform, the available filters, which are the building block of any DirectShow application, are not the same as in the desktop equivalent library. Microsoft has not developed a large number of useful filters in versions 5.*x* to 6.*x* of their mobile OS. This means that it is either up to the mobile device manufacturers or the application developers to implement those filters. This made it exceptionally complicated to handle the video stream of the mobile camera, in comparison to desktop applications. The reason is that not every Windows Mobile device supports the same pixel format of video frames for capturing as well as rendering them. The most common format for Windows Mobile devices is considered to be RGB565, which supports 16-bit colours per pixel. As a result, different handlers have been developed, according to the input and output formats. These handlers can process video content from various sources and different format types, therefore making the AR subsystem compatible with most currently available devices. These handlers can support the following video stream sources, depending on the device capabilities.

- Generic DirectShow-compatible *Filter Graph*, for most devices;
- Limited DirectShow-compatible *Filter Graph*, for devices with limited recording and rendering capabilities (e.g. i-Mate JasJar);
- Full DirectShow-compatible *Filter Graph*, for the latest devices with advanced recording and rendering capabilities (e.g. HTC Touch Diamond).

The series of filters that are interconnected in order to perform a specific task is called a *Filter Graph*. The application communicates with DirectShow via the *Filter Graph Manager*. This is a high-level software component, which controls the data flow between the filters. The *Filter Graph Manager* produces events that the application has to respond to and receives commands (e.g. play, stop) that should be executed. Due to the nature of the application, synchronisation between the internal processes and the external user interactions needs to be achieved. The way to realise this functionality is through the use of threads that could manage the processing time for each request. In order to build a *Filter Graph,* two methods are provided; the automatic way, which connects all the required filters for rendering a video stream, and the manual way that allows the developers to connect the output pins of each filter with the appropriate input pins of the following one. The latter method needs explicit negotiation of the format that is being exchanged according to the format supported by each pin. The following diagram shows, which are the core DirectShow filters that are required for operating the AR interface.

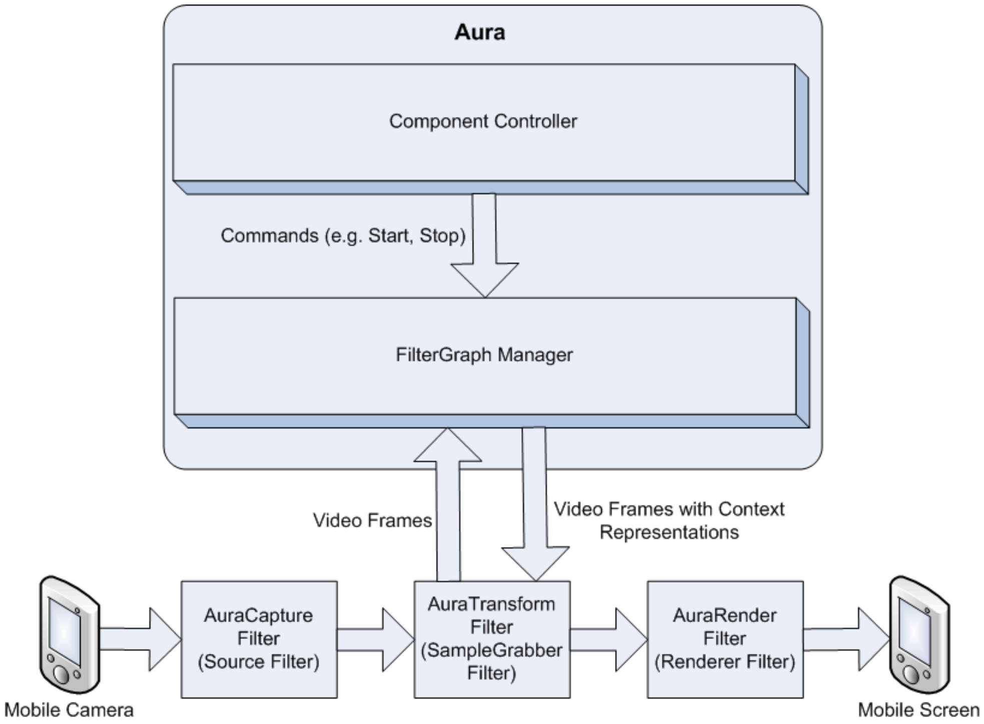


Figure 5-19: Aura's Filter Graph

The automatic connection method provided by DirectShow, in order to join the required filters of a *Filter Graph*, is useful only in a few cases, such as when there is a need to capture a video stream and, either save it on a file, or present it on the display. In our case, that it needs to acquire, examine and enhance each frame that is produced by the camera, we need to apply the manual filter connection method. The reason is that we introduce brand new filters, which are not offered by any API. The author has exclusively developed these filters. For the purpose of this research and the development of a working AR solution, three distinct filters have been implemented, which work on Windows-enabled mobile devices. These filters are:

- Video Capture Filter (DirectShow-compatible Source Filter);
- Sample Grabber Filter (DirectShow-compatible Transform Filter);
- Video Renderer Filter (DirectShow-compatible Render Filter).

Because the production of these DirectShow filters was a very crucial element of the AR interface, further analysis has been provided on the following subchapter. The following paragraphs of this subchapter present information on the implemented process, which has been applied on the framework to acquire video content and present it on the device display.

AuraGraphManager is the main class, which implements the *Filter Graph Manager* and most AR functionality offered by the framework. Furthermore, *ArDlg* is the interface, which the user can call to adjust the internal settings of *AuraGraphManager*. Although *ArDlg* depends on *AuraGraphManager*, both classes are highly cohesive and decoupled. This means that *AuraGraphManager* may be used in other implementations, which require a *Filter Graph Manager*. An analytical description of these classes containing a full list of attributes and procedures can be found in the 5TH Appendix.

AuraGraphManager is a powerful class, which is based on the *Singleton* design pattern. The main reason is that it has to be concurrently accessed by several other classes of the framework. The most important are *ArDlg* and *AuraContextManager*. The structure of *AuraGraphManager* is composed out of two elements, which are described below.

1. Component Controller
2. Filter Graph Manager

Component Controller

The first element is the *Component Controller*, which permits the class to communicate with the external host application (e.g. standalone application or web browser). Its main functionality is to receive commands from the user for triggering and stopping the operation of the system, as well as for handling the input of the video content that needs to be transferred to the internal DirectShow structure. The commands which implement these functionalities are COM-based, which means that they can be called by using any external means, such as JavaScript in the body of an HTML document. This was found particularly useful while developing ARIE, which can run on web pages, achieving real-time functionality. Because the *Component Controller* part of this class is interacting directly with the user interface, a robust solution was required. The main reason is that the user must have absolute control over the development and operation of the *Filter Graph*. Consequently, the *Component Controller* controls the *Filter Graph* by implementing several high-level functions so that it can manage potential errors and inform the user of the result of each process. These commands are listed below, in order of execution, and can also be found in Appendices V and IX. The *Component Controller* includes two wrapper commands, which can be used to execute the *Filter Graph*, in cases where the default configuration is needed without the need of verbose error reporting (e.g. when the component is run through an external application).

- `short CAuraGraphManager::StartAuraGM(HWND* hwnd);`
    1. `short CAuraGraphManager::InitialiseThread(void);`
    2. `short CAuraGraphManager::BuildCaptureGraph(void);`
    3. `short CAuraGraphManager::RunCaptureGraph(void);`
    4. `short CAuraGraphManager::StartPreviewVideo(void);`
- `short CAuraGraphManager::StopAuraGM(void);`
    1. `short CAuraGraphManager::StopPreviewVideo(void);`
    2. `short CAuraGraphManager::CleanCaptureGraph(void);`
    3. `short CAuraGraphManager::TerminateThread(void);`

Table 5-26: Aura’s Component Controller commands for the AR interface

For the smooth operation of the system, the *Component Controller* manages the two concurrent threads of the class. The first thread creates and handles events, which process the commands generated by the user interface and by extension, the user. The second thread processes internal DirectShow generated events, such as the destruction or the repaint of a window region. It is vital for the operation of the system that the thread, which manages the user interface, should initiate the process.

Filter Graph Manager

The second element of *AuraGraphManager* is the *Filter Graph Manager*, which handles the internal functionality of DirectShow. The *Filter Graph Manager* implements the creation and connection of the required filters sequentially. Due to the fact that the system must be compatible with devices that have different input (i.e. camera frames) and output (i.e. rendering images) specifications, the *Filter Graph* must handle several options. This has been accomplished by a series of commands, which are listed below in order of execution and also presented in Appendices V & IX.

```
1. short CAuraGraphManager::CreateCaptureGraphInternal(void);
2. short CAuraGraphManager::RunCaptureGraphInternal(void);
3. short CAuraGraphManager::StartPreviewVideoInternal(void);
4. short CAuraGraphManager::StopPreviewVideoInternal(void);
5. short CAuraGraphManager::CleanCaptureGraphInternal(void);
```

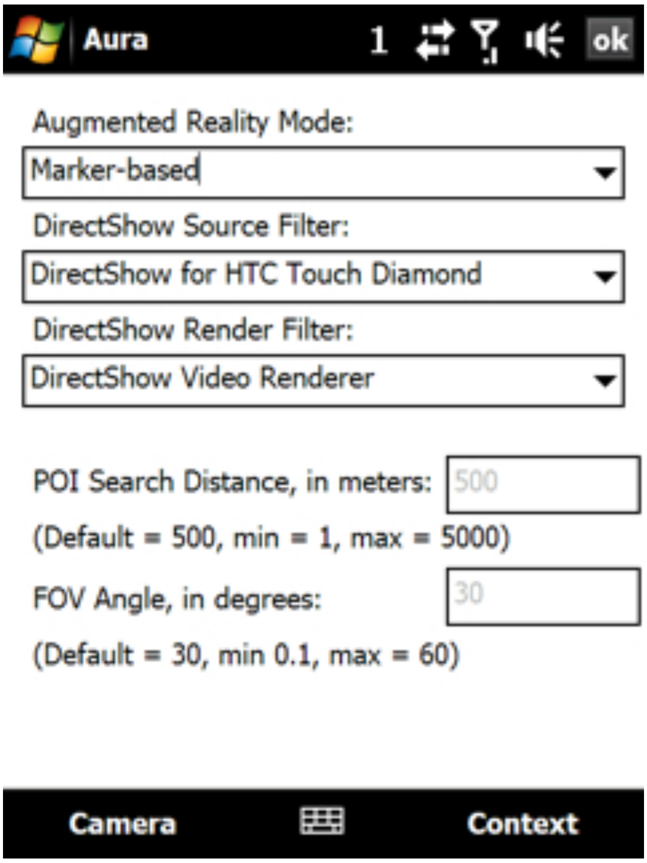
Table 5-27: Aura's Filter Graph Manager commands for the AR interface

The most important command is *CreateCaptureGraphInternal()* because it checks the device specifications, creates the actual *Filter Graph* and attaches it to the suitable source, transforming and rendering DirectShow filters. All filters that have been used in *AuraGraphManager* have been custom-developed for the purpose of this project. An analysis of each developed filter will follow in the next subchapter. After the execution of *StartPreviewVideoInternal()*, the *Filter Graph* should start running and visible feedback should be available for the user. At this point *AuraGraphManager* accepts every frame, which passes from the *Filter Graph* and controls the whole operation. In order for this class to receive a video frame, a system event has been created, which calls the function *OnSampleProcessed()* of *AuraGraphManager*. Another, external, class triggers this event. This class is called *AuraTransformFilter* and it effectively implements the transform filter used by *Aura*. This transform filter belongs to the *SampleGrabber* type of filters, which retrieve every frame that passes from the *Filter Graph*, in order to enable post-processing and image enhancement functionalities. When *OnSampleProcessed()* is triggered, *AuraGraphManager* communicates with *AuraContextManager* and superimposes additional content on the selected video frame. *AuraGraphManager* also implements utility functions, which can perform low-level operations on each frame, mainly for compatibility reasons. These functions are listed below, but we won't further analyse them because their functionality is self-descriptive.

```
•  short CAuraGraphManager::CopyFrame(IMediaSample* pSource,
   IMediaSample* pDest);
•  short CAuraGraphManager::ChangePixelColor(BYTE* imageData,
   int pixelPosX, int pixelPosY, int imageWidth, int
   imageHeight, int bytesPerPixel, int red, int green, int
   blue);
•  short CAuraGraphManager::FlipImageVertical(BYTE* pImage,
   int width, int height, int bytesPerPixel);
•  short CAuraGraphManager:: FlipImageHorizontal(BYTE* pImage,
   int width, int height, int bytesPerPixel);
```

Table 5-28: Aura's Utility commands for the AR interface

An additional external library that has also been integrated to *AuraGraphManager* is *ARToolKitPlus*. ARToolKitPlus is the mobile version of ARToolKit, described at the beginning of Chapter 5.5. This software library has been used to retrieve the position and orientation of a camera relative to the physical markers that exist in the immediate environment. ARToolKitPlus has been developed by Daniel Wagner (Wagner and Schmalstieg, 2007), but there have not been any new updates for the public since 2006. ARToolKitPlus has been superseded by Studierstube Tracker (Schmalstieg and TU Graz, 2010); a library that is not publicly available. The reason that this software library has not been presented on Chapter 4.1.2.2 is that it does not constitute an integral part of this research because the tracking method that has been utilised is based on the recognition of markers. It has been attached to a certain version of *Aura* in order to measure its effectiveness and because the development of our framework allowed straightforward integration of that component. It must be underlined that ARToolKitPlus does not read camera images or render any geometry. Our framework provides the bedrock on which it can be attached, offering a secondary solution for AR functionality. *AuraGraphManager* is the class, which acts as a connector with ARToolKitPlus and is capable of presenting the potential of the marker-based AR approach. Further description regarding the development and integration of ARToolKitPlus has not been provided, as it is not an objective of this project.



Augmented Reality Mode:
Marker-based

DirectShow Source Filter:
DirectShow for HTC Touch Diamond

DirectShow Render Filter:
DirectShow Video Renderer

POI Search Distance, in meters: 500
(Default = 500, min = 1, max = 5000)

FOV Angle, in degrees: 30
(Default = 30, min 0.1, max = 60)

Figure 5-20: Aura's interface for accessing marker-based AR

5.5.6.2 DirectShow Filter Production

The main element of DirectShow is the *Filter Graph*. The *Filter Graph* is used to control the multimedia streams, which can take the form of images, video or audio. Using DirectShow gives developers the opportunity to implement several multimedia functionalities, such as previewing a file, saving a live video stream on a file or even changing the format of a file. To achieve the potential functionality, the *Filter Graph* separates the process into smaller tasks. A dedicated filter of the Filter Graph accomplishes each task. The filters are also known as codecs. Therefore, a *Filter Graph* is composed out of a series of interconnected filters, which provide the required functionalities. There are three categories of filters. These categories are *Source*, *Transform* and *Rendering*. Source filters can either retrieve data from a file, the web, or from a live source, such as a camera or microphone. Transform filters have been used to compress/decompress, add effects, multiplex/de-multiplex and for most other transformations that need to take place before rendering the multimedia content. Rendering filters present the decompressed content that resulted after it came through the former filters. Each filter may have several input and output *Pins*, which are used to connect it with other filters. Pins negotiate and transfer content between filters. Pins can only exchange content that is compatible with their media type. Each filter must have at least one pin, according to its type. A source filter may have more than one output (e.g. preview or full resolution) but no input pins. Alternatively, a rendering filter must have at least one input but no output pins. Finally, transform filters must have at least one input and one output pin, according to the functionality that they offer. If an application that has an implemented *Filter Graph* needs to use a specific filter, it must search in the device registry for the appropriate UUID of that filter, or its source code must implement the necessary filter functionality. In the first case, the filter should have been installed on the selected device in advance. Every filter that has been developed for this project, apart from the source filter, comes as a reusable component (i.e. dynamically-linked library), so that it may be exploited by other applications and various devices.

Video Source Filter

The source filter has not been implemented as a standalone component because it must load the camera driver of the device, which is dependent on the hardware specifications. For that reason, *AuraGraphManager* implements that functionality in the body of the class. Chapter 5.5.6.1 mentioned that three different handlers have been developed, relevant to the capabilities of the device. Therefore, three different versions of the source filter have been implemented. The first version is compatible with every Windows Mobile device supporting DirectShow. The second version offers minimum operational capabilities, whereas the third exposes the full functionalities offered by the framework. The first action that is accomplished when building the *Filter Graph* is to attach the source filter. Every video source filter version is initialised by loading the camera driver of the mobile device. Windows Mobile devices, which have an embedded camera, retain in their registry and in persistent storage, a series of software drivers. One of those drivers is hardware independent and is available on every DirectShow-compatible device. That is the driver called by the first version of the video source filter in *AuraGraphManager*. The registry UUID of this driver is {CB998A05-122C-4166-846A-933E4D7E3C86}. The drivers that are loaded on the other two versions are hardware-dependent. For the second version, a generic HTC camera driver has been used; compatible with most HTC branded and rebranded devices. For the third version of the source filter, a new driver, compatible only with HTC Touch Diamond, has been utilised. After loading the camera driver to the source filter, *AuraGraphManager* must retrieve the specifications of the frames produced by the camera (e.g. video size, pixel format) so that the following filters of the *Filter Graph* are set up accordingly. After this information has been gathered, the video source filter is manually attached to the *Filter Graph*. This filter has 2 output pins. The first pin produces reduced quality multimedia content (i.e. PIN_CATEGORY_PREVIEW) and the other one produces full resolution content (i.e. PIN_CATEGORY_CAPTURE). *AuraGraphManager* uses the one that produces the best quality, although it requires more resources.

Video Transform Filter

The use of the next filter is more complicated than the source filter. That is because it has been developed as a dynamic library and it must be installed on the device before it is called by any instance of the AR interface. This filter is an *in-place* transform filter. In-place means that when the filter receives a new video frame, it sends it for post-processing to *AuraGraphManager*, waits until it has been manipulated and then passes it to the next filter in sequence, so that it can repeat the process with the following frame. This solution was found ideal for meeting our requirements. The filter could also work by taking a copy of the frame from the source filter, sending it to *AuraGraphManager,* and without waiting for the post-processing to be completed, to forward the current frame to the next filter and carry on with the following frame. This scenario has not been found suitable because the flow of frames is not managed and the result could be that some frames have context representations overlaid, while some others do not. Such functionality will produce incoherent results that may disrupt the visualisation experience of the user. *AuraTransformFilter* has one input and one output pin. When the filter receives the instruction to connect to a source filter, it initially checks that the input pin accepts the same media type as the output pin of the previous filter. Following next, for every frame that it receives, it checks the header of the frame so that it can verify the pixel format, the supported colours, the bit count, the stride, the width and height of the image. Then, the transform filter determines the buffer size that is required and allocates memory for that frame. When the frame is loaded, the transform filter notifies the *Filter Graph* so that it can start processing the current frame by triggering an event of *AuraGraphManager*. This event is registered when adding *AuraTransformFilter* to the main *Filter Graph*. The memory pointer holding the video frame data is passed to *AuraGraphManager* as a parameter of *AuraTransformFilter:: Transform()*, including any additional information about the frame format. The function of *AuraGraphManager* that is called every time that *Transform()* is executed is *OnSampleProcessed()*, which was described in Chapter 5.5.6.1. The UUID selected to represent *AuraTransformFilter* is {974029F8-32DD-48b6-8ED1-9127375A3AAA}.

Video Rendering Filter

The last filter connected to the *Filter Graph,* is the rendering filter. The presentation of the video content and the additional context representations takes place in this filter. *AuraRenderFilter* is also responsible for managing the window parameters (e.g. size, focus) that encompasses the visualisations. The thread that executes this filter is responsible for handling any user-generated actions, which target the video window. Similarly to the source filter, the rendering filter has more than one implementations, according to the device hardware capabilities. The first version is the default filter, which is provided by DirectShow. This version is compatible with most Windows Mobile devices and it is used to present video content in almost every mobile application that requires it (e.g. mobile Windows Media Player). The second version has been developed to enhance the rendering capabilities of devices that support advanced visualisation functionalities. Although the source filter produces the actual video content of each frame and *AuraTransformFilter* embeds the context representations, *AuraRenderFilter* is responsible for presenting the final result according to its type. It was mentioned earlier that the AR interface is capable of superimposing 2D context representations but a lot of effort has been made, in order to be able to superimpose 3D representations as well. Therefore, the first version of the rendering filter aims to satisfy the presentation of 2D graphics, whereas the advanced second version to overlay 3D elements on top of video content. As a result, the internal architectures and the utilised technologies are dissimilar for both versions. More information about the type of visual effects supported by each version of the rendering filter and how do contextual representations get overlaid on the video content is provided on Chapter 5.5.6.3. When *AuraRenderFilter* is attached to the *Filter Graph*, it queries each frame and the output pins of the previous filter for information about the video content. This information is used to modify the presentation window settings. These settings make the window come to the foreground and become visible, as well as making it a child window of the main *Aura* window. Furthermore, *AuraRenderFilter* performs the following actions in order of sequence. It acquires the available client area of the owner window (i.e. main *Aura* window). It also retrieves the native video dimensions, disregarding the source rectangle that might have already been automatically set. Then, if possible, it tries to adjust the original video rectangle according to the video dimensions. The renderer will scale or crop the video, if it is required. Following next, *AuraRenderFilter* sets the default destination position of the video window so that the renderer uses the entire window for playback (Microsoft, 2010). After that the filter calculates the rectangular in the client area, which will be used to display the video. If the target rectangle is too big, its width and height will be resized to fit in the client area. In all cases, the video window will be centred in the owner's window client area. The UUID of *AuraRenderFilter* is {75B6423E-13AD-4bda-9B1C-0344E1CF3D74}

5.5.6.3 Superimposing Elements

The final element of the Augmented Reality interface is concerned with overlaying digital descriptions of the sensed context (e.g. remote user, POI) on top of the video feed. To achieve such functionality, two solutions have been implemented, according to the hardware rendering capabilities of the mobile device. The only fully functional solution, though, is the one that can sustain the presentation of 2D elements, like textual annotations or descriptive symbols like arrows. The second solution was designed to present attractive 3D representations, but it was not fully implemented due to the fact that we did not possess a Windows Mobile device that could support the achieved software design. As a result, the conceptual design and part of the advanced second implementation is in place, but has not been fully debugged to work on Windows Mobile devices. According to the functionality of each solution, the process of superimposing the descriptions of context entities takes place at different parts of the framework. The reason that did not enable the development of a unifying solution is related to the type of the employed underlying technologies.

Figure 5-21: Aura's AR interface pointing towards Wood Green Crown Court

Mobile DirectDraw & GDI Output

The first module that has been produced is compatible with every Windows Mobile device available (version 2002 onward). The information that passes through this component is depicted on a live video stream for the time that is required and on a predefined position on the display real estate. Native functions have been extended to visualise graphics and textual content. Illustrations can be selected from a library of distinct icons that can be translated to relevant navigation instructions (e.g. move forward, turn left, turn right). Alternatively, the generated video frames can be enhanced with textual annotations, which describe relevant context entities. The graphical attributes of the text (e.g. spatial location, colour and size), as well as the content can be manipulated in response to context changes. This process takes place in the *AuraGraphManager* class, by retrieving input (i.e. the context descriptions) from *AuraContextManager* and input (i.e. video camera frames) from *AuraTransformFilter*. After applying the context descriptions to the selected video frames, the resulting output (i.e. frames with overlaid context descriptions) is then directed back to *AuraTransformFilter* to be presented on the mobile device display. The result of this process is shown in Figure 5-21. The current user location coordinates are presented at the top left part of the display. The description of the investigated remote context entity is shown in the middle of the screen. An additional description of the distance between the user and the remote entity is presented at the top right part of the mobile display. Further informative descriptions, such as the current orientation, can be revealed on the device display and were included in the version of the system that has been extensively evaluated. Therefore, the AR interface has been kept as simple as possible in order to avoid information overload. Two native software-drawing libraries have been utilised for the implementation of this module, because it had to support a wide range of mobile devices. The execution of the module queries the device for its hardware capabilities and loads the library, which is compatible with the underlying characteristics. More advanced devices, in terms of graphic rendering capabilities, use the *DirectDraw Mobile* library to acquire the video surface and draw 2D symbols on it. In contrast, mobile devices with limited graphic-subsystem specifications employ the *GDI* library as a fallback mechanism to draw textual annotations. It has been essential to implement both versions of the module for drawing 2D content because it presents the output of the AR capabilities of the framework. A procedural list of the GDI commands used to visualise textual context descriptions in the AR interface is presented below.

```
1. SetBkMode();
2. GetTextMetrics();
3. CreateFontIndirect();
4. SetTextColor();
5. TextOut() or DrawText(); *
6. SelectObject();
7. DeleteObject();
```

*_TextOut()_ is a lighter function, but offers less options and does not word-wrap or align the result.

OpenGL ES Output

The second module is more advanced, in terms of the quality of output. It also requires the use of powerful devices, which support the underlying technologies. The balance between the effort needed to develop this module and the final outcome was not satisfactory, because between 2005 and 2009 OpenGL ES devices were not commercially widespread and the latest did not support the platform that we had committed our development efforts to. The device, which was selected to produce the module that would overlay 3D content on top of the streaming video, was HTC Touch Diamond. This device supports OpenGL ES with Common Profile v1.0. The ideal functionality of this module would be to superimpose 3D representations of objects at precise screen positions according to their real world coordinates. This would produce better visual results and it would enhance the immersion of the user to this augmented environment. The combination of DirectShow Mobile and OpenGL ES is required to accomplish such functionality. A description of the conceptual process, which brings this module to life, is following.

Due to the fact that we had already developed a custom renderer filter for DirectShow, we could use this filter to produce the combination of 2D and 3D graphics. The major difference would be that the renderer filter should not create a default DirectShow window, but instead it should create an OpenGL ES window by using EGL as the native platform graphics interface layer. EGL is a platform-independent programming API, which provides the surfaces that OpenGL can draw on. The produced window should be registered as a Windows class in the operating system. After the new window has been created, the EGL environment should be configured according to the device specifications. After successful configuration, the window surface becomes available for drawing. The window surface produces a handle for the device context, which we are going to use for the rest of the process. The next step is to draw a simple rectangle, which will hold an OpenGL texture. This rectangle should fill the whole visible surface, so that there is not going to be any blank areas displayed on screen. The texture is intended to present the contents of the video frame. *AuraRenderFilter* obtains the current video frame and stores it in memory buffer as a BYTE array, after receiving it from *AuraTransformFilter*. After binding the named 2D texture to the target and setting its parameters, the system is ready to receive and present the first frame. Before continuing, it is extremely important to set the perspective of the 3D environment to match the perspective of the camera. This is accomplished by taking into consideration the variables produced during the camera modelling procedure, which was described on Chapter 5.5.5. After this step, the initialisation of OpenGL has been completed. Whenever a new frame arrives to the graphics pipeline, it can be presented as the content of the OpenGL texture that has been created. OpenGL will render the BYTE array image natively and without any transformations because it can accept unsigned characters, which are in essence bytes, as the content of the texture. The format of the bytes describes RGB565 structures, which are compatible with most Windows Mobile devices. Now that the 3D environment has been set up and can present video frames as background images, we can introduce the representations of real world contextual entities. These can either be textual annotations, 2D images or 3D objects. The introduction of animating 3D objects in precise locations over streaming video that describes the immediate surroundings is undoubtedly a feature, which would enhance the system appeal towards potential users. It can also enable the implementation of several applications, which can be supported by the framework and are also presented in Chapter 6. This process can be repeated for every new video frame and for any change in context, until the user stops the execution of the AR interface. By destroying the *Filter Graph*, every DirectShow filter is destroyed and the memory is released. *AuraRenderFilter* will also be destroyed, as will the OpenGL surface and context, in turn.

Further development of this module is intended to progress in future releases, even out of the context of this research. Newly available devices, which support other platforms such as Apple’s iPhone (Apple, 2011) and Google’s Android (Google, 2011a), render the development of this module far simpler. Furthermore, we are also expecting the latest Windows Mobile operating system (i.e. Windows 7) to become available, which - according to Microsoft - will offer better integration with custom multimedia applications. It becomes possible because new devices are going to support OpenGL ES 2.0 or OpenGL ES 1.1 with Common Profile. Furthermore, recent devices include supplementary OpenGL-specific extensions, which make the process more straightforward. The functionalities of these extensions include the automatic presentation of video content on texture surfaces, as well as 24-bit colour, a feature that is indispensible for AR applications due to true transparency.

5.6 Networking System

With the advances that have taken place in network availability and especially the straightforward accessibility of the web infrastructure, a vast number of users have been collaborating on a daily basis by using digital communication technologies. The role of such communications is quite wide, ranging from simple, friendly text messaging to more complicated information exchange that influences and strategically affects large business organisations or independent users in their daily activities. Sophisticated networking infrastructures have already been developed, which satisfy the underlying requirements of the relevant processes. Peer-to-peer and more complex distributed topologies represent the progress that has been achieved in order to successfully serve their users in complex tasks such as file-sharing or online gaming. Alternatively, *mature* client-server architectures are still serving most interconnected systems as they are continuously becoming more efficient and offer different characteristics to the more advanced distributed technologies. As a result, there is a suitable network topology for every application that needs to exchange information with remote parties according to the requirements and business models.

The introduction of mobile platforms has created new networking needs that must be accommodated to establish communications between two or more mobile devices or between mobile devices and static hosts. They have also fundamentally elevated the functionality of the integrated applications by employing real-time data and allowing the management and distribution of a large volume of information even in the context of GIS. Consequently, new services can and have been developed, which offer to their users superior functionality, even on mobile platforms. Due to several technical and business-oriented limitations, the variety of networking options has not reached those defined by non-mobile equivalents. More precisely, in the context of the research there are several factors, which have been described in the previous chapter that instruct the design and implementation of a new networking protocol that would be built upon the aforementioned technical and functional requirements.

This subchapter presents the description of the features of the networking protocol that has been developed, as well as further low-level considerations that have been adopted during its implementation. This component of the framework tries to satisfy the requirements, which have been identified on Chapter 4, as well as to enable scalability when *Aura* launches.

5.6.1 Protocol Description

Aura's networking protocol is a telecommunication system, which has been designed and developed during the course of this project and can be used by any application that needs to create groups of interconnected hosts, in order to exchange context variables in real time. The system is based on the client/server topology and the interconnected subsystems (i.e. mobile devices) can be attached either through wired or wireless physical-layer media. The configuration of the system is variable, depending on the governing application. In every case though, two or more units must exist and one of them must be assigned as the server.

The formulated network architecture is compatible with the TCP/IP requirements and supports connectionless or asynchronous message exchange for the reasons stated in Chapter 4. Furthermore, all interconnected hosts are synchronised with a common clock. Network synchronisation is provided by a preselected unit, which is the server unit. Selecting a particular host to act as the server of the network is a function that is fully controlled by the software of that host. In terms of functionality, the networking software component, which has been embedded in the framework, can offer both client and server functionalities.

Two main classes have been introduced, which implement the networking functionalities. These are *CAuraSocket* and *CSocketController*. A complete description of these two classes is illustrated in Appendix V. *CAuraSocket* is a low-level class, which implements the event handling of the networking component, whereas *CSocketController* is concerned with the formulation and distribution of the messages between the interconnected hosts. These messages can handle two distinct types of parameters. The first type is concerned with the exchange of personal user information (i.e. user profile) and the second type is concerned with the exchange of location information. Because both types of information are considered sensitive and could identify a particular user, each participating host may choose not to make that information available to the public. This functionality is offered by the networking protocol and overrides all other settings due to the user-applied privacy restrictions. A host has the option to either discard any user profile requests, any location requests, or both, depending on the current personalisation options.

Address Localhost ☑ Connected

Mode Server

Other: Maria Ges, 25
Me: hi Maria
Other: hello
Me: where are you?
Me: Remote location request sent!
Other: Location exchange rejected by remote
Other: i can't tell you
Other: we just met

you can check my profile, if you want

Figure 5-22: Aura's social communication interface

5.6.2 Server Mode

The server unit is the main entity of *Aura*'s networking protocol. This unit can provide transmission and reception of digital contextual information to the clients of the service. It can have the form of an independent application, residing on a dedicated host (i.e. server) or it can be embedded on a mobile device application. The latter is the case, which has been applied, in order to implement the communication requirements, which have been identified by the research.

In the server mode, the networking component supports five networking functions. These are listed below.

1. Create Socket;
2. Listen to Socket;
3. Accept Connection;
4. Send or Receive Data;
5. Terminate Connection.

Initially, the server creates a socket on a specified port. In the case of *Aura*, the preselected port number is *7777*. After the socket has been successfully created, the server is ready to start listening for connections triggered by the remote clients. The server waits for incoming connections from every available network interface and IP address that is bound to that host. In more detail, a mobile device may be concurrently connected to more than one Wi-Fi network, USB and possibly Bluetooth PAN. That is why the protocol is capable of scanning all data-link layers and listen for incoming connections on every available channel. When a client is willing to connect to our host it generates a request. The server then checks if it has already been connected to that or any other client. If an active connection has already been established, the server rejects the new connection and informs the new client by sending a predefined message (i.e. `"Aura: Too many connections, please try again later."`). This version of the networking protocol does not support multi-client connections because it was not found obligatory for the course of the project, as the main objective has been to transfer real-time context and not to develop a complete networking solution supporting more than one live client. On the contrary, one-to-one communication has been fully developed and tested to work sufficiently for satisfying our requirements. In real-world conditions, though, concurrent multi-client compatibility is a must, which requires proper implementation through multithreading mechanisms. If the server verifies that there is not any other connection available, it continues by accepting the connection from the client on a specific socket. When a connection has been established, the server automatically creates a new profile for the remote user and fills it with the default variables. The remote profile is then visible to the user of the server. Consequently, the setup process has been completed and the actual functionality may take place. The server's primary functionality includes transmitting and receiving information about the user (i.e. server/client profiles) and about the location that is currently occupied. This is accomplished by replying to messages generated by the other host and transmitted over the existing connection. Furthermore, the server is capable of exchanging custom messages between the users of the system, similar to an instant messaging application. The user can terminate the connection with a client at any point, by selecting the appropriate user interface command, which in turn discards the socket connection.

5.6.3 Client Mode

The other instance of *Aura*'s networking component is the client. This unit is responsible for accessing a server and requesting up-to-date information. Similarly to the server unit, it can have the form of an independent application residing on a dedicated host or it can be embedded on a mobile device application (i.e. *Aura*).

In the client mode, the networking component supports four main functions. These are listed below.

1. Create Socket;
2. Connect to Socket;
3. Send or Receive Data;
4. Terminate Connection.

The client must create a socket in the same way as the server does. After the socket has been created, it can be used to connect to the server unit. To achieve that connection, two parameters need to be fused to the component. These are the remote host name (i.e. IP address) and the port number, which is a constant for both units (i.e. 7777). The *Connect to Socket* function is particularly important because it searches for hosts, which conform to the input parameters. As a result, it may take a while to locate a particular server or it may fail due to an unresolved error (e.g. resource temporally unavailable). In either case, the client unit must resubmit the request until it finds the host, or even after the connection has been lost due to unforeseen reasons. Likewise, the client must also create a new remote user profile after a valid connection has been established. The profile is then filled with the values that have been retrieved from the server. When two hosts have been successfully associated, the client is ready to start exchanging information with the server. This is accomplished by formulating the standard messages of the protocol, which are described in more detail in the following subchapters, or by sending custom user-generated text messages to the remote user. These messages are then transmitted and a relevant response is expected. If a message is lost or received incorrectly, the functionality is not affected at all because the protocol is connectionless, which means that it works over UDP. This was found particularly important if the connection is established over a wireless medium (e.g. Wi-Fi, 3G). The simplest way to resolve a lost request is to resubmit it to the server once again and expect for the reply. Finally, the local user may wish to manually terminate the connection at any time by selecting the available user interface command. This way the server is notified and closes the bound socket.

5.6.4 Protocol's Message Format

The following table describes the general format of the messages that can be exchanged between the mobile devices. It was found obligatory to define a maximum number of bytes that each message would be capable of transferring. The main reason was that the protocol is not very complex and, if we did not introduce this limitation, we should have introduced a new field for the message format, which would denote the full length of each message, in terms of bytes. That is why the maximum size of the exchanged messages was set to 256 bytes, data and content inclusive.

PD (5 bytes)	**CT** (10 bytes)	**Params** (Max: 241 bytes)

Table 5-29: Typical message format

This table maintains the structure of the data that can be exchanged between a server and a client that is using *Aura*'s network protocol. In more detail, the *PD* field is the *Protocol Descriptor*. It is a constant, which contains a predefined array of bytes. The size of this field is always 5 bytes. The value of this variable is always the same (i.e. Aura:). If a message has been received and it does not start with the default PD field, then the message is discarded by the networking component.

The *CT* field defines the *Command Type* that is transferred by the message and is variable, depending solely on the transmitted message. The size of the command type is always 10 bytes. The value of this field always starts with a specific character (i.e. ^) and is followed by 9 bytes that describe the command.

Finally, the *Params* field is constituted by a variable-sized byte array. It is directly influenced by the *Command Type* and the number of included parameters is also related to the corresponding command. The maximum size of this field can be 241 bytes. The parameter values, which are attached to the message, start with a specific delimitation character (i.e. %) and each value is separated from the next one with the same character. Following the last parameter, the delimitation character is added again, which makes this field conform to the format '%12345.6789%', in case there is only one transmitted variable. In contrast, if the command type is a request for information, this field is empty and its default byte size is 0.

The following table describes the available command types and their values, which vary depending on the purpose of each message.

Field Value	Description
REQMAIPRO	Request main user profile
RESMAIPRO	Response with main user profile attached
REQCURLOC	Request current remote location
RESCURLOC	Response with current remote location attached
(Empty)	Instant text message

Table 5-30: Comprehensive description of the available protocol Command Types

5.6.5 Detailed Message Description

In this section, the reader can find an analytical description of all available messages that can be exchanged on top of *Aura*'s networking protocol. Furthermore, all possible responses to those messages are listed and an explanation of their cause is documented.

5.6.5.1 Request Main User Profile

Aura: (5 bytes)	**^REQMAIPRO** (10 bytes)	**VOID** (0 bytes)

Table 5-31: Request for main user profile message format

The purpose of this message is to request from the remote party the main profile of the user that is currently logged in or is using the device and, consequently, the application. The main user profile holds minimum information about the user, which has been generated by that person through the user personalisation interface. This personal context can be used by the local application to identify and possibly match the interests of two persons. Currently, the variables include limited information about the user and a new command type may need to be developed, to exchange more detailed personalisation settings. This is a special message because it is transmitted immediately and automatically after a connection between two mobile devices has been established. This functionality serves three purposes. Initially, both the local and the remote user can verify with whom they have established communication. Additionally, users can exchange the privacy settings that they have applied regarding their profile distribution. Finally, by automatically exchanging profiles after a connection has been established, the protocol can accomplish a self-test in order to verify that the technical communication channel is operating efficiently.

5.6.5.2 Response with Main User Profile

The response expected by a host, after successfully sending a request for the main user profile to a remote device, may conform to one of the following three formats.

Aura: (5 bytes)	**^RESMAIPRO** (10 bytes)	**%** (1 byte)

Table 5-32: Profile exchange rejected by remote host message format

If a host receives the above message in reply to a request for the main user profile, it means that the remote device user has not allowed any personal information to be distributed over the network. This message does not produce an error, nor does it restrict operation of the system functionality in any way. The remote user profile fields are then filled in with the default values. If at any point, the remote user decides to raise that limitation, the full profile may be retrieved after resubmitting the request.

Aura: (5 bytes)	**^RESMAIPRO** (10 bytes)	**%Name%Surname%Age%** (Min: 4 bytes – Max: 241 bytes)

Table 5-33: Valid profile has been received message format

The above message shows the valid format of a message, which holds the main profile details of a remote user. The variables that have been introduced are *Name*, *Surname* and *Age* of the user. *Name* and *Surname* are strings of characters and *Age* is a number, which is represented as an integer. The minimum size of the parameter field can be 4 bytes because of the delimitation characters and the maximum size can be 241 bytes. For the content of these variables, the maximum available size is 237 bytes. The following table shows a valid message that includes the details of the author, at the time that this document was in preparation.

Aura:^RESMAIPRO%Stelios%Papakonstantinou%28%

Table 5-34: Valid profile has been received demo message

Aura: (5 bytes)	**^RESMAIPRO** (10 bytes)	**RANDOM** (Min: 0 bytes – Max: 241 bytes)

Table 5-35: Unrecognised profile has been received message format

The previous table shows the third possible message format option, which may be received by a host. This option is discarded by the networking protocol, because the format does not conform to the protocol design. The parameter field size may vary, but the maximum length should be 241 bytes or less.

5.6.5.3 Request Current Remote Location

Aura: (5 bytes)	**^REQCURLOC** (10 bytes)	**VOID** (0 bytes)

Table 5-36: Request for current location message format

The purpose of this message is to request the latest location that has been occupied by the remote party. The latest location is acquired through the sensors and saved on the remote host. After receiving this command, the remote host needs to reply with the current location, if the remote user has authorised such activity. This message can be either user generated or application generated. Furthermore, the transmission of this command can be recursive. This takes place when the local user wants to *follow* the route of a remote user and present the trajectory in one of the available visualisation interfaces (e.g. VR). The parameter field of this message is always empty.

5.6.5.4 Response with Current Remote Location

The response expected by a host, after successfully sending a request for the current location to a remote device, may conform to one of the following four formats.

Aura: (5 bytes)	^RESCURLOC (10 bytes)	% (1 byte)

Table 5-37: Location exchange rejected by remote host message format

If a host receives the above message in reply to a request for the current location, it means that the remote device user has not allowed the location information to be distributed over the network. This message does not produce an error, but it prevents the local user from observing the movement of the remote user on any visualisation interface. The remote user location fields are then left blank. If at any point, the remote user decides to raise that limitation, the remote location may be retrieved after resubmitting the request.

Aura: (5 bytes)	^RESCURLOC (10 bytes)	%Lat%Lon%Datum%Grid% (Min: 5 bytes – Max: 241 bytes)

Table 5-38: Valid 2D position has been received message format

The above message presents the valid format of a message, which holds the current 2D position details of a remote user. Two-dimensional positioning is available when the remote host does not hold any information about the altitude from the mean sea level. This may happen due to limited GPS satellite availability. The variables that have been introduced are *Latitude*, *Longitude*, *Datum* and *Grid*. *Latitude* and *Longitude* are decimal numbers and they are represented by using the double format (e.g. ±*123456.789012*). Precision of the transmitted position is high because the protocol supports 6 digits after the decimal point. *Datum* and *Grid* are represented as an integer. Each integer is matched with the corresponding entry in *Aura*'s supported datum and grid tables. This way, *Aura*'s context manager is capable of interpreting many different types of position data. As a result, communication between two parties can be achieved without incompatibilities and without overloading the user by interfering with needless transformations. The minimum size of the parameter field can be 5 bytes because of the delimitation characters and its maximum size can be 241 bytes. For the variables' content, the maximum available size is 236 bytes. The following table shows a valid message that describes the location of the author based on the WGS84 datum and the British grid.

Aura:^RESCURLOC%51.527327%-0.103788%1%3%

Table 5-39: Valid 2D position has been received demo message

Aura: (5 bytes)	**^RESCURLOC** (10 bytes)	**%Lat%Lon%Hei%Dat%Grid%** (Min: 6 bytes – Max: 241 bytes)

Table 5-40: Valid 3D position has been received message format

The above message illustrates the valid format of a message, which holds the current 3D position details of a remote user. Three-dimensional positioning is available when the remote host holds information about the current altitude from the mean sea level. The variables that have been introduced are *Latitude*, *Longitude*, *Datum* and *Grid*. *Latitude*, *Longitude* and *Height* are decimal numbers and they are represented by using the double format (e.g. ±*123456.789012*). Precision of the transmitted position is high because the protocol supports 6 digits after the decimal point. *Datum* and *Grid* are represented as an integer. Each integer is matched with the corresponding entry in *Aura*'s supported datum and grid tables. This way, *Aura*'s context manager is capable of interpreting many different types of position data. The minimum size of the parameter field can be 6 bytes because of the delimitation characters and the maximum size can be 241 bytes. For the content of the variables, the maximum available size is 235 bytes. The following table shows a valid message that includes the author's location, including altitude information.

Aura:^RESCURLOC%51.527327%-0.103788%87.7%1%3%

Table 5-41: Valid 3D position has been received demo message

Aura: (5 bytes)	**^RESCURLOC** (10 bytes)	**RANDOM** (Min: 0 bytes - Max: 241 bytes)

Table 5-42: Unrecognised position has been received message format

The previous table demonstrates the fourth possible message format, which may be received by a host. This option is discarded by the networking protocol, because the message format does not conform to the protocol design. The parameter field size may vary, but the maximum length should be 241 bytes or less.

5.6.5.5 Instant Text Message

Aura: (5 bytes)	**Payload** (Min: 1 bytes – Max: 251 bytes)

Table 5-43: Instant text message format

This type of message does not follow the request/response format that all aforementioned messages do. It is a special message, which can be generated by the user. Its main purpose is to allow users to communicate via text messages at any time, while the link is up. Instant messaging takes place in real time so that the users will be able to exchange information about personal or any other topics. There is no obligation by any party that received an instant text message to reply to the requesting party. Similarly to the other messages, the *Protocol Descriptor* remains unaffected at the beginning of the message, in order to verify that another instance of *Aura* has generated the message. The main difference between this message type and the rest is that this one does not include a *Command Type* field. This way, the payload, which in essence is the content of the text message, can use the whole available space. In order for this message to be considered valid, the payload must hold at least one character. Likewise, the maximum payload is 251 bytes.

CITY UNIVERSITY
LONDON

Technological Framework for Ubiquitous Interactions using Context–aware Mobile Devices

---

STYLIANOS K. PAPAKONSTANTINOU

Submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy (PhD) in Information Science

School of Informatics

Department of Information Science

Volume II

Supervisors: Prof. Vesna Brujic-Okretic & Prof. Jonathan Raper

London, August 2012
U.K.

6 Potential Context-Sensitive AR Applications

This chapter presents a detailed analysis of the potential applications that may evolve by customising the developed framework. Several applications have been identified during the course of the project, but this section describes those, which can reflect certain capabilities of *Aura*. The chapter also presents possible solutions for commercialisation, which have been triggered either by interrogating potential users, by research on this field, or by actively getting involved in the promotion of the framework features to potential investors (i.e. stakeholders). A potential commercialisation model can be found in Appendix XIV of the report.

Researchers and application developers have been working with Mixed Reality applications for the past decade, trying to establish their products in specific markets. The developed solutions have been mostly dedicated to certain application domains, and porting them to a different domain has not been a straightforward process. The main reason that has affected such development efforts is the advance of the underlying technologies and concepts that have guided the implementation of those systems. In more detail, Augmented Reality applications depend heavily on the use of hardware resources, especially on the mobile platforms. Furthermore, the techniques and algorithms that have been introduced in such applications are mostly based on the recognition of natural features or markers, which are not widely available or cannot be easily populated in uncontrolled environments, for dynamic and effective scenario walkthroughs. Lately, due to the rapid technological evolution of smartphone devices and the miniaturisation of sensors, the true potential for context-aware Augmented Reality applications has been unveiled. Software product manufacturers, like Mobilizy and Layar, have introduced applications, which utilise techniques that acquire real-time context, such as location and orientation, and overlay digital content over a live camera feed. The solutions provided by both firms focused on Google's Android (Google, 2011a) and Apple's iPhone (Apple, 2011) platforms because they offer camera accessibility, GPS reception and most importantly, orientation information integration on most compatible mobile devices. In contrast, there are not many Commercial Off-

The-Shelf (COTS) products available for Windows Mobile devices, which focus on user collaboration and their social activities.

The following table compares a generic version of *Aura* with six commercial AR applications. All of them are context-sensitive applications operating on top of various mobile platforms. These applications can be considered as POI browsers presenting information on an AR display. The browser evaluation criteria and a more detailed explaination of the terms used can be found in the report provided by Ben Butchart (Butchart, 2011). The generic version of *Aura* that is presented on the following table is considered as a mix of the *Spatial Search* and the *POI Querying* solutions, which are explored in more detail in Chapter 6.1.3 and 6.1.4 respectively. Furthermore, none of the described applications, excluding *Aura*, offers a full VR interface in addition to AR. The alternative interfaces that are provided by some applications are map representations of the environment or textual descriptions of a selection of the environmental elements (i.e. POI).

Product	**Location Sensor-based**	**Marker-based**	**Marker-less**	**Built-in User Actions**	**Publishing API**	**Application API**	**AR Content**	**POI Actions**	**Offline Mode**	**Platform**
Aura	Yes	Yes (ARToolkit Plus)	No	Post text, Post 3D, Social	Open key, (Crowd Sourced)	Customise	2D, (3D)	Info, Map, Search, Event	Offline	Windows Mobile, (iPhone)
Layar	Yes	No	No	WebView	Open key	Customise	3D, 3D-anim, 2D	Info, Audio, Music, Video, Call, Email, SMS, Map, Event	Online only	iPhone, Android, Symbian
Junaio	Yes	Yes	Yes	Post text, Post image, Post photo, Post 3D, Social	Open key, Crowd Sourced	Customise	3D, 3D-anim, 2D	Info, Audio, Video, Map, Event,	Online only	iPhone, Android, Nokia (N8)

Wikitude API	Yes	No	No		Bundled	Open	3D, 2D	Info, Event	Offline	iPhone Android
Wikitude Worlds	Yes	No	No		Open key	Customise	2D	Info, Map, Email, Call	Cacheable	iPhone Android Symbian
Sekai Camera	Yes	No	No	Post text, Post photo, Post sound, Social	Restricted, Crowd Sourced	Commercial	2D	Info, Audio, Map, Social	Online only	IPhone, Android, iPad, iPodTouch
Libre Geo Social	Yes	Src	Plug-in	Post text, Post picture, Post sound, Social	Crowd Sourced, Open src	Open	2D	Info, Audio, Map, Social,	Online only	Android

Table 6-1: AR Browser Comparison (Butchart, 2011)

However, these commercial applications have been developed to satisfy user needs for limited domains like see-through browsing of the world and its elements of interest. Consequently, the consumers have perceived such applications as an expansion to the use orientation information. In order to clarify this issue we have had to disassemble the AR module of the framework, which has been developed during the course of this project, into distinct technological entities. This method has assisted the identification of potential commercialisation options that may promote the use of context-sensitive Augmented Reality, when combined with other applicable technologies. These components have been briefly summarised in the following list and a detailed description can be found in the former chapters of this report.

1. Visualisation Interfaces
    a. Augmented Reality algorithms
2. Context Management
    a. Location algorithms
    b. Orientation algorithms
3. Networking Infrastructure

By identifying the core elements that comprise the Augmented Reality subsystem of the framework and by examining applications that have been proposed by other researchers or developers, several potential applications have become evident. In Appendix XVI a description of each solution has been provided and all of them will be compared in terms of their commercialisation capabilities and limitations. The method that was selected to accomplish this assessment conforms to the *Five Forces* model that has been proposed by Porter (Porter, 1980). In order to match the aims of this part of the report we have not followed a strict implementation of the Five Forces model, but we have customised it for our specific needs. The custom specifications are listed below.

- Maximising potential income;
- Decreasing the risk;
- Managing the limited resources;
- Exploring the portability options.

6.1 Potential Solutions

6.1.1 Technology Licensing

Although this is not an actual application requiring more time to design and implement, it is a very good initial approach to promote the already developed system to external organisations. These organisations can either be mobile operators, smartphone manufacturers or software developing companies, which may generate new revenue streams by extending the framework to support custom applications of their own. The sole responsibility of the researcher and the enterprise unit of the university (i.e. City University Research and Enterprise Unit, aka CRUE) will be to transfer the knowledge and technology to the external parties. The profit generated by the university can have the form either of a sole payment, value sharing, revenue sharing or a custom-tailored arrangement that will be a blend of the aforementioned options. Furthermore, if such a deal is accomplished, the university may have the opportunity to pursue new research objectives that will be instructed by the external body. This will keep a communication channel open and potentially offer long term cooperation with national or international businesses.

6.1.2 Urban Navigation

Although the experimental systems, described in the Literature Review and other parts of this report, focus on some of the issues involved in navigation, they cannot deliver a functional system capable of combining all accessible interfaces, consumer devices and web metaphors in real time. The motivation for the research reported on in this document has been to address those issues, namely the integration of a variety of hardware and software components to provide effective and flexible wayfinding tools. In addition, we have explored potential solutions for detecting the user location and orientation attributes in order to provide more advanced urban navigation applications and services.

To realise it, we have had to design a mobile platform based on both VR and AR interfaces, as a main objective of this research (Liarokapis et al., 2006a). To understand in depth all the issues that concern location and orientation-based services, first a VR interface was designed and tested on a personal digital assistant as a navigation tool. Subsequently, we incorporated the user feedback into the design of an experimental AR interface. Both prototypes require the precise calculation of user position and orientation parameters, to achieve registration. The VR interface is coupled with the GPS and digital compass output to correlate the model with the location and orientation of the user, while the AR interface is not dependent on detecting features belonging to the environment.

The main functionality of this application is to assist a mobile user in finding his or her way in an urban environment. This is accomplished by presenting navigational aids allied to the path that must be followed (i.e. road) and superimposed on distinct physical structures (i.e. decision points, such as landmarks and/or buildings), which are represented through the video see-through display. The advantage of this application is that it solves certain issues that have been identified in expensive and pseudo-3D PNDs (e.g. TomTom) that were implemented on dedicated devices, as well as on inexpensive but heavily network-utilising 2D solutions (e.g. Google Maps). As a result, this application can be a low-cost commercial solution, which does not require any external resources in order to offer the requested services and can run on a variety of already user-adopted devices. The use of an augmented reality system conforming to our framework specifications was presented during the *KTN Flagship Projects Day,* at the National Physical Laboratory, Teddington, U.K. (Papakonstantinou and Liarokapis, 2007)

6.1.3 Spatial Search

Although spatial proximity is the most frequently used measure of geographic relevance, it may not always be the most appropriate filter to describe custom queries of potential users. Geographic relevance can be quantified, which means that selected information can be retrieved from a database by examining this group of attributes. If a user executes a query and receives a series of spatially referenced results, according to specified search parameters, this application could describe the results in a graphically geo-referenced interface like AR. After retrieving the results, which in essence are POIs in the real environment, the user may receive additional information about the geographic relation with each of them. This can be accomplished by gesturing, such as turning the device towards different directions, and by visualising the result descriptions on the AR interface, when the direction overlaps with the position of the POI. The description of every POI that is being displayed may contain information about the geographic relevance with the user, such as the spatial proximity and direction, as well as some contextual information, such as the type of the POI (e.g. restaurant). The advantage of such an application is that it combines virtual and real world information in an egocentric interface, so that users can make more efficient decisions based on quantifiable criteria. The visualisation and interaction options should also be upgraded, compared to other solutions, to match the needs of mobile users. Furthermore, open-data sources of current location-aware search engines (e.g. Flickr) can be investigated through the provided API, but with the additional need of stable 3G connections. Certain stakeholders may benefit from such an application, by promoting their products or services though the system-enabled search mechanisms (i.e. information retrieval relevance). The basic characteristics of *Aura* operating in this scenario are compared with the characteristics of six commercial AR applications such as *Layar* and *Wikitude* in Table 6-1 of this chapter.

Figure 6-1: A user navigating with the help of the Augmented Reality interface

6.1.4 Point of Interest Querying

The main objective of this application is to make it possible for the user to retrieve information about objects of interest or specific locations that exist in the immediate environment. Even though this application can be considered similar to *Spatial Search*, in fact it can work as an extension to it or even as standalone. The use case includes an actor that points his mobile device on a specific POI and receives relevant digital content, which describes the investigated entity. The information can then be presented on the device screen and in a custom interface, not similar to the one provided with the *Spatial Search* application, as the requirements are not limited to geographic relevance. This application may have several purposes, as it can be linked with various online information sources. It can present location-sensitive news if it gathers data though a broad subject portal, such as the BBC website or more specific information if, for instance, it is linked with a history-based network. Another difference with the *Spatial Search* application is that *POI Querying* is intended to be used in more confined environments, which have a narrower focus (e.g. post-emergency management or archaeological excavation site).

6.1.5 Marketing

This application aims to deliver rich media content ads, either in 2D or 3D formats, onto the mobile phone screen directly, to whoever they are relevant, whenever they are relevant and wherever they are relevant. The core structure of a potential solution consists of three entities. These are the *Ad Server*, the *Ad Distribution System* and the *Client Application*, which is installed on the mobile device. Apart from those three elements, this m-Advertising solution functionally depends on the *Advertising Agency*, which needs to create the content of the ads on behalf of their customers. Furthermore, by taking advantage of the GPS and compass sensors, the *Client Application* is always location and orientation-aware. Initially, the *Advertising Agency* needs to pass the contents of their marketing campaign, as well as some metadata describing that content, to our *Ad Server*. This is accomplished by utilising a standard web interface. Following next, the *Advertising Agency* needs to select the target user group of the campaign or to place a generic ad, which will reach all subscribers. The available options, when targeting specific user groups, are based on their location and/or other personal information, such as their occupation. When the user selection process is completed, all information about this campaign is passed to the *Ad Distribution System*. The *Ad Distribution System* is responsible for processing the user contextual information, as well as for the distribution of the ads to the bound devices. In order for the *Ad Distribution System* to obtain the latest user information, the device needs to initialise a data connection at specified intervals. The data that is transmitted to the *Ad Distribution System* contains parameters related to the user and the current location coordinates. Additionally, some device context needs to be passed, which describes the hardware specifications, such as screen resolution, mainly for compatibility reasons. In response, the *Ad Distribution System* transmits the ad content, which can then be visualised on the device display. This way, we are capable of presenting the ads not in an application like a web browser, but on custom interfaces including over real-time video feeds. We have to mention that the ad contents can have the form of 2D animations created with applications such as Adobe Flash, or more complex 3D animations based on VRML or OpenGL ES. After the ad has been successfully distributed to the client, the *Ad Distribution System* must create statistics/analytics about this campaign and pass it on to the *Ad Server*. The analytics can then be queried by the *Advertising Agency* to provide valuable customer feedback. One of the benefits of this application is that it is easier for the *Advertising Agency* to make the users engage with their immediate surroundings.

This happens by drawing user attention on the device screen when necessary. The advantage for the users is that they receive ads that have been filtered to match their personal interests (e.g. based on the age group), in order to avoid information overload. Furthermore, by examining the latest location context, the application can ensure that the ads are spatially relevant. In conclusion, mobile operators should be enthusiastic about collaborating in this attempt, because it can generate a new revenue stream that we are willing to share with them. A positive response has been received from *3 Mobile*, in order to discuss potential collaboration in the field of mobile marketing, as well as from Google for initial consultations. The use of an augmented reality system conforming to our framework specifications has been presented at the *Idea to Product Global* (I2PG) summit, in Austin, Texas, USA (Papakonstantinou and Bhatia, 2009).

6.1.6 Confined Space Entertainment

This scenario takes place in a restricted environment (i.e. Formula 1 track). The advantage of a restricted environment is that the LOD of the virtual world can be enhanced to the maximum. The stakeholders, who host the service, are responsible for sustaining a dedicated, low-latency network infrastructure and for distributing contextual information to the participants. Promotion of this service targets the full-exploitation of the available space. The use case includes one or more actors and at least a F1 car that will have embedded the required sensors and a transmitter. This way, the precise location of the car can be made available to the public. The system may even enable competitive racing between the users and any driver, while the race is still taking place. This is accomplished by accumulating almost real-time data. Apart from real-time competition, the modes of play may include observation of the actual race from any point of the track and exploration of the surrounding space. Effectively, this scenario can prove useful because the actual tracks are enormous and several hours are needed in order to fully explore them while walking. The quality of the modelled environment is a determining factor towards the satisfaction of the user. Additionally, the AR implementation may be useful for retrieving information about each car of the race or about other features of interest. Sensed, derived and explicit data can be processed. The advantage of this scenario is that anybody with a compatible mobile device can participate in an event, while it is taking place. Stakeholders can benefit by selling subscriptions to their events, as well as from promoting certain features or services. A presentation of such an application has been given at the *International*

Conference in Serious Games and Virtual Worlds (Papakonstantinou and Brujic-Okretic, 2009b).

6.1.7 Open Space Entertainment

The second entertainment scenario is mostly applicable to urban environments because there are more options available when people are geographically close to each other. This way, the application may trigger meaningful interactions between actors. This scenario can have several modes of play and specific extensions can be defined that can render it more intriguing for the users. In the action mode, AR can be used to *seek and destroy* an opponent. In adventure, location-based context is invaluable for finding and interacting with any objects of the environment (e.g. collection of a virtual item residing at a specific location). Furthermore, in adult mode social behaviour is triggered through personalised context, fed to the application explicitly by the user. A 3D urban model can be automatically generated out of spatial data and downloaded to the device, or accessed in a distributed fashion. Optimally, information about the participating characters should be modelled based on independent privacy concerns. By implementing this scenario for the AR interface, detailed texturing of the environment is no longer a necessity but becomes optional, because a photorealistic representation of the surroundings is made available by the onboard camera. Some of the advantages of this scenario are that it can run in variable-sized areas and can trigger social interaction between its users. In the literature, this kind of applications is referred to as pervasive MR games. Similarly to the previous application, an Open Space Entertainment scenario has been presented at the *International Conference in Serious Games and Virtual Worlds* (Papakonstantinou and Brujic-Okretic, 2009b).

6.1.8 Virtual Surveillance & Exploration

The scenario focuses on evocative rationale and can take place in an urban environment. Optimally, it should be applied in context-rich areas with several available objects of interest (i.e. shopping centre or marketplace). The participating entities include one or more actors. The main user can efficiently navigate with the help of the application and the rest can observe their partner or friend move in real time and in real space, from a simulated world made available to all parties. The 3D representation of the environment has to be automatically generated by a centralised system and transmitted to all parties utilising the available network infrastructure. Additionally, specific parts of the virtual world (e.g. a shop) could be manually optimised if the relevant stakeholder has invested in doing this. These places are considered as geo-bookmarks, or hotspots, and may offer further interactions, such as querying the shop stock. Furthermore, the proposed functionality offers mixed reality representations of previous and current paths that have been explored and interactions with the available POIs. Collaboration can be triggered between users, if one of them needs guidance on which location to visit in the unfamiliar environment. In such a case, the other user becomes the source of information that offers in–context advice. The main advantage of this scenario is virtual and ubiquitous presence of anybody, in a place that is being or has been visited by a familiar person. The use of an augmented reality application, in the context of virtual surveillance and exploration has been reviewed in the *International Journal of Computer Graphics* (Papakonstantinou and Brujic-Okretic, 2009a).

7 End-User Evaluation and Results

This chapter presents how the evaluation phase of the developed framework was set up in order to measure the effects of the acquired requirements. Several evaluation cycles of variable extent were accomplished during the course of the project. This section will present the scenarios that have been selected in order to evaluate the framework and describe in more detail two evaluation tasks. The first, Preliminary Evaluation, took place during the first half of the research span while the second one is an Extensive Evaluation of the system performance and usability aspects that occurred at the end of the project.

7.1 Requirements Validation Process

The final step in the process of bringing a new system to life is to validate its effectiveness against the requirements that directed the architectural design and development methodology. In the case of our framework, these requirements are separated into two categories: the requirements that have been generated by taking into account the user-expected functionality (i.e. *Requirements Acquisition Survey*) and the research reviewed more analytically in Chapter 3, as well as the system design specifications that have been produced to guide the implementation for achieving the required system functionality. The requirements, which evolved from the design specification, have been presented in more detail in Chapter 4. Likewise, Chapter 5 describes how the design specifications have been implemented in order to produce the final framework, which is the main technical achievement of this research. This chapter examines if the high-level user requirements are met by the developed system and in which way. Furthermore, Chapter 7.1.3 presents some usability factors that have been taken into consideration while designing our system.

During the course of the research, several specialised applications have been identified which our framework can effectively support. These applications, including the original ones, which have been perceived by the author, formed the main boundaries within which the framework requirements have evolved. Therefore, these applications can, in essence, form the use cases which certain aspects of the framework will be evaluated against. Another way for evaluating a solution is by examining the prototype and measuring its usefulness towards its users. Weidenhaupt et al. (Weidenhaupt et al., 1998) described how most projects combine both techniques; impact scenarios and prototype testing, because the produced results are more accurate. Consequently, an additional step is added to the development cycle, which will increase the strong points and decrease the flaws of the design. The description of the whole process, which has been followed, is provided at this point.

At the beginning of the research, the author had conceived two scenarios, according to which he believed that the system would be useful for its users. The scenario list has increased by considering other applications that the framework could apply on, which were established during the course of the research. According to this list and to users' opinions, the documentation of requirements evolved. These requirements constitute the foundation on which the development of the prototype has progressed. After the Preliminary Evaluation, which is described in Chapter 7.2, more requirements were added and the required functionality needed to be altered. Finally, in order to evaluate the developed prototype, the identified scenarios were applied.

The scenarios can evaluate the higher-level requirements, as well as the effectiveness of the prototypes. As a result, this process assisted: the enhancement of the scenarios, the advancement of the framework and the final acquisition of the requirements. This way an evolutionary development process was shaped, which the research methodology was based on.

7.1.1 Prototype for Validating Requirements

Prototyping is a time consuming process, which most systems do not actually need in order to evaluate their requirements. However, because we have been working on a socio-technical system, the development of a prototype was considered inevitable. The reasons that rendered it crucial for this project are the following. First of all, this is a completely new system whose full set of features was difficult to conceive. Consequently, the potential users and domain experts who had been interviewed were not particularly familiar with either the proposed product or the underlying technologies. This led to their experiencing difficulties determining the requirements and conceiving the system feasibility. That is why a high fidelity prototype was required, which would practically allow visualising the originally conceived framework. Therefore, the prototype can be also described as a consequence of the specifications that directed the development.

The high fidelity prototype, which was developed, provides certain advantages compared to a low fidelity prototype. The main advantage is that it can describe the complete functionality of the system as accurately as possible. It was built in a way that could render the research-based functionalities evident to an observer. Furthermore, it is user-oriented and fully interactive. Performance optimisation for mobile devices and extensive debugging provided a robust working system. Although it does offer an application navigation interface, it does not have the appearance of a definitive commercial solution. Additionally, an objective of this research project was to examine the commercialisation options of the framework, and the high fidelity prototype offered a benchmark that allowed us to compare certain aspects of our application with other available solutions.

It was mentioned earlier that a lot of effort is required to build high fidelity rather than low fidelity prototypes. The introduction of certain aspects, though, rendered the development of our high fidelity system easier and faster in certain ways. Implementing certain parts of the system, such as the AR interface and the *DirectShow* filters, as reusable components made the integration of these components to the core architecture easier and minimised the potential errors, as well as the total time required for debugging. Furthermore, the use of object-oriented languages and techniques produced reusable libraries, which have been attached to other parts of the system, apart from those that they were originally intended for. Following the same rationale, we identified and embedded supplementary software that was implemented by other developers, therefore reducing the need to *reinvent the wheel*. A good example for this case is the use of *PocketCortona* for the implementation of the VR interface, which significantly reduced the effort of producing the virtual environment and allowed us to focus on improving interaction with the user.

Apart from the benefits, our high fidelity prototype has certain compromises. We have developed only the features that have been considered crucial for the research and for satisfying the requirements. Other aspects, which are expected to be found in commercial solutions, have been selectively discarded. An example is multiuser interaction in the VR environment. System testing took place by introducing only one additional actor, although the networking protocol can be customised to support multiple connections. Another compromise has been that the user interface (i.e. menu design) of the system may be complete, but it is not very appealing. Microsoft's *Visual Studio* provided the core libraries used for the development of the application interface. For a final product, additional libraries are required to make the interface more attractive to the user. Some core usability factors that guided the design of our framework are presented in Chapter 7.1.3. Furthermore, specific parts of the framework have been more important to model making them interoperate smoothly rather than other parts, which intended to accomplish simpler, more fundamental tasks. Therefore, the control and management of errors has been very extensive and can handle almost every possible outcome.

7.1.2 Impact Scenarios

A scenario presents a comprehensive description of a specific situation. Modelling a specific scenario assists in predicting the development of a process, while reaching its ending. Scenarios can also be used to determine the impact of changes on the system. Furthermore, they are capable of verifying the reaction of the users, when certain information is exposed to them. Therefore, scenarios are useful for discovering external events, which the system must be able to cope with. Every scenario must predict the way the system will work under specific conditions and the impact that it is going to have on the user and on the related environment. Investigating the behaviour of actors, certain actions, and the environment, can produce valuable results by employing scenarios. That is why unwelcome results may be avoided, in a way that is faster compared to the actually exposing the prototype to certain lab conditions. To be more effective, scenarios should be applied and examined after the acquisition of most requirements and especially the most fundamental.

While developing impact scenarios to stress test the system, a series of actions needed to take place. Each scenario was formed by taking under consideration the changes, which the system would pose after it had been applied on the current user process. Some scenarios required a lot of changes before being finalised, while others were converted with minimal effort. The scenarios, which needed more changes, were the main source of additional requirements. Moreover, while validating the requirements, we discovered unnecessary functionalities and pinpointed the requirement, which had been its source.

The scenarios, which have been conceived and tested for satisfying the already acquired requirements, are three. These scenarios evolved by taking into account the conditions under which the system would operate, as well as the state of the users. In addition, each scenario was accompanied by a different version of the prototype, which had the requested functionalities implemented. Therefore, the evaluation of each prototype and scenario combination is presented in *chronological* order. The scenarios that the prototype system has been evaluated against are the following.

1. Virtual Reality Wayfinding (Preliminary Evaluation);
2. Virtual Reality Navigation (Extensive Evaluation);
3. Augmented Reality Navigation (Extensive Evaluation).

As we have seen in Chapter 2.5.1, a number of definitions of wayfinding involve just the cognitive decision-making process which occurs when planning movement. Moreover, the accepted definition of navigation in this project, involves wayfinding *and* goal-directed motion from one location to a new one. Although the first evaluation scenario should literally not include any user movement due to the accepted definitions, it has been named like that in order to underline that the focus of this scenario was on how did the developed system affect the decision-making process of its users while trying to establish a route towards their destination. In contrast, in the second and third scenario, the term navigation was selected because the evaluation process distinctly involved the cognitive element (i.e. wayfinding) as well as the movement in the environment. Therefore, in the latter scenarios, the relative effectiveness of *Aura's* visualisation interfaces, as navigation tools, have been compared and evaluated by examining the navigation performance of its users. The issues that have been examined during the VR wayfinding scenario, also, reflect that the focus of this evaluation task was on the cognitive aspects. Namely, these are *registration* in the virtual environment, *movement* with the device and *decision points*. Despite the fact that movement contradicts with the term wayfinding in this scenario, a closer look in Chapter 7.3.2 reveals that we examined certain issues and technical features that can affect movement while using *Aura* but we did not evaluate user performance in any way.

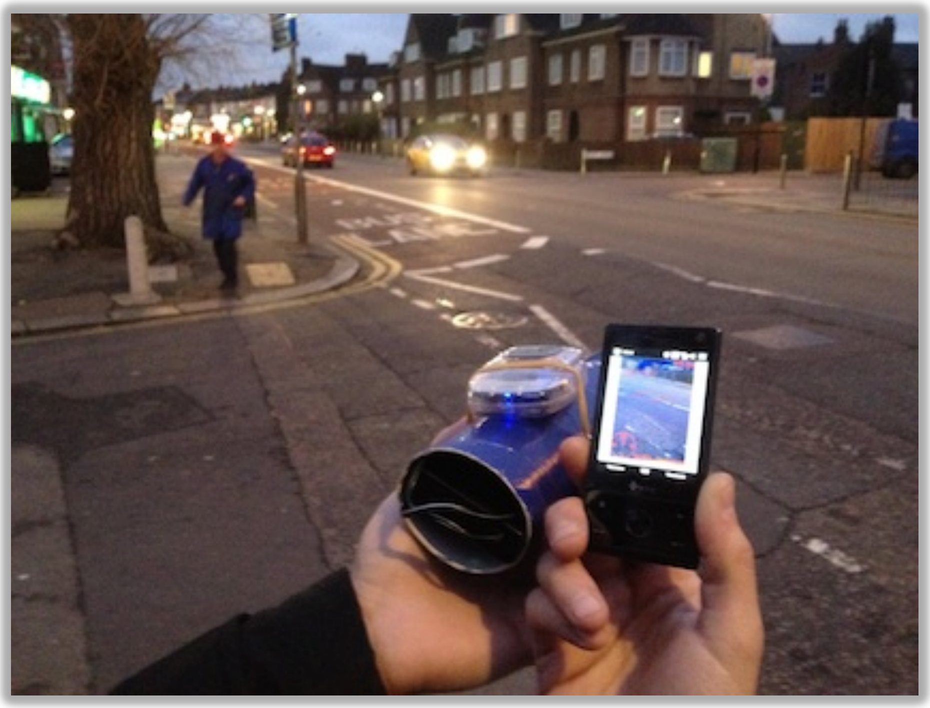


Figure 7-1: A user searching for points of interests in the environment with the AR interface

The reasons that only wayfinding and navigation tasks have been selected to form the evaluation methodology are four. First of all, the framework and its applications are considered to be capable of ubiquitous operation, which means that operation while in motion is considered as core functionality. The next reason is that the four most important rules, which describe virtual environments, explore the content, the intensity of information, user immersion and interaction features. That is why a simulation of the real world and not just an abstract representation of a confined space is a better source of information. The third reason is that the route-guidance functionalities of the framework can be reused in most other potential applications, which have been described in Chapter 6 of the report. Finally, the commercialisation potential, which has also been explored in Appendix XVI, shows that accelerating speed of entering a market is crucial. Therefore, focusing on a single market at the beginning of a commercialisation strategy may prove more efficient and yield better results.

Furthermore, for a real-time user-centred system, the evaluation of certain aspects regarding the usability of the overall architecture is equally important, especially if there is the opportunity and belief that further commercialisation activities could prove beneficial. That is why every scenario evaluation has a secondary objective, which is to examine the core usability features of the system. The following subchapter presents the factors which may enhance usability if they are intelligently implemented. As a result, the evaluation of the three scenarios was achieved by accomplishing two major evaluation tasks, apart from the minor expert evaluations, which took place due to the *Rapid Prototyping* approach of the whole project.

- The first major task, which is described as *Preliminary Evaluation* (Chapter 7.2) in this report, took place in the middle of the duration of the research, when certain technical developments, such as the implementation of one of the two virtual environments (i.e. VR), had been finalised. The results of the preliminary evaluation are presented in Chapter 7.3.
- Following next comes a more *Extensive Evaluation*, which took place at the end of the research. This evaluation task comprehensively compared the developed interfaces with regard to how they can assist a user in accomplishing a navigation task and how they help him make better decisions by exposing the right information. Furthermore, during this task most of the system functionality was presented to the users, who provided their feedback on several usability and effectiveness criteria. The *Extensive Evaluation* can be found in Chapter 7.4, whereas the results produced by this process are found in Chapters 7.6 and 7.7.

7.1.3 Usability Factors

In this section, we present some usability factors that have been identified while designing the prototype system that has been used in both evaluation tasks. Although the utility of the system can be examined by assessing the proposed functional requirements, these considerations intend to render the system more usable (e.g. efficient, effective, rewarding and satisfying). This list is a suggestion that we can use to assess and embed the required characteristics. During the evaluation of our system we did not implement every factor in the prototype. Some trade-offs have been done in order to achieve the best result. First of all, the utility of the system should not be sacrificed for providing better user experience. Secondly, we did not want to overload the user with unnecessary information and redundant interactions so that the he or she can focus on the task in hand, especially during the evaluation. Finally, not every usability factor is considered relevant to the scenarios that were selected for the evaluation tasks. For instance, some of the suggestions may be relevant for other applications, like entertainment (Boyd Davis and Carini, 2005), and may not be suitable for navigation tasks. But because the framework is intended to be expandable and customisable for satisfying other user needs as well, the additional factors are also presented. In his book (Schneiderman, 1998), Schneiderman provided eight gold usability rules that describe the key principles of interface design. Furthermore, Nielsen has presented a generic list of ten usability heuristics that can be used for improving and evaluating the design of an interface (Nielsen, 1994a). Another source of usability characteristics is Norman’s book who summarises user-centred design into seven distinct principles (Norman, 2002). Although the previous sources of usability characteristics are good guidelines for designing user interfaces, not every factor described may be appropriate for every interface and scenario combination. All of them, though, contribute positively towards the development of usable interfaces and for improving the overall user experience as described in Chapter 3.3.6. The usability factors which are presented in this section focus, primarily, on the user interface and scenario combination of the evaluation tasks and, secondarily, on other potential applications of the framework, such as those described in Chapter 6. The usability considerations are separated into four parts. The classification is not absolute because issues of one section may fit to another section as well, especially if they utilise characteristics of more than one part. The *Application* section describes usability goals which are relevant to the wayfinding scenarios that have been applied on the evaluation. The *Context* section presents some usability issues regarding the acquisition and management of context. Finally, the *Visualisation* and *Interaction* sections involve issues about the presentation of the output and specific input suggestions. The majority of usability considerations presented at this point have been either analytically or briefly examined in previous parts of the report.

Application

- Supporting all wayfinding tasks (i.e. primed search, naïve search and exploration);
- Assisting the acquisition of spatial knowledge (i.e. landmark, route and survey knowledge);
- Walking in the virtual environment is appropriate for simulation scenarios and increases user presence;
- Flying in the virtual environment is appropriate in order to explore the environment;
- Simulating temporal (e.g. sun position) and if possible other environmental information in order to increase believability and registration.

Context

- Using an appropriate tracking technique or sensor (e.g. GPS for ubiquitous operation);
- Tracking all orientation parameters in order to increase user presence;
- Tracking all orientation parameters assists the user's searching tasks;
- Providing high accuracy sensors for natural interactions to enable simulation scenarios;
- Enabling gesture interactions by identifying sequence of actions (e.g. rolling the device twice on one side changes the visualisation interface);
- Coupling the FOV with the visualisation perspective by sensing orientation parameters;
- Using low latency sensors;
- Using high accuracy sensors;
- Deriving new context from already sensed context.

Visualisation

- Providing a well-designed model of the environment (e.g. landmarks, paths, directional aids);
- Simulating physical user behaviour in the virtual environment (e.g. position and orientation);
- Validating that graphics do not obstruct the user's awareness;
- Providing informative feedback when selecting a remote entity to examine;
- Maximising the visualisation frame rate;
- Emphasising on the currently selected remote entity;
- Focusing on a remote entity should be accomplished even while moving;
- Offering accurate representation of a remote entity's position and orientation in the virtual environment;
- Minimising graphics latency and increasing frame rate;
- Allowing the users to change certain parameters (e.g. colour, avatar) of a remote entity;
- Making the avatar simulate the user's viewpoint and behaviour in the virtual environment;
- Making the avatars provide a relevant and comparable to the user's frame of reference;

- Offering an egocentric perspective when the users need to experience a strong sense of presence;
- Offering a different visualisation perspective according to required context accuracy;
- Simulating as accurately as possible (e.g. 6-DOF) the representation of a user;
- Allowing the users to manually change their representation as well as a remote user's;
- Allowing the users to manually change the bound visualisation perspective;
- Presenting remote entities that are relevant to the user's task and goals;
- Simulating a remote entity's physical behaviour in the virtual environment (e.g. position and orientation);
- Representing user interaction with remote entities in the virtual environment;
- Allowing remote entity behaviour to dynamically change according to context and user actions;
- Presenting a high-quality virtual environment (e.g. photorealistic in AR) to improve user presence;
- Presenting high-fidelity photorealistic textures in virtual environment if the scenario requires simulation;
- Presenting virtual elements that promote user activity and goals;
- Designing a graphics engine that will not sacrifice LOD for improving latency;
- Avoiding information overload;
- Naming the interface actions accurately;
- Supporting user engagement through meaningful messages;
- Offering an allocentric plan view for browsing large environments;
- Presenting context in original format (e.g. numbers) as well;
- Allowing the users to manually change the FOV;
- Minimising interface boundaries (i.e. virtual environment on full screen).

Interaction

- Supporting social interaction and communication between users in the virtual environment;
- Keeping the users informed about their current behaviour (e.g. where are they, where is the next goal, how can they reach it);
- Making the selection of a remote entity (e.g. POI, user) in the virtual environment easy (e.g. click on it in VR);

- Reducing manual interaction latency;
- Making the selection of a remote entity (e.g. POI, user) in the virtual environment easy (e.g. click on it in VR);
- Presenting all available actions associated with a remote entity after selecting it in the virtual environment;
- Offering a different visualisation perspective according to speed of movement;
- Selecting which of the available DOF to examine according to the task in hand (e.g. roll is used to interact with system or does it simulate physical behaviour);
- Removing the DOF which are not required when a specific visualisation perspective is bound (e.g. pitch is always equal to 90° when allocentric plan view is selected);
- Making the system output consistent and connected to user activity;
- Reducing cognitive load by interacting with the system through natural interactions;
- Offering more than one means of interaction (e.g. touch screen);
- Allowing the user to explicitly select either the sensor-controlled mode or the manual interaction mode of operation;
- Allowing the transition from one interface to another through natural interactions (e.g. from VR to AR when device pitch alters).

7.2 Preliminary Evaluation

Although, the intension of this evaluation task was to provide an understanding about how effective has our initial context-sensitive VR prototype been in satisfying the users' main expectations as ubiquitous mobile guide (i.e. Research Question 2), it also provide several hints about meeting the requirements proposed by the 1st Research Question of this project. For the purposes of the Preliminary Evaluation, we acquired feedback from 8 participants. We did not include the evaluation document in the Appendices due the *Thinking Aloud* evaluation strategy that was employed and its simple structure. The document was formed out of 3 blank sections (i.e. Registration, Movement, Decision Points), according to the topic that we investigated, and allowed us to record the participants' observations. The objective of the first scenario was to examine certain aspects of the system that influence user interaction performance, while being immersed in the geo-referenced Virtual Reality interface and to observe how these aspects affect their spatial cognition (Hart and Moore, 1973). The elements investigated include the ability of the users to recognise and locate elements of the environment (e.g. POIs), the ability to match their real-world orientation with the one presented on the mobile device display and the ability to cognitively correlate features of the virtual and the real world (Cheesman and Perkins, 2001). In this scenario, only a subset of the Context Management System (CMS) functionalities and only the VR interface of the Information Presentation System (IPS) of the framework had been completed and put to work. The type of feedback that was expected regarded the usefulness of the system, in order to examine the following characteristics.

- The overall user experience, based on mobile interactivity and presentation of up-to-date information;
- Discovering any potential technical issues that affect user interaction;
- Generating user ideas that may direct the development of further system functionality.

This wayfinding scenario offered valuable feedback because it examined differences and similarities between the use of a digital map and a VR interface, when used as a navigation tool. The fact that the VR interface offered, at that time, only egocentric and allocentric oblique perspectives of the simulated environment gave us the hint that by implementing the allocentric plan view, the use of the 2D map interface would become obsolete, as the VR environment would be able to cope with all required variations. This was a very interesting observation, which in technical terms reduced the footprint of the mobile application and helped us form the decision of terminating further implementation of the pure 2D component.

7.2.1 Use Case Description

The feedback that we wanted to get by applying this scenario, regarded the subjective verification of the cognitive load that was required by the users, when they needed to match the real-world frames of reference with the visualisations depicted in the mobile device and, in essence, interrelate the two worlds and their elements. In order to get a complete description, the scenario had to examine the initial registration (i.e. location and orientation) of the user, their performance while moving with the device and the decision-making ability when reaching a decision point.

The objective of the scenario instructed the user to move from one location to another. In the virtual environment, the two geographic locations were connected with a series of lines. We tried to produce a 3D model, which would have been a faithful reproduction of the real environment, by utilising the techniques presented in Chapter 5.4.1 but without having any textures attached on the façades of the 3D objects. The ideal result was to achieve a complete simulation of the user's physical state, the surrounding world elements, as well as the user-performed interactions. This means that the position and orientation of the camera in the VR world should match the position and orientation of the user in the real world. Furthermore, the main objective for the user was to reach the destination, by following the connected lines in the simulated world. In order to have the virtual position and orientation altered in real time, continuous acquisition of sensor readings was required, as well as their application to the settings of the camera in the virtual world. The virtual bearing could be altered either automatically or manually, according to the user's preference. The system did not use a dedicated orientation sensor (e.g. digital compass or accelerometer). It acquired the latest orientation parameters by examining a sum of GPS readings. We had not had this method evaluated before but during this assessment, it proved to be fast and produced accurate results, reasonable to the point that it could replace an expensive dedicated sensor. Moreover, the user had the opportunity to change the visual perspective between the egocentric and allocentric oblique options at any time, thus having multiple complementary views of the environment.

The location where the evaluation took place was the boundaries of the main City University Campus, in Northampton Square, Angel, London, U.K. The produced virtual world ran on a mobile device and clearly depicted a series of geographic positions, with distinct starting and destination points. The distance between the two POIs was approximately 500 metres. The line was not straight. Every point (e.g. corner) where the line bent was considered as a decision point.

7.2.2 Experiment Methodology

Completing the initial subjective evaluation of the system produced feedback that described the interaction capabilities and specific usability issues of the framework. At that time, the system did not focus on the requirements of any particular user group and the core functionalities could only support restricted navigational purposes. The system was exposed to a limited number of users (i.e. 8), mainly students and academic staff of City University, thus the observations presented in this chapter are preliminary. However, they have indicated new and valid directions in which further research and prototyping should follow. The Preliminary Evaluation has also offered important information to assist the transformation of the framework into a functional LBS. Before starting the experiments, each participant had been informed about the evaluation methodology and about the task that should be accomplished. For the completion of the task, every user had to follow a predetermined path represented by a highlighted line, until they reached the target.

A *Thinking Aloud* evaluation strategy was employed. This form of observation allows participants to describe their activity while they are performing, and report what they believe to be happening, including any difficulties that they face whilst interacting with the system. This qualitative evaluation type was found highly appropriate for the small number of participants that tested the prototype software. The majority of usability issues were discovered by following this testing procedure (Dix et al., 2003). In addition, this subjective study could effectively evaluate certain visualisation aspects, especially since only a small sample of expert users was involved. The users accomplished the test phase in minimum time and they reported their observations in an informal and descriptive way. When each user got informed about the scope of the testing procedure, they realised that their role was to document their natural behaviour, while interacting with the system. The aim of the supporting documentation was to clarify potential perplexities about the functionality of the application, with the intention of making participants ask as few questions as possible. The reason was to make the evaluators obtain minimum control over the users, during the actual activity.

This user-centred, practical approach (Papakonstantinou, 2005) produced a set of relevant notes. These notes were triggered either by self-motivation from the participants or by asking specific questions about the flow of events. Each user, by having a different level of experience, could summarise his or her interaction with the system. The *Black Box* technique that was utilised offered the advantage of not requiring the user to possess any low-level knowledge about the design and implementation issues of the system. This method, allowed us to retrieve important information concerning the visualisation, interaction and immersion potential of the VE, as well as the connection to the real world. At the beginning of the wayfinding task, the attached sensors were initiated and flow of data was established between the CMS entity of the framework and them.

7.3 Virtual Reality for Wayfinding Results

Although commercially available mobile device technologies have advanced, there are still limitations which prohibit full-scale promotion of complex software implementations. Specifically, for the interface design advancements defined by the research, significant processing resources are required. Additionally, due to the large volume of input information that must be processed synchronously, multitasking capabilities of the main processing unit are required. The evaluated prototype can be widely used, if dedicated graphics output mechanisms are implemented on a greater range of mobile devices. For instance, it was found that the VR interface coped well and there were only marginal differences between the observed devices, because the rendering process was software-based and it produced synthetic results. The only device that utilised a dedicated, embedded graphics accelerator scored 32% higher on average when compared to the rest. However, it should be noted that visualisation, interaction and context-acquisition have not been standardised on Windows Mobile devices up to version 7.*x*. New methodologies are sought that can manage and resolve compatibility issues. Also, in the case of variable-size displays, content visualisation should be facilitated through appropriate unifying functionalities. Due to the semi-structured (i.e. Thinking Aloud) evaluation technique that was preferred, the presentation of the collected data has not been attached in the Appendices of the Thesis. The complete analysis of the data is presented in the following 3 subsections.

7.3.1 Registration in the Virtual Environment

A point of this investigation has been to test whether users could understand their exact position in a VR scene, in correspondence to their natural position. The initial orientation and level of immersion has also been evaluated after minimum interaction with the application and understanding of the available options. The information that has been recorded by the users concerned four topics that include: *level-of-detail (LOD), user perspective, orientation and field-of-view (FOV)* (Liarokapis et al., 2006b).

Most of the participants have agreed that the LOD of the 3D environment was not sufficiently high for a navigation application. In contrast, most users disagreed about what would be required to make navigation in the VR environment easier. Some have concluded that textured, 3D models would be more suitable but others expressed the opinion that more abstract succinct annotations would also be helpful, if the task did not require absolute environment simulation. Both groups of answers can fit in the same context, if all interactions could have been visualised concurrently from more than one perspective. A suggested improvement has been to introduce *geo-bookmarks* (i.e. Hotspots) that would have embedded information about the nature of the structures or even their real purpose (Liarokapis et al., 2006a). If a single prompt solution was required for the VR environment textual annotations over specific objects could have replaced the absence of distinct landmarks.

Most users have expressed a different optimal solution about the preferred perspective for navigation, but some have concluded that more than one visual perspective of the environment would be required, to fully comprehend and align their current position and orientation accurately and fast enough. The visualisation perspective must be selected in conjunction with the level of detail of the available virtual world. Every perspective is useful for different reasons and under different circumstances. During initial registration, it is better to view the model from an allocentric plan point of view, which can cover a larger area and by minimising the LOD just to include annotations over buildings and/or roads. This way, the level of immersion, which is offered by the system, is increased but the user is not directly exposed to particular information such as the structure or façades of buildings. In contrast, the egocentric perspective has been considered very productive when the user was in constant movement. While in motion, the VR interface is constantly updated and the number of decision points is progressively increased. The results showed that further studies should be made on how the system could assist a user in accomplishing common everyday tasks, but having a variable user perspective is considered useful in most cases (Liarokapis et al., 2006a). After considering these replies, an implementation of the allocentric oblique perspective was added, which combined some of the advantages of both preceding solutions.

While operating in sensor-controlled mode, there are two options for applying the orientation parameters in the VR camera. The first maintained the users' selected heading whilst the second restored the camera to the orientation provided by the sensors. The first option was mostly selected when the users wanted to lock on a remote object and intended to move towards that direction. In such case, the heading was not simulating the user's movement but was altering according to the angle of the user in relation to the remote object. Complete simulation of the real user orientation was fully supported by the second option. This option retrieved the sensor-generated variables, transformed them to the camera coordinate system, and applied them on the orientation parameters of the VR camera in order to simulate the user's behaviour. Some users explicitly noted that inclination (i.e. 45°) towards the ground gives better appreciation of virtual navigation. Another topic, which the majority of users have agreed in, is the occurrence of fast updates. This can make it difficult to navigate, because the user needs to align the camera on three axes and not two. Based on the experiments, we have noticed that the utilised orientation mechanisms were inadequate for navigational purposes and that it was imperative that the scene should be aligned in the same direction like the device in the real world (Liarokapis et al., 2006a). After examining these suggestions, further development took place to accommodate at least the minimum needs of this issue and the new, fully automated functionality is described in Chapter 5.4.4. Likewise, new mechanisms that would assist user orientation were identified. The first solution was to present a compass at the top part of the device display. This would help more the users that have better cognitive understanding of the environment than others that do not. A compass object would help solve the occlusion problem by presenting a distinct mark towards the final destination or waypoint. The other technique has to do with the LOD of the world representation. Date and time are context variables that can be easily recorded. Consequently, a VRML headlight node could have been introduced on the sky of the VE, which would represent the actual position of the sun in the real world. This could make orientation tasks, in some cases (i.e. distinct environmental behaviour), easier to accomplish because another direct relation with the real-world conditions would be available to process.

Furthermore, all participants appreciated the user-maintained FOV functionality of the system. The control had the form of an adjustable slider bar. They agreed that the default FOV should be wide enough to include as much information on the screen as possible. They added that, in the primary viewing angle, recognisable landmarks should also be included, in order to make the user comprehend the initial position. One participant mentioned that the orientation should stay constant between consecutive decision points and that it should not be gesture-triggered. Most users agreed that the functionality of the VR interface provided adequate functions and viewing angles, which enabled recognition of selected surroundings even when they were positioned between groups of buildings with low level of detail and visibility conditions.

7.3.2 Movement with the Device

The purpose of this preliminary stage has been to explore how respondents interpreted interaction with the device while moving. The main characteristics include the *large number of updates* as well as the *change of direction* followed by the user. Other topics discussed in this section, consider the issues associated with making navigation easier, selecting the most appropriate perspective, accuracy of the underlying system, as well as the performance issues that affect the application.

A disadvantage that was observed by some participants was the lack of accurate direction symbols or waypoints that could assist route guidance. In essence, only a highlighted line was not adequate for accomplishing complex wayfinding tasks. However, this aid was considered partially inadequate because the user expected further guidance when reaching a decision point. Some participants suggested the use of arrows, on top of the route line, which would either be visible for the duration of the movement or when a decision point has been reached. Moreover, it was accurately suggested that the route line should be more evident, minimising the probability of missing it while moving. Some participants expressed the opinion that the addition of recognisable landmarks would have provided a clearer cognitive link between the VR environment and the real world scene (Liarokapis et al., 2006b). However, the outcomes of this approach have been found useful only for registering the users in the scene and not for navigation purposes. Finally, it was suggested that a *Track Log* of previously visited locations and paths, would be invaluable for post-visualisation. This was an outstanding observation that has triggered a complementary functionality because the user may become capable of viewing stored data about the whole journey and extract information about previous preferences. The implementation of log files (i.e. text and GPX) has been based on this observation.

Two participants included in their answers information about the performance parameters of the system. Both of them were satisfied from the operation of the system. This is considered as an important factor because in the VR environment the camera position changes when the external sensor processes new data. Describing the position transition process as *smooth* reflects that one of the objectives is to obtain new information about the position, at the exact time that it becomes available. The overall latency of the system is effectively equal to the latency of the hardware sensors. This means that the performance of the application is dependent on the quality of the operating hardware, with some delay introduced when run on low-end devices. The resource-intense visual environment could effectively reflect positional changes, which occur either over long or short distances. In addition, it is important to note that by implementing an allocentric plan perspective, the resources reserved for moving in the VR environment are effectively decreased. For supporting the allocentric oblique view, the camera orientation is continuously adjusted on two axes in order to to reflect the movement of an avatar that is presented on-scene and occupies the location, which is indicated by the GPS data.

Furthermore, it has been found that a different perspective should be selected according to the user speed levels. When the speed is low, the egocentric perspective should be presented because it can simulate the immediate real-world scene and help the user relate with it. On the contrary, if the user‘s speed is above a specific threshold, an allocentric plan view is more suitable because it can represent a wider area. While moving with average speed the allocentric oblique perspective has been found most suitable. This functionality may assist in the performance of the graphics subsystem as well, because when moving at high velocity, 3D drawing of the surroundings becomes an intensive process.

The opinions about the accuracy of the system differ (Papakonstantinou, 2005). One respondent was convinced that the accuracy provided by the GPS receiver was inside the acceptable boundaries, which reflected the hardware specifications. He also stated that the level of accuracy between *urban canyons* reflected the real conditions in a competent manner. On the other hand, a second participant claimed that the occlusion problem was in effect due to GPS inaccuracy issues. He stressed that, when GPS positioning was not accurate enough, the possibility to miss the route line or any other navigation-assistance method, increased. Both opinions are equally respected and the need for additional feedback is considered crucial.

The author's personal opinion on this issue is that accuracy on *X* and *Y*-axis was adequate for wayfinding tasks, even when the number of satellites and the quality of their signal touched the minimum borderline. On the *Z*-axis, positional accuracy was solely dependent on the visible satellites, but this functionality was not examined because none of the participants actually moved at any height over the surface of the earth. A new problem arose, when the developer tried to test the application to evaluate the accuracy of height. The problem had to do with the ellipsoid shape of earth. The 3D models that had been constructed were flat, which produced errors, when high accuracy was needed on the *Z*-axis for positioning the user. Thus, a new context measurement was introduced in the framework; height over ellipsoid, in addition to height over mean sea level. Further polishing of the 3D model reconstruction process improved the overall system functionality.

Furthermore, specific performance issues occurred when there was a high frequency of inbound measurements, for instance, when interaction was set to the sensor-controlled mode and the polling interval was set between 250 ms and 500 ms. At 500 ms, the problem appeared only when both logging mechanisms (i.e. text & GPX) were used in conjunction. Any lower settings than that yielded inaccuracies in other system-related operations. As a result, further improvement concerning the performance of the system was deemed necessary and this guidance has been followed in subsequent versions. Performance is mostly hardware related but certain considerations for uninterruptible operation need to be examined.

During this test, it became apparent that the use of a positioning sensor to indicate the user heading is not adequate and it should be used only as a fallback mechanism, in cases when there is no dedicated orientation sensor attached to the system. It was first seen while the participants were standing still, because there were minor position changes, which influenced the most recent direction variations. Consequently, it was concluded that orientation measurements generated by a positioning sensor should be taken into consideration, as an orientation-context source, only when the user is moving over a predefined speed.

7.3.3 Decision Points

The last stage was concerned with the decision points and the ability of the users to successfully continue interacting with the system when they reach them. A brief analysis of the users' replies has been provided in order to recognise the existing disadvantages and identify ways to improve the design.

It was in our intentions to make the user feel free to move towards any direction, without being restricted by any visualisation limitations of the computer-generated environment. However, this feature may provide exactly the opposite result. The users may feel overwhelmed by the numerous options that are available and become confused about the action that they should take next (Liarokapis et al., 2006b). We had to take into consideration that a large proportion of users are not sufficiently experienced with 3D navigational systems. Thus, some time must be allowed, in order for the users to familiarise themselves with the system functionalities.

The preliminary feedback suggested that some users would prefer the application to be capable of manipulating their perspective automatically, when a decision point has been reached. This should help absorbing more information about their current position as well as supporting the future decision-making process. Under ordinary circumstances, the actor should follow the predefined route. Nevertheless, in everyday situations the user may want to change route, in response to a new external requirement (e.g. visit a friend). These requirements are met if the user could manually add geo-bookmarks or POIs in the VE, which would actually represent points in space with supplementary personal context. To support this functionality, we introduced waypoints, which are managed by the GPX parser of the framework and are used to capture information about a location from a user.

A well-proposed solution has been to include an avatar, a sprite in human shape, which would depict the actual position, orientation and simulation of the real situation (Papakonstantinou, 2005). If this solution was implemented, the user would be able to view the avatar, while moving on top of the route line or when wandering. The avatar is available only when the allocentric oblique perspective or the allocentric plan view is selected. The avatar is considered as a reference point so that the users can match the position in the real world with the position in the VE. Without an avatar, pinpointing the exact location in the VE is not possible when the oblique view is selected, as it is presented in the following figure. In contrast, pinpointing the exact location in the VE, without an avatar, is possible when the allocentric plan view is selected because the display is centred over the exact position, but due to screen size or zoom level the accuracy may vary accordingly. Therefore, by including an avatar when the allocentric oblique or the plan view are selected, the user does not need to change the perspective to egocentric in order to gain sufficient support for the decision-making process. Moreover, the user would become capable of trusting a more abstract navigation assistance method. In this case, the feeling of freedom that dominates is preserved, but also the idea of unrestricted freedom does not overwhelm the user as it did before. Consequently, the user is free to explore the environment as well as to complete any personal task.

Figure 7-2: Allocentric Oblique View Without Avatar

Furthermore, the development and presentation of landmarks that exist close to a decision point has been suggested as a useful aid. This would make the decision point easier to identify. The use of a photorealistic Augmented Reality interface was suggested as an alternative to the VR egocentric perspective, complementing the one presented by Burigat and Chittaro (Burigat and Chittaro, 2005). The detailed analysis of all responses has been taken into account and was put into effect in further developments of the system.

7.4 Extensive Evaluation

Following a period of further development, expert evaluation and prototyping, the researcher planned a more sophisticated end-user evaluation of the developed framework. In this Extensive Evaluation, the full framework functionality was implemented on a single application, in contrast to the Preliminary Evaluation. For the purposes of the Extensive Evaluation, we acquired feedback from 23 participants. The Questionnaire that accompanied this evaluation task is presented in Appendix XIII. The application put to test was the latest version of *Aura*, which uses a combination of Mixed Reality interfaces to present real-time information to mobile individuals, on context-aware smartphones. As a result, *Aura* became the means to investigate how mobile context-aware devices can be used to assist their users by offering in situ real-time information about the immediate environment and its features. The approach for accomplishing it comes by examining the differences between the two available interfaces which facilitate interaction and visualisation in the proposed context-sensitive framework. This is the objective of 4th Research Question of this project, but this evaluation task can partially contribute to the other three Research Questions, if the framework endures further customisation with the passage of time.

The main aim of this evaluation was to assess how different interfaces for mobile, context-aware, information systems can assist their users in a variety of scenarios, but with their main emphasis on wayfinding tasks. In more detail, this process tries to identify which features of each interface provide better support to the user, while accomplishing the specified task. A secondary aim is to assess people's reactions when using their real time context (i.e. position & orientation) as a way of interacting with a mobile device. More specifically, the hypotheses and objectives of the Extensive Evaluation have been grouped to the following categories:

Effectiveness

1. To compare user performance differences between the 2 visualisation interfaces (i.e. VR and AR), while accomplishing a task;
2. To verify that the use of the system architecture (i.e. implemented features) helps the user to make better decisions, regarding the applied wayfinding tasks;
3. To identify which interface is better for making users understand the surrounding environment more quickly and comprehend its contents more

accurately, in order to become aware of the current goal and how to achieve it. The same number of entity representations should be displayed in situ on a see-through vision system and on a simulation system.

Usability

4. To identify the preferred virtual environment or technology (i.e. AR vs. VR) while moving (pedestrian);
5. To identify the preferred user perspective of the immediate environment (i.e. egocentric vs. allocentric oblique vs. allocentric plan view) while moving (pedestrian);
6. To identify if the use of AR and VR enhances the enjoyment of a mobile context-aware service;

Technical

7. To examine if the accuracy provided by the selected GPS sensor is adequate for AR and VR real time positioning. Performance Requirement 6 (PR6) defines the acceptable operating boundaries for the positioning sensor;
8. To examine if the accuracy provided by the selected compass sensor is adequate for AR and VR real time orientation. Performance Requirement 7 (PR7) defines the acceptable operating boundaries for the orientation sensor;
9. To examine if the performance of the underlying system is adequate for visualising the immediate environment and for interacting with the representations of contextual entities;

Further investigations

10. To discover new potential applications of the framework;
11. To examine if the framework is commercially viable.

7.4.1 Experiment Methodology

The evaluation methodology has two distinct forms; an objective and a subjective assessment (Gabbard et al., 2005) (Swan and Gabbard, 2005) (Dünser et al., 2008). Taking into consideration the feedback and the performance measurements of 23 people that volunteered to participate in this task has produced the results of this evaluation.

The objective measurements record the task completion time and distance covered by each user, as well as the coordinates of the points that have been occupied (i.e. user track). Furthermore, by exploring this information, we can derive the minimum, average and maximum user speed while accomplishing each task. This study employs a statistical analysis of the recorded variables and includes a descriptive analysis of the results. The objective assessment examines the user's task performance, within the context of each interface and environmental conditions, in order to understand how the system creates an impact on the accomplishment of the underlying task (Pingel and Clarke, 2005).

The subjective assessment studies the users by utilising a questionnaire, which presents the user ratings or judgements, and through an interview with the evaluator that took place while filling in the questionnaires. This study employs a statistical analysis of the feedback and a more descriptive interpretation of the results. The subjective assessment focuses on identifying issues regarding the usability of the system (Hix et al., 2004). It has been accomplished by considering the answers provided in the following domains: *Usefulness*, *Ease of Use*, *Ease of Learning*, *Satisfaction* and *System Performance*.

7.4.2 Ethical Issues

Although there were no major ethical issues to consider while accomplishing this research task, two minor issues had been identified and needed to be addressed. The main ethical issue was to communicate to the participants that we were evaluating the performance of the mobile applications and not their own personal performance. It was underlined that there is no such thing as doing *badly* at the test. If they had felt the test had not gone well and the task had taken them longer than it should have, then this would have been reflected upon the application, which was captured as part of the evaluation and does not in any way constitute a criticism of their abilities. We were interested in the reactions to these prototypes, whether these were positive or negative. By taking part in this test, the participant would help to improve the usability of the mobile applications. Thus, the first ethical issue was to communicate that we were testing the application and not the people. The second ethical issue was that while conducting the exercise, the participants' trajectory was recorded by using the position determination system (i.e. GPS) integrated to the mobile apparatus. It was made clear to the participants that their location was being recorded, and that this was done with the sole intention of evaluating the effectiveness of the mobile applications that supported their navigational task. The participants were also advised that their location would only be recorded whilst they were carrying the device, and that all recorded positional data would be anonymised before testing. For the purpose of this research task the researcher had to get the approval of the governing university body, which was the *Research Ethics Committee* of the School of Informatics, at City University London. The approval to proceed with the execution of this study was granted on the 22nd of September 2010.

Figure 7-3: A participant familiarising with Aura

Apart from the ethical issues, some minor health and safety risks associated with being outdoors in an urban area were identified, in order to ensure the safety of the participants. These were potential trip, slip and fall hazards. Thus, the main responsibility of the researcher, while conducting the experiment, was to accompany the participants and warn them if they were about to bump into anything or trip over something on the ground, while executing the task. Furthermore, in cases where the participants had to walk in the middle of a road, the researcher had to examine the environment for any potential vehicles or pedestrians that would be in a collision course with the participants and notify them, in order to act accordingly (e.g. pause).

7.4.3 Use Case Description

For each participant, the duration of the data gathering and user feedback tasks required 1 hour 15 minutes on average to complete. During that time each participant had to accomplish a series of tasks. Before beginning the actual testing, everyone was asked to complete a questionnaire designed to assess the background experience and familiarity with the concepts, technologies and applications, which have been investigated in this project. This way, we would be able to assess the expertise of each participant and assist the statistical analysis by forming different groups according to the levels of familiarity.

The main part of the testing procedure consisted of two subtests, lasting approximately 20 minutes each. For every test, the participants had to complete a simple wayfinding task. The mobile device was utilised as the primary navigation aid, in order to follow one of the two pre-defined routes, from point *A* to point *B*. In each test, the system configuration and interface was slightly altered. Every user had to follow one route by using the Virtual Reality interface and the reverse one by using the Augmented Reality interface. The order in which they had to accomplish each task was swapped, according to the preferences of the previous participant. It was planned that way in order to receive more accurate results and reduce the bias by avoiding familiarisation with the application or the environment. As a result, 12 users accomplished the first wayfinding task by using the VR interface and the second task by using the AR interface. On the contrary, 11 users started by using the AR interface and finished with the VR interface. The wayfinding task that they had to accomplish involved *reaching* a series of waypoints, which were located in ascending order, until they *reached* the last one in the queue. In order for the participants to *reach* a waypoint of the track, they should have approached it until they landed within a 5 meter radius from the predefined target point. The range of 5 metres was calculated on a 3D coordinate system. The applied distance constant has also proved very effective due to the variable accuracy of commercial positional sensors that could affect data collection. The concept of *reaching* a waypoint had an additional factor for every interface attached. These factors will be analytically presented in Chapter 7.4.4, in relation to the governing interface.

Following each test, the participants were interviewed to assess the usability issues of each interface and how effective the application was in helping them to complete the task. The feedback was recorded on a questionnaire, which presented the same questions for both interfaces. After completing both tasks and being exposed to both interfaces, the participants were also questioned about their overall experience and preferences. These observations were particularly useful because they presented the opinions of each user regarding the operation of the whole system and further potential applications that it may have.

During the experiment, the researcher accompanied the participants throughout the tasks. His responsibilities included the provision of background information, description of the tasks, answering questions and looking after the participant's safety. He also conducted (and recorded the responses to) the follow-up questionnaire and interview. Additionally, he provided technical support and collected data by observing the users. The following list describes more analytically the duties that the researcher had to carry out in order to complete the evaluation process per participant.

1. To turn on the GPS receiver so that it could acquire a position fix;
2. To collect the consent formed and verify that the participant had signed it;
3. To fill in the first part of the questionnaire regarding the previous experience of the participant.
4. To verify that the batteries of the GPS, digital compass and mobile device had been charged at a significant level (i.e. at least half of their capacity);
5. To verify that the Bluetooth interfaces of the GPS, digital compass and mobile device had been switched on;
6. To initiate the software application (i.e. *Aura*);
7. To verify that the sensors had been connected and produced (valid) data;
8. To setup the Virtual Environment, either VR or AR;
9. To load the waypoints that should be reached (i.e. GPX file);
10. To start recording the track and time (i.e. *Aura*) at the beginning of each task;
11. To stop recording the user track and time (i.e. *Aura*) at the end of each task.
12. To fill in the second part of the questionnaire regarding the wayfinding task and interface combination;
13. To fill in the third part of the questionnaire regarding both wayfinding tasks and interfaces.

7.4.4 Experimental Environment

During the evaluation, the wayfinding tasks that the participants had to carry out involved walking on a predefined route. The area, which had initially been selected, was in the boundaries of City University London, Northampton campus, London, EC1V 0HB, U.K. The selection of this area proved to be inappropriate due to several reasons. To begin with, the period during which the evaluation took place, was at the beginning of a new academic semester (i.e. September and October 2010) and the campus was very crowded due to the arrival of new students. The second reason that required the change of the testing site was that the campus area (i.e. Northampton Square) does not support the design of a route, which can objectively assist the conceived wayfinding task. The main requirement for the route was that it should have had the same length and number of turns (i.e. decision points), in both directions. Designing a route like that around the university campus would have increased the potential risk, which the participants would have been exposed to. This means that they would have had to cross busy roads or be exposed to danger, which would subsequently have influenced the results of the evaluation. The final reason that directed the change of location involved technical issues that were identified during the design of the process. In more detail, the university buildings, as well as the buildings that surround the campus are very tall, which may have significantly affected the time for the first fix (i.e. cold boot) and the accuracy of every GPS receiver that has been pre-tested to work in that environment. Furthermore, the architecture and construction material (e.g. glass) of some buildings yielded inaccurate results due to GPS multipath errors.

As a result, the location that the experiment took place was changed. A map describing the exact area can be found in Figure 7-4. The new location was selected according to the following criteria. The area should have to be relatively quiet, without having a lot of people passing by. Thus, the selection of a residential area was found more suitable for accomplishing the wayfinding tasks. The area should also be reasonably safe for the participants, which required a relatively small number of cars and bicycles passing by, in cases where the user would have needed to cross a road. The first two requirements had been fully met, for almost every day that experiments were being conducted, until 5.00 pm. After that time, the streets were relatively busier due to the start of the rush hour. The next requirement was that the architecture of the buildings would not pose issues relating to the accuracy of the positional measurements, which was the most sensitive context variable processed by the system. Even though it is a dense urban area, the height of the buildings did not restrict signal reception and the GPS receiver required, on average, less time to acquire a valid position fix compared to the previous location. It has been noticed that the GPS cold-boot latency mostly affected the first participant of each day. After receiving some redundant measurements, the sensor proved to work more efficiently for the people that followed on the same day. The selected area is located behind Angel Tube station, in Islington, London, U.K.

Figure 7-4: The area in which the data collection has been conducted (Google, 2011b)

The total distance on ground, which the participants needed to cover for each wayfinding task, was 438.05 metres. The track has been designed in a way that would pose only minimal risk to the participants. This means that they would not have to cross any busy roads or be exposed to danger. The position coordinates of the track under test had been acquired by meeting this requirement. Thus, both tasks could be accomplished by walking on the pavements along the streets of the area. Additionally, the researcher was observing the participants during the task, assessing their safety at all times. The participants had been made aware of any potential hazards, such as proximity to busy crossroads, cycle paths and protruding tree roots. The selected environment was particularly safe since good lighting and open spaces minimised the chance of trips, slips or falls occurring. The track is comprised out of 13 distinct waypoints and their layout has been presented on Figure 7-4. A more detailed description of the waypoint coordinates can be found in Appendix X. For each wayfinding task, the participants had to start from *Waypoint 1* and reach *Waypoint 13*. After completing the first route, they had to complete the route a second time, but in the reverse direction. In that case, the waypoints were automatically renamed in order to reflect the actual progress (e.g. the name of *Waypoint 13* became *Waypoint 1*). Every participant, though, accomplished the first test by starting from Waypoint 1 and the second test by starting from Waypoint 13, as depicted in Figure 7-4. Each test was accomplished by using a different visualisation interface, either VR or AR. During the experiment, we identified two small areas, in which the accuracy of the GPS receiver was evidently reduced. This occurred due to the fact that tall trees are found in these areas. The location of the first area lies between *Waypoint 5* and *Waypoint 6*. The second area is close to Waypoint 9. The decision of not trying to solve this issue was made because it would not reflect the actual conditions, which affect urban areas. The trajectory of the track has been designed in a way that each participant would have to make the same number of turns for every task and in each direction. More specifically, the route included two left and two right turns.

Apart from the location, other environmental conditions that contributed and affected the course of the experiment were time, weather and visibility conditions. The plan was to conduct all wayfinding tasks during daytime with bright light. This requirement was achieved in most cases (N=21) except 2, which took place while dusk was growing. Furthermore, the weather conditions and especially the sky were typical for London, during the period that the data collection occurred, which spanned across the 4th of October until the 20th of October 2010. In 9 cases the weather was sunny, in 2 cases it was partially cloudy, in 9 cases it was overcast and in the final 3 cases it was raining mildly. The visibility conditions proved ideal (i.e. clear line of sight, without any fog) for every participant that executed the wayfinding tasks.

7.4.5 Experimental Apparatus

The experiment involved three dedicated devices, which the participants had to carry while accomplishing both wayfinding tasks. The devices were a Windows Mobile phone, a GPS receiver and a digital compass. The selected Windows Mobile phone was an HTC Touch Diamond, with a 528MHz processor. Unfortunately, we could not acquire a device that embedded the newest Windows Phone 7 operating system, which had been just released (Q3/4, 2010). The selected device was used in various stages of the research and maintained a good performance-to-size ratio for the task that we needed it for. Although the smartphone had an internal GPS receiver and accelerometer, two external sensors were selected for the experiment, because their accuracy was significantly better. Another reason that forced us to use external sensors was that we had to watch out for the battery level of the smartphone, especially on the days that we had to assess more than two participants. Both sensors were connected to the phone via Bluetooth. The GPS sensor was a Pharos iGPS-BT 360, which is a common commercially available receiver. The following figures show the mobile phone and GPS receiver, which have been utilised for the experiment.

Figure 7-5: a) HTC Touch Diamond (P3700) b) Pharos iGPS-360 sensor & Bluetooth adaptor

The digital compass circuit was custom made, based on the Honeywell HMR3300 magnetic sensor chipset. Due to the cost of purchasing a release version (i.e. standalone product) of the digital compass, we acquired the PCB and assembled the rest of the circuit. The whole circuit is constituted out of two major components. These are the magnetic sensor PCB and the Serial-to-Bluetooth adaptor for transferring the data to the phone. The two components communicate over the RS232 protocol by connecting their transmission (i.e. TX) and reception (i.e. RX) channels. Furthermore, two sets of batteries, four for the Bluetooth adaptor (i.e. 4x1.5V AA) and one for the magnetic sensor (i.e. 1x9V), were attached through a custom- made circuit. The following images illustrate the individual components of the digital compass, as well as the setup of the test system.

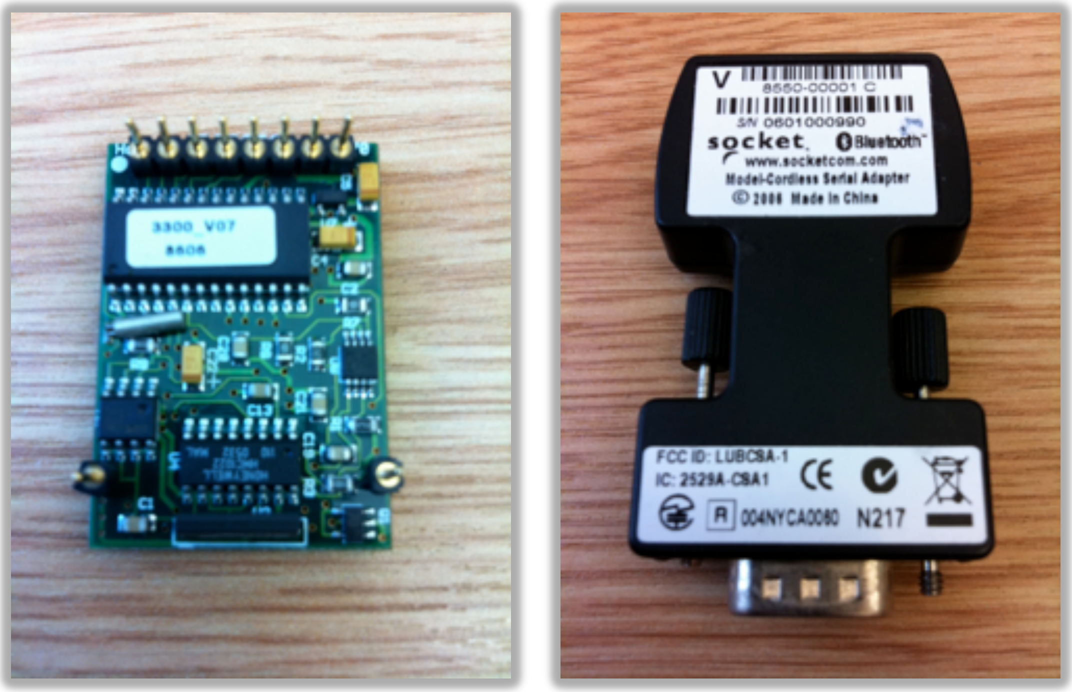


Figure 7-6: a) Honeywell HMR3300 sensor b) Socket Cordless (Bluetooth) Serial adapter

Figure 7-7: a) Component Assembly b) Test Apparatus

7.4.6 Experimental Software

For evaluating the usability of the system and for comparing the performance of both visualisation interfaces on a wayfinding task, the most recent implementation of *Aura* was utilised. It was release version 0.6.53, offering full functionality. The full source code of the version of *Aura* that was used in this experiment has been attached to Appendix IX. Although this version had integrated every feature that was described in the implementation chapter, for the purpose of the evaluation only a reduced set was required. The features, which were not exposed to the users, were those that were not found crucial for assisting them in the task that they had to accomplish. The networking component was not put to work because it would overcomplicate the task for the users and it would also require another device to participate, having a specific role (i.e. actor) in the scenario. It would also render some statistical measurements inappropriate because the participants would be dependent on another person, who could ultimately affect their performance. Furthermore, the 2D/Map interface was not used at all, because the allocentric (i.e. birds-eye view) perspective of the 3D/VR environment was a more elaborate substitute that could represent changes on every axis (i.e. 6-DOF).

Figure 7-8: The Egocentric perspective of the VR interface

The participants had the opportunity to interact with the system, principally, through the visualisation interfaces. Each interface (i.e. VR and AR) provided a digital representation of the physical environment and by taking advantage of its special features it provided guidance on how to reach the next waypoint of the route and subsequently assist the users in reaching their final goal. In chapter 7.4.1 we mentioned that each interface had an additional requirement in order to *reach* the following waypoint. This redundant functionality was implemented only for the purposes of the evaluation because we had to be sure that the users completed each milestone sequentially and not arbitrarily. We also had to be sure that they did it intentionally and also because they needed to be informed about their progress. These reasons instructed a minor deviation from the normal functionality, which would not else be required. Thus, in order for a user to *reach* a waypoint in the VR interface, he or she should have moved in less than 5 metres distance from the POI and also clicked on the message box, which appeared for this reason. The message box provided a brief description about the accomplishment of the goal and about the next target that the user should *reach*. Alternatively, in the AR interface, the users had to get close to the waypoint, at the same distance as in VR, but they should also have it inside the conceptual field-of-view polygon, which was generated by the device and sensors. If the position coordinates of the POI lay between the boundaries of the polygon then a message was embedded on the video feedback, presenting the same information as the similar message did for VR. For the VR feedback message, the users should press the *OK* button in order to be allowed to continue. Conversely, the AR feedback message was displayed on the screen for 7 seconds and during that time the users were not able to continue with their task because information on how to reach the next waypoint was not visible. These delays should be mentioned, because they occurred for every participant 13 times, once per waypoint and for both tasks.

Figure 7-9: The Allocentric Oblique perspective of the VR interface

In order to reach their goal, the participants had to act in a different way and process different kind of information according to the utilised interface. In VR, the waypoints were represented as distinct yellow balls and they were connected with a rectangular 3D *carpet*, which was presenting the ideal trajectory for reaching the following waypoint.

The *carpet* was dual-coloured (i.e. red and green) in order to be more conspicuous, even when it was dark and even if it was overlaid on top of grass or brick walls, respectively. The users had not been exposed to any numerical information, such as the distance to the next POI or orientation data. The physical behaviour was simulated in the virtual environment according to the calculated 6-DOF. The supplementary representation of the track was loaded automatically at the beginning of each task by reading a GPX document, which had been pre-configured by the author and described the waypoints of the track. The location of each waypoint entry found on the document was loaded in VR and it was translated to a yellow ball. The production of the *carpet*, which connected every waypoint, was automatically accomplished by *Aura* by accumulating the POI coordinates. The representation of the physical environment was produced, by following the technique described in Chapter 5.4.1. The 3D model was textured, but it was also generalised in order to demand minimum operational resources. Also, the image quality of the textures was significantly reduced. The feedback of the users included answers regarding how the LOD of the 3D model and its features affected their performance on the applied task. Finally, users had the opportunity to work with every perspective of the VR environment (i.e. egocentric, allocentric oblique and allocentric plan view) and select the one that they preferred in order to reach their target. At the beginning of the VR test, each participant got informed on how to alter between the available perspectives so that they could accomplish it by themselves during their journey. There was not any software mechanism that objectively recorded which perspectives were selected during the task, although it would have been a very nice idea for further investigations. Each user's preferred perspective has also been measured by their subjective responses and it has been presented in the following sections. In order for the users to complete the wayfinding task in VR, they should have tried to get close to every waypoint of the track until the message box appeared on screen, then clicked the confirmation button and repeated this process for every waypoint until the last one, where they received a concluding message. The next figure shows a screenshot of the actual VR environment, including the waypoint and track representations. The ball that is closer to the user is the representation of *Waypoint 3* and the one that is illustrated at the back represents *Waypoint 13*.

Figure 7-10: VR wayfinding in Aura (Waypoint 3)

The operation and functionalities provided by the AR interface were not the same as those provided by the VR interface. One of the advantages of AR is that it does not need additional data in order to recreate a digital representation of the immediate environment. This is also the reason why the cost of development is significantly lower compared to VR. The camera of the mobile device has been used to capture the environment and the digital representations of any features of interest (e.g. waypoints) were superimposed on the device screen. How, for how long and at which part of the display real estate these features were represented, was determined by the sensor data. In contrast to VR, graphics on this interface were not very intense. We tried to reduce as much as possible the use of redundant graphics and replace them with informational descriptions about the objectives of the scenario. This information included numerical descriptions of the position coordinates of the user and of the next waypoint, the distance between them, and also the user's orientation. The only graphic that was presented on the bottom left part of the screen was a mini-map that depicted the heading of the user and the location of the next waypoint. The process of locating and reaching the waypoints in AR consisted of the following tasks. Initially, the users should have rotated themselves until their heading matched the direction of the next waypoint. This was communicated by embedding a descriptive message on the video. This message appeared only when the next POI lay in the user's field of view. The distance, which was scanned, was set to 200 metres for the purposes of the evaluation, but it could have been manually or even automatically adjusted to the distance of the closest POI. After the user had *locked* to the next waypoint, he should have started to walk until he landed in the target zone (i.e. a 5 metre radius from POI). The distance indicator was continuously offering updated feedback. When the user was in the vicinity of the waypoint and the system setup was facing it, a new message was presented, which provided information about the achievement and about the next objective. This message stayed on screen for 7 seconds and after its disappearance the user had to repeat the same process for the following waypoints. Similarly to VR, the waypoint details were loaded from the same GPX files. For the purpose of this evaluation, two GPX files were developed and used. Their only difference between them was the naming of each waypoint, according to the requirements of the task. The waypoint coordinates in both files were identical. These GPX documents have been attached to the 10th Appendix.

7.4.7 Participant Information

This section presents the selection criteria, which have been applied in order to determine the sample size and it provides a description of the participants' background. The sample group consisted of 23 participants with no specialist expertise in the fields explored by this research, although all of them were familiar with the use of computers and mobile phones. The researcher was responsible for finding volunteers that wanted to participate. Most of the participants were university students (i.e. 8 postgraduate and 8 research) and 2 were members of the academic staff. This group was either approached in person or via emails sent by the researcher, who politely requested them to take part in this study. The advantage of this group was that the people constituting it had an established academic background on the use of computing systems. Furthermore, participants also included people without any relation to the university (N=5). This group proved not to be ideal because we needed to establish their background knowledge before conducting the evaluation and as a result we had to reject some people, who only had very limited experience with information technologies. Although the developed system is intended to fulfil the information needs of people from various backgrounds, the purpose of this evaluation task is to examine the system as a mobile guidance tool. The application depends on the user's physical location and POIs in the immediate environment and is also intended to be used while users are moving to new locations. These issues rendered the evaluation of the system challenging both for the researcher and the participants. Furthermore, examining the participants' interaction with the system is a very time consuming process and can be affected by users who are unfamiliar with the process (Kjeldskov et al., 2005). Nielsen has examined in detail the cost and benefits between end-user evaluations against expert evaluations (Nielsen, 1994b) and we are not going to explore this topic further. But due to the scope and the aforementioned particularities of this project, we tried to conform to a certain user model. As a result, the selected sample was formed neither by expert users of the applicable technologies nor from users without any substantial relevant knowledge. We regard our sample as *regular specialists* (Nielsen, 1992) with UI and usability experience, but no expertise in the system functionalities or explicit knowledge about the underlying technologies. In Nielsen's study, finding around 80% of usability problems required 3–5 regular usability specialists. This kind of proficiency was helpful due to the small number of participants who provided feedback that can assist the agile development process and provide contextual information as well. Finally, because this evaluation task involved both objective and subjective tests, we believed that by selecting an IT-literate group would provide more comprehensive and consistent results.

While recruiting the participants, the researcher provided a brief description about the topic of the evaluation and the task that they would have to accomplish. Additionally, the researcher tried to establish an opinion about the physical and mental suitability of each participant as well as their previous expertise in the topics covered by the research. This was accomplished by asking them a question regarding any physical or mental issues that might restrict or endanger them while conducting the exercise. Furthermore, it has been verified that every participant owned a mobile phone and was familiar with the majority of its features. The participants had been informed in advance that the assessment would be conducted outdoors and that they should be suitably attired. Due to the fact that the exercise involved small display devices the participants were asked to bring their glasses if they had problems with their eyesight. None of the candidates mentioned having major vision-related issues that could affect the results of the test, nor did express any problems while conducting the test. The participant's eyesight could not be objectively measured, because it would require a special approval by the Ethics Committee due to the fact that is considered a health issue. If they agreed to take part, they received the *Participant Information Sheet* found in Appendix XI, which presented a detailed description about the study, as well as 2 identical *Consent Forms,* presented in Appendix XII. At that point, the date and time that the test would take place was established. Between the booking and the test session, the participants had enough time to study the description of the evaluation process and fill in the consent form. On the day of the evaluation and before commencing the assessment, if the participant was still interested in taking part, he or she had to submit one of the *Consent Forms* to the researcher. Another copy of the form was available on site, in case the first one got lost. Furthermore, if any of the participants appeared either before or during the testing to be in any form of distress, such as suffering from cold or fatigue due to the travelled distance, they were advised that they could withdraw from the exercise at any time. In addition, if we identified a potential risk or issue regarding a participant, the evaluation would immediately terminate and the participant would stop being used as a subject.

Initially, we wanted the participants to reside in a certain age group. That is between 20 and 40 years old, with no participants younger than 18 years or older than 65 years of age. Additionally, we aimed for a near to equal gender split, but the majority of participants were males (*N*=16, 69.6%), in contrast to 7 females (30.4%). The following box plot shows a comparison between the gender and age of the participants.

Figure 7-11: Box plot of Participants' Age and Gender

Several studies have proven that wayfinding behaviour may be affected by a number of reasons. Malinowski and Gillespie concluded that sex, previous experience, mathematical ability and map-use skills are significant predictors of wayfinding performance (Malinowski and Gillespie, 2001). Lawton has also examined gender (Lawton, 1994) and cultural (Lawton and Kallai, 2002) differences that can affect wayfinding strategies and concluded that females prefer to apply a route strategy, in contrast to men who prefer to apply an orientation-based strategy on wayfinding scenarios. Furthermore, O'Laughlin and Brubaker validated the gender performance differences after applying a mental rotation test while examining the use of landmarks in cognitive mapping (O'Laughlin and Brubaker, 1998). In a very interesting review on gender differences in spatial orientation (Coluccia and Louse, 2004), the researchers concluded that differences emerge only when wayfinding tasks require a high load of Visuo-Spatial Working Memory (VSWM). The authors support that "*(...) the VSWM load could be a determinant factor, able to increase or level off individual differences in orientation abilities. Males would show better orientation performance, because of their larger VSWM span. When the orientation task does not involve a high load in VSWM, gender differences would disappear*" (Coluccia and Louse, 2004). Rubio et al. evaluated three methods that can examine a user's cognitive load (Rubio et al., 2004), such as NASA's Task Load indeX (NASA TLX) test (Hart et al., 1988) (Hart, 2006). Furthermore, Hegarty proposed a methodology for establishing a participant's spatial abilities from a subjective test that is called Santa Barbara Sense of Direction Scale (SBSOD) (Hegarty et al., 2002). During the evaluation of the LOCUS project, both SBSOD and NASA TLX tests were applied because the tasks required high load of Visuo-Spatial Working Memory from the participants. In the evaluation of *Aura*, we chose not to apply any of these tests because the users were not required to recall the locations that they were visiting. In fact, because in both experiments they had to follow the same way twice, we did not want them to remember the route. Apart from following the navigational aids, the comparison between the two interfaces did not require the participants to make any other important decisions about the task. Although the deviation may have affected the results, the aim of this study was not to compare the performance of different gender or age groups, and such study would require a more strict approval of the University's Ethics Committee about data handling and applicable privacy issues. The range of ages for both gender groups is from 24 to 42 years, with a mean of 30 years for females, 32.19 years for males and 31.52 years for both. The median age of both groups is almost identical with values of 30 and 30.5 years, for females and males respectively.

7.5 Exploring Previous Experiences (PE)

The first part of the questionnaire was designed to acquire the previous experience of the participants in certain fields relevant to our research. The full set of data can be found in Appendix XIV. The following bar charts illustrate how the participants rated their familiarity with VR (PE-1) and AR (PE-3) interfaces, as well as with the use of real-time context-aware systems (PE-5) and the level of their computer skills (PE-8), amongst other questions. This way, we could verify that the sample group was in the boundaries of our initial target. The selected sample group has had, at least at some point in their lives, some experience with the concepts and technologies explored in the following sections.

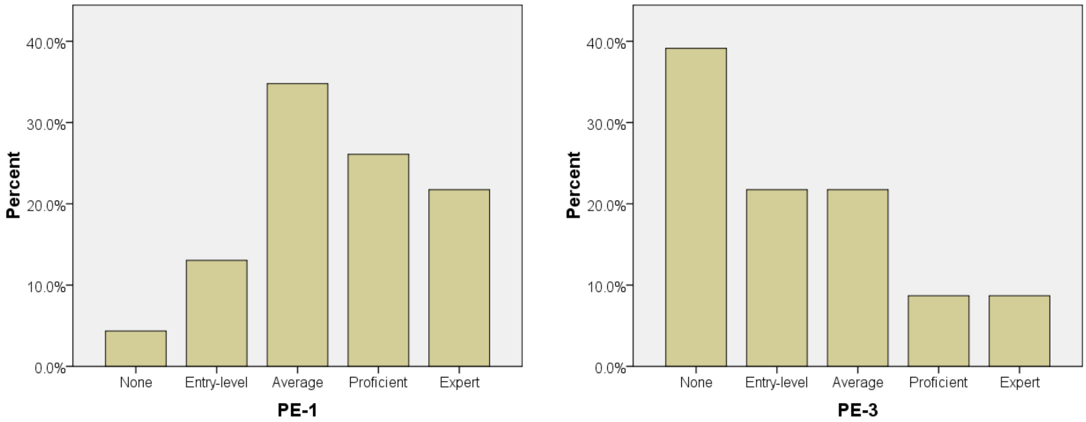


Figure 7-12: Previous experience with a) VR (PE-1) and with b) AR (PE-3)

From the output presented above, we know that more people had previously used an application that represented information in a 3D manner, rather than a video see-through system. More than a third of the participants (39.1%) had never worked with or may not have seen an AR application before, whereas in the same category for VR the score is 4.3%. Those that had some experience with VR applications provided some feedback regarding those applications. Five people had visited a 3D cinema, 14 had played a 3D game, 8 had browsed a 3D map, 5 had navigated with a 3D SatNav and 5 did some sort of VR programming. Three persons replied that they had obtained experience through a research project, biometric application or other mobile applications, respectively. In contrast, for AR, 4 persons had used some sort of POI locator, 4 had been to a simulator (i.e. F1 or Flight), 3 had played an AR game and 2 had observed a research project. Moreover, 3 persons replied that they had obtained experience through art installations, biometric applications or an elaborate aircraft system.

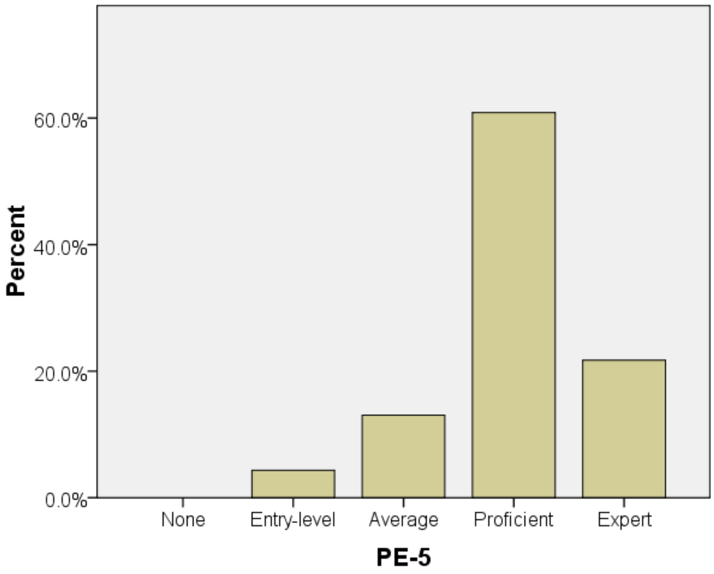


Figure 7-13: Previous experience with real-time, context-aware applications (PE-5)

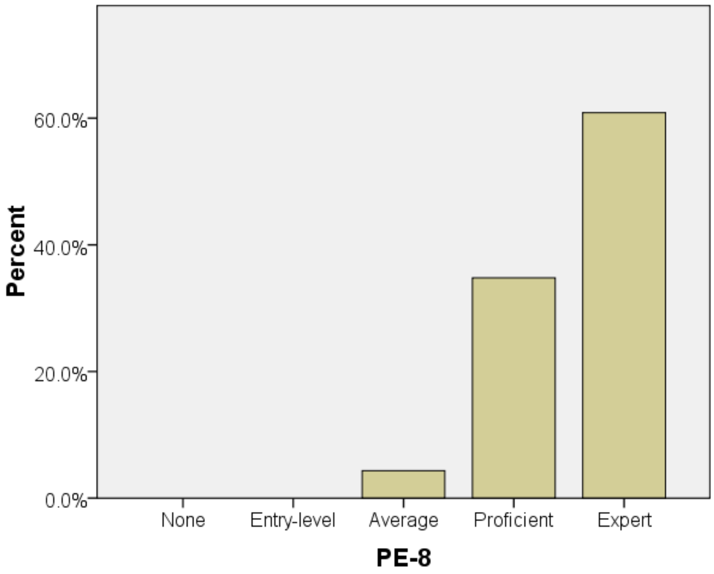


Figure 7-14: Level of computer skills (PE-8)

As we can see from the chart in Figure 7-13, the participants rated themselves quite generously regarding their familiarity with real-time context-aware applications. Every person had used at least once an application that processed real-time context, whereas 82.6% felt that their expertise was more than average. The applications, mentioned by the participants have been using several types of context. From the collected responses, 18 applications were dealing with location context and 17 were using personal information in order to enhance an existing process. Additionally, 14 responses described an application, which operated in a commercial environment (e.g. shopping) and 2 were research-oriented applications. Furthermore, 95.7% of the participants felt that the level of their computer skills was more than average. This high score reflects the decision to select well-educated participants for this evaluation. Although the majority of the sample cannot be considered as expert users, the feedback provided by them will be beneficial for the quality of the evaluation results because they have an established level of expertise.

The results of the survey analysis, presented in Chapter 3, that has been accomplished in order to gather the requirements of the framework and the results described in the previous paragraphs, have shown that most people had the opportunity to experience a Virtual Environment (i.e. mostly VR) through 3D games. Furthermore, the most commonly utilised context-aware applications are those that employ position information. The last 4 diagrams produced by this survey explore these observations.

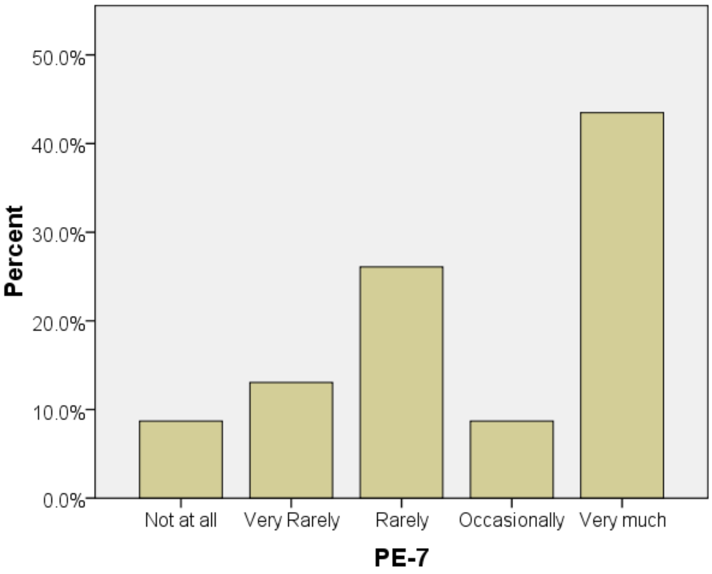


Figure 7-15: The extent of Smartphone use in the participants' daily activities (PE-7)

The previous bar chart demonstrates that the use of more advanced mobile devices (i.e. Smartphones) and services has grown in the U.K. and in the boundaries of E.U., which was an expected consequence since the initiation of this research project. The participants were informed that question PE-7 described the use of functionalities excluding making and receiving phone calls. The provided feedback revealed that 21% does not have or very rarely uses the advanced functionalities of their smartphones during their daily activities.

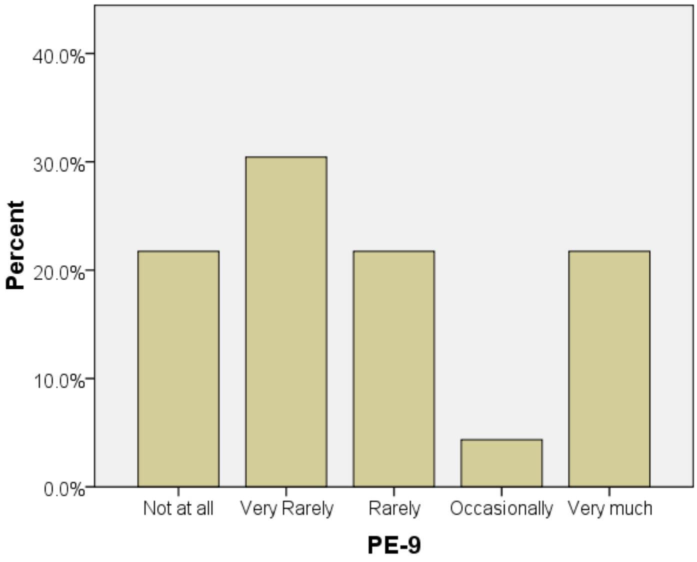


Figure 7-16: Responses on how often participants play Computer Games (PE-9)

In Figure 7-16, we observe an almost equal distribution of the proportion on every scale except occasional frequency, which was offered as an answer by only one participant. 21.7% of the participants has never played or does not play computer games anymore. From the rest of the answers we can deduce that the majority (43.4%) plays games rarely or less rarely. If we take under consideration the fact that the median age of the participants was 30 years, we can assume that if they still play games, they must have played earlier in their lives and will still play in the future but with variable frequency. Thus, we consider that the majority of participants have played at some point in their lives a 2D or 3D computer game and that they understand the concept of achieving a goal through accomplishing smaller tasks in a simulated environment.

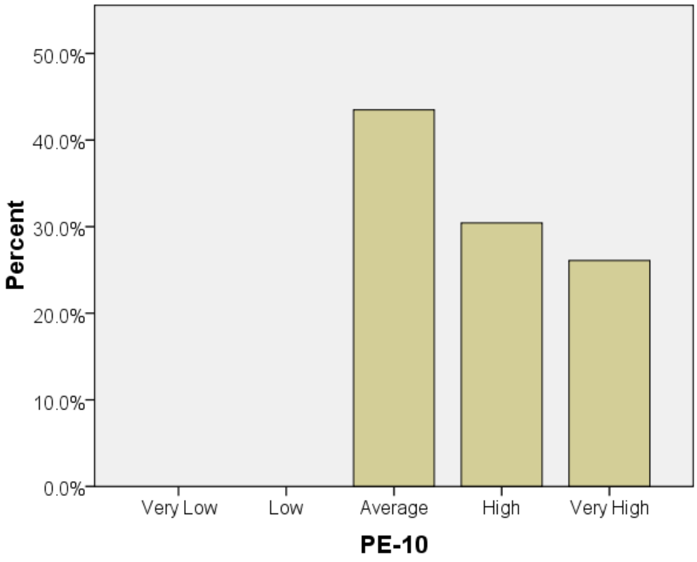


Figure 7-17: Subjective rating on the level of participants' Sense-of-Direction skills (PE-10)

As the functional objective, which the participants had to accomplish, was a wayfinding task, we introduced a question regarding their skills on that field but, specifically, in the physical environment without obtaining the assistance of any navigational aid. All participants rated themselves as having average or more than average sense of direction. But without a more objective evaluation (Hegarty et al., 2002), accurate results cannot be produced. Thus, we assume that the sense of direction is average for the majority of the participants, but keeping in mind that lower scores probably exist.

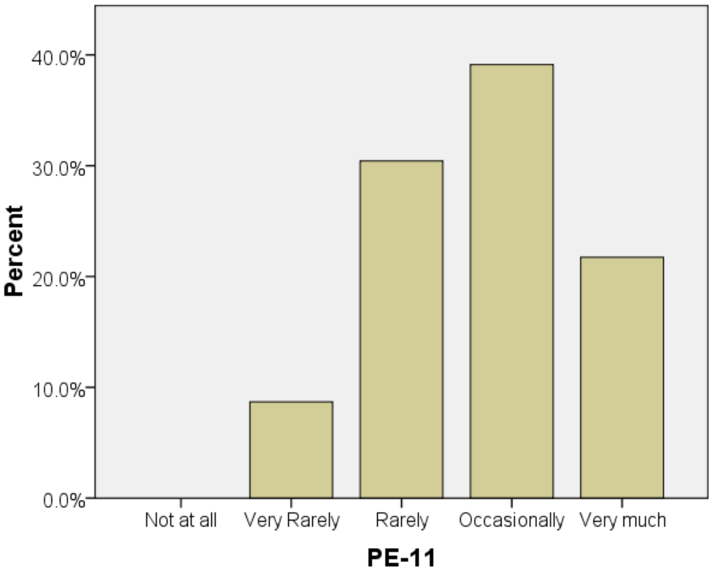


Figure 7-18: Responses on how often participants use wayfinding applications (PE-11)

Even though in the previous question (PE-10) most participants answered that they have average or better sense of direction, Figure 7-18 reveals that they also rely on wayfinding applications. All of them have used such applications at some point, either for real-time navigation or for simpler map-browsing applications, without advanced functionalities. 39.2% of the participants believe that they make average or rare use of wayfinding applications, whereas 60.8% uses them more frequently.

7.6 Virtual and Augmented Reality for Wayfinding Results

This part of the report presents the results produced by the objective measurements of the evaluation. These variables have been either recorded by the system, while the participants were executing the wayfinding tasks, or have been derived by processing the recorded variables after the participants completed both tasks. During the experiment, every measurement was recorded on a single GPX document that stored information about each task accomplished by every participant (i.e. VR and AR navigation). The Questionnaire can be found in Appendix XIII whereas the collected data in Appendix XIV. The recorded variables are presented in the following list.

1. Task *Start Time*;
2. Task *End Time*;
3. *Distance Covered*.

The derived variables, which have been produced, after examining the recorded measurements are presented in the following list.

1. *Total Time*;
2. *Minimum Speed*;
3. *Average Speed*;
4. *Maximum Speed*;

The selected statistical analysis has focused on the values retrieved from both groups of measurements. Evidently, there are two variables, which do not directly influence any conclusions. These are the *Start Time* and *End Time*, which describe the initiation and termination time of each wayfinding task respectively. These measurements have been used to produce a new variable, which is the *Total Time* required by each participant to complete the task. Although just the measurement of the *Average Speed* would be considered satisfactory for verifying the performance of each user and interface blend, we needed to introduce *Minimum Speed* and *Maximum Speed*, in order to describe the level of assistance that each interface provided for each tasks. From *Minimum Speed* and *Maximum Speed* we can more objectively measure specific aspects such as; if the users required to stop at a decision point in order to acquire the new heading towards the next waypoint and with which interface they felt more confident so that they could move faster towards the next goal.

7.6.1 Statistical Techniques

The main objective of this part of the research was to assess the differences, in terms of navigation performance, of the users in both wayfinding scenarios. The two conditions, which are being investigated, consist of the use of the system with the VR interface and the AR interface, respectively. That is why it was found compulsory to apply a mix of statistical techniques that would compare the differences between the two groups. The primary aim of every technique was to provide an assessment on whether the difference between the groups is statistically significant, which means that the results have not occurred arbitrarily. Generally, there are two kinds of statistical tests, which can produce valuable results for this assessment. These are either *parametric* tests or *non-parametric* tests. Although there is an extensive literature on the comparison of these types (Field, 2005) (Pallant, 2007), we have to mention some of their differences, which affected the progress of the analysis. *Parametric* tests can be considered more accurate than *non-parametric* tests. The reason is that they make a number of assumptions about the population from which the sample is drawn and about the type of data, which is recorded. Conversely, *non-parametric* tests are not that strict in fulfilling the assumptions and have been found more suitable for smaller sample sizes or when the type of data is measured at ordinal or higher level (Pallant, 2007). Generally, four types of variables exist in statistics (Field, 2005). These are *Nominal*, *Ordinal*, *Interval* and *Ratio*, mentioned according to the volume of information that they can describe. *Nominal* and *Ordinal* types are *non-parametric* data, whereas *Interval* and *Ratio* can be used in *parametric* statistical tests (Changing Minds, 2009). The variable types can also be separated into three categories, which are *categorical*, *ordinal* and *continuous* data (Pallant, 2007). Apart from the measured data types, *parametric* tests must also conform to four general rules (i.e. assumptions) and additional specific rules according to the requirements of each individual test. The four assumptions of *parametric* tests are listed below.

1. Normally distributed data
2. Homogeneity of variance
3. Level of measurement
4. Independence of observations

The validation of these assumptions makes the results of *parametric* techniques more accurate than those of *non-parametric* techniques (Pallant, 2007). The concern of the first assumption is to validate that the sample was drawn from a *Normally Distributed* population (Field, 2005) (Pallant, 2007). There are two ways in which this assumption could be verified. The first way is to visually inspect the histograms produced by the data, in order to verify that the distribution scores do not deviate from normality. This can be accomplished by evaluating the *skewness* and *kurtosis* of the distribution scores. The second way that can be used to assess that the distribution is normal, is by using a specialised statistic test. This method yields more objective results because it compares the scores of our sample with the scores produced by a normal sample that had the same mean and standard deviation. For every single measurement examined by this research task, the *Kolmogorov-Smirnov* and *Shapiro-Wilk* (Shapiro and Wilk, 1965) tests have been conducted, in order to verify whether the distribution of our sample was considered normal. Field argues that the *Shapiro-Wilk* test is more accurate in exploring normality (Field, 2005). That is the reason why we are going to report only the values produced by this test when we verify the normality assumption. The tests were accomplished by using *SPSS v17.0 for Windows* (IBM, 2010) and the full set of data has been attached to Appendix XIV. If the result of the test is less than *0.05* it means that it is significant, so not normal, and subsequently it shows that the distribution is not normal either. The second assumption of *parametric* tests is concerned with the *Homogeneity of Variance* between the groups that are examined. In our case, the same people formed both the VR and AR groups. A famous statistical test that evaluates if the difference between the variances of the groups is equal is *Levene's* test (Levene, 1960). The null hypothesis is not valid when there is a significant difference between the variances. This happens when the result of *Levene's* test is significant (i.e. $p<0.05$). In such cases the homogeneity of variance assumption is violated and a *parametric* test is not found suitable for execution. The *Level of Measurement* assumption instructs that the type, which the data is measured against, is at least of interval level. Every objective measurement collected in this evaluation conforms to this assumption of parametric tests. The last assumption is concerned with the *Independence of Observations* between the subjects. The design of the evaluation ensures that even though we had been planning to execute repeated measures tests, there was no influence on the behaviour of the participants or at least every possible effort was made not to, as explained in the previous sections.

One criterion that influenced the selection of the techniques for accomplishing the statistical analysis is the conformance to the aforementioned assumptions. Because the objective was to compare the performance of groups formed by the same participants, who had been evaluated on two conditions, it was found that a *Paired-Sample T-Test*

(Fisher, 1925) is appropriate, when we can use a *parametric* option. If all assumptions of *parametric* tests are validated then it is safe to execute this test. The *Paired-Sample T-Test* can show if there is a statistically significant difference in the mean scores produced by the participants on both occasions; while accomplishing the wayfinding task with the VR interface and with the AR interface, respectively. In a *Paired-Sample T-Test*, the statistic result t can be found by dividing the mean of differences by the standard error of differences. If the probability value p is less than 0.05 then we consider that the result is statistically meaningful. Finally, the effect size r measures the magnitude of the changes in both conditions. Cohen has suggested a categorisation of the effect size according to the value of r (Cohen, 1988) (Cohen, 1992). In contrast, when the assumptions of *parametric* tests are not validated, the best alternative is to use the *Wilcoxon Signed-Rank Test* (Wilcoxon, 1945) (Siegel, 1956). This test has the same input and works similarly with the *Paired-Sample T-Test* apart from that it converts scores to ranks and compares them for both occasions. Similarly to the *Paired-Sample T-Test*, we are going to present the results of individual measurements in terms of t, p and r values, for the *Wilcoxon Signed-Rank Test* as well. Although it would be interesting to examine the relation of the output produced by the participants according to their gender and age, we decided not to accomplish such correlations. It was due to the fact that the objectives of this evaluation did not instruct the fulfilment of such assessments and because the decision to conduct such analysis would require a different and more time-consuming process in order to get the approval from the governing body. In that case the governing body would be the *City University Research Ethics Committee* and not the *School of Informatics Research Ethics Committee*. The *University Committee* meets three times per academic year and the planned date for their next meeting would postpone the evaluation process for not less than three months.

In the very extensive literature that exists about statistics, authors (Pallant, 2007) propose several options in order to select the statistical technique according to the validation of the assumptions. In this research task, if all assumptions have been validated, then a *parametric* test has been applied on the data. In cases where not every assumption has been validated and if the transformation of the data could validate the assumptions thus rendering a *parametric* test suitable, then this kind of test will be selected. Finally, if there still are invalid assumptions, which cannot be justified, then a *non-parametric* test will be applied. In each case, when the results of a test are presented, we have tried to provide the most descriptive statistics, which are found applicable to the test and justify our intentions.

7.6.2 Statistical Results

At the beginning of Chapter 7.6, there is a description of the measureable, quantitative data that has been either recorded or derived. In order to perform the planned post-hoc tests, we needed to verify if the data validated the parametric tests' assumptions. The assumption about the level of measurement is validated, because the data is of integral type for every measured variable used in this part of the evaluation. The assumption about the independence of observations is also tenable because although the scores in the experimental conditions are not independent for a given participant, the behaviour between different participants is independent (Field, 2005). The third assumption, which must be explored, needs to verify that the data is normally distributed. To verify that the assumption of normality is tenable, we need to evaluate the histograms describing the data and to examine the significance of the results produced by the *Shapiro-Wilk* (and *Kolmogorov-Smirnov*) test. The following table illustrates the results of the normality tests, which every objective variable was subjected to.

Tests of Normality

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	Df	Sig.
VR Distance Covered	.326	20	.000	.414	20	.000
VR Min Speed	.294	20	.000	.714	20	.000
VR Avg Speed	.137	20	.200*	.930	20	.157
VR Max Speed	.453	20	.000	.298	20	.000
VR Total Time (sec)	.224	20	.010	.669	20	.000
AR Distance Covered	.270	20	.000	.595	20	.000
AR Min Speed	.459	20	.000	.588	20	.000
AR Avg Speed	.189	20	.058	.957	20	.491
AR Max Speed	.384	20	.000	.541	20	.000
AR Total Time (sec)	.236	20	.005	.775	20	.000

a. Lilliefors Significance Correction

*. This is a lower bound of the true significance.

Table 7-1: Normality Tests for Distance Covered, Min, Avg, Max Speed & Total Time Required

Furthermore, the normality results are presented in the following figures. The figures show the histograms, normal Q-Q plots, detrended normal Q-Q plots and the box plots, which describe the data. The order in which the figures are presented is based on the measured variables, to make the visual comparison between the different scores, according to each interface easier.

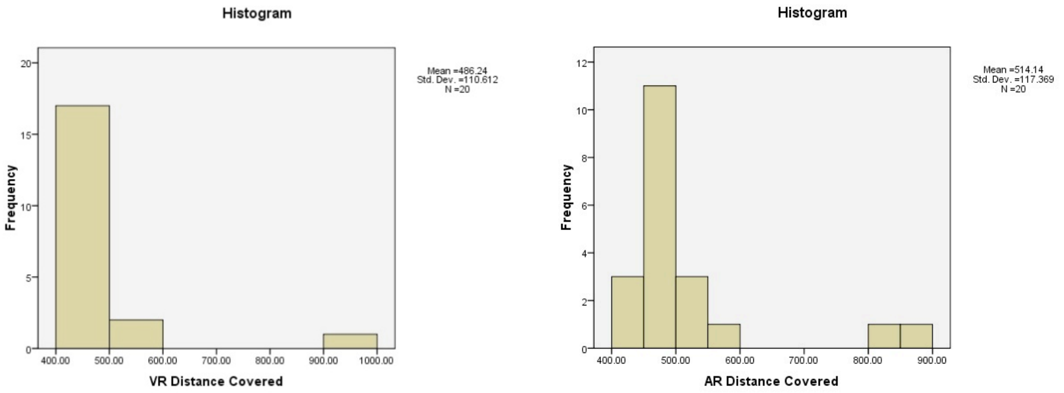


Figure 7-19: Histograms for the Distance Covered with the a) VR interface and b) AR interface

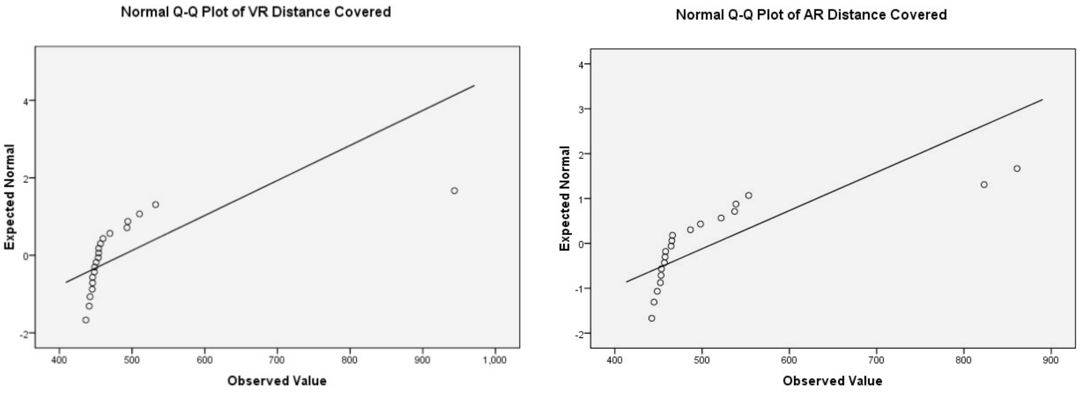


Figure 7-20: Normal Q-Q Plots for the Distance Covered with the a) VR interface and b) AR interface

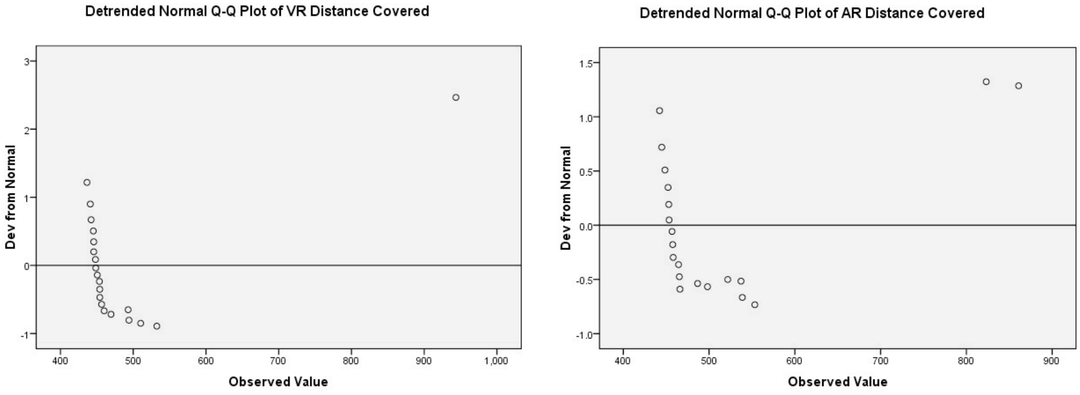


Figure 7-21: Detrended Q-Q Plots for the Distance Covered with the a) VR interface and b) AR interface

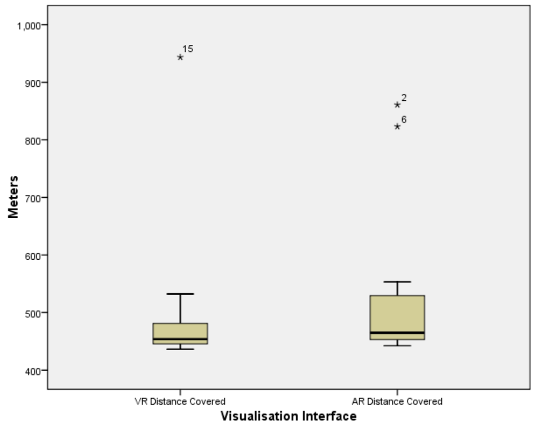


Figure 7-22: Box plot for the Distance Covered with each interface

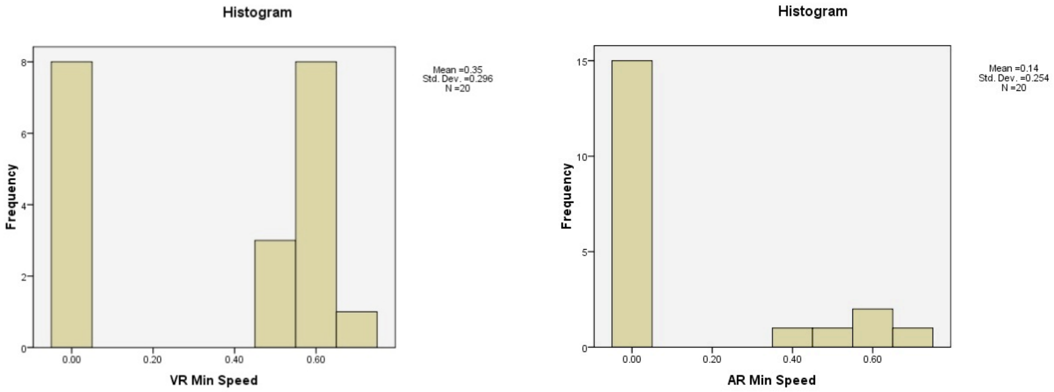


Figure 7-23: Histograms for the Minimum Speed with the a) VR interface and b) AR interface

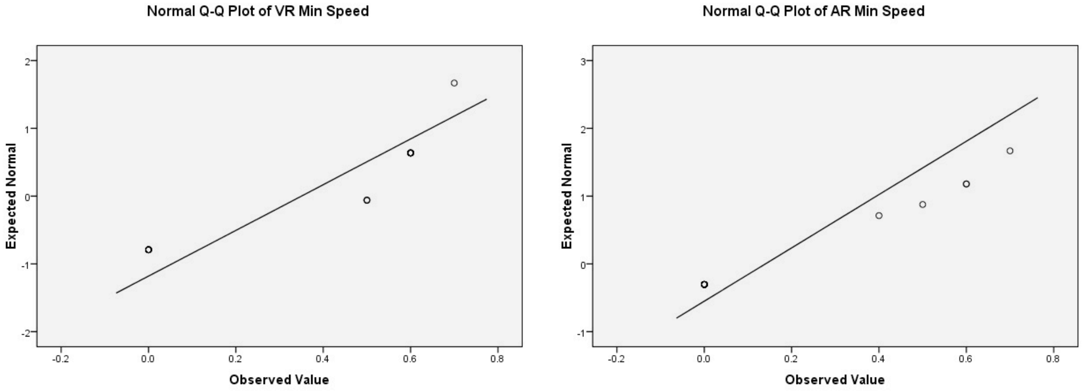


Figure 7-24: Normal Q-Q Plots for the Minimum Speed with the a) VR and b) AR interface

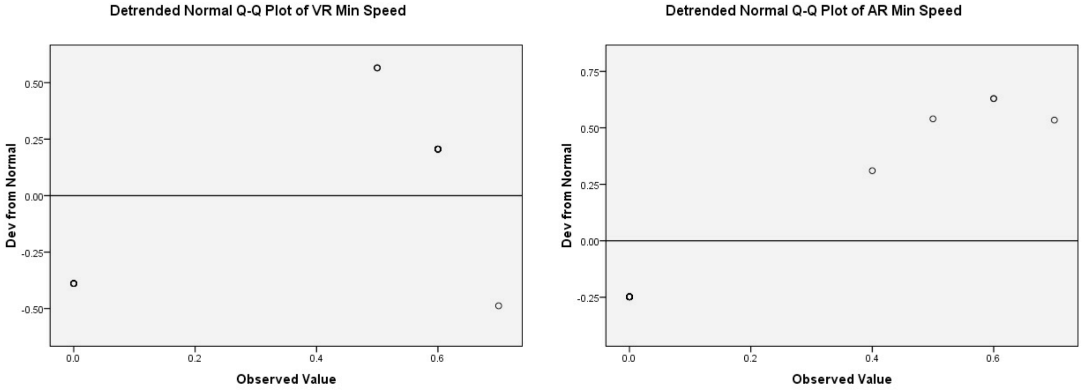


Figure 7-25: Detrended Q-Q Plots for the Minimum Speed with the a) VR interface and b) AR interface

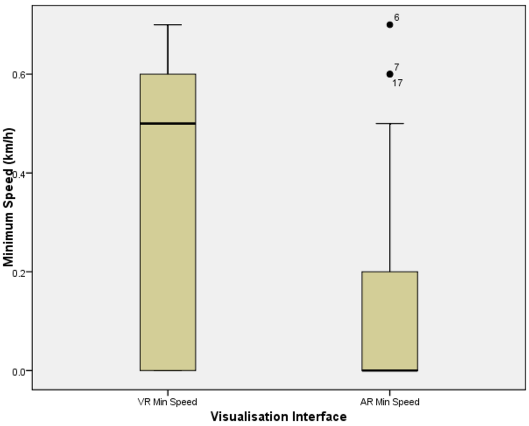


Figure 7-26: Box plot for the Minimum Speed with each interface

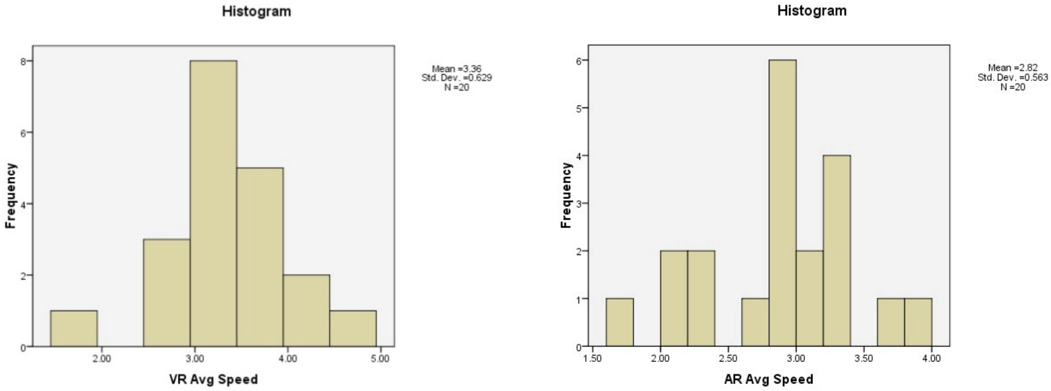


Figure 7-27: Histograms for the Average Speed with the a) VR interface and b) AR interface

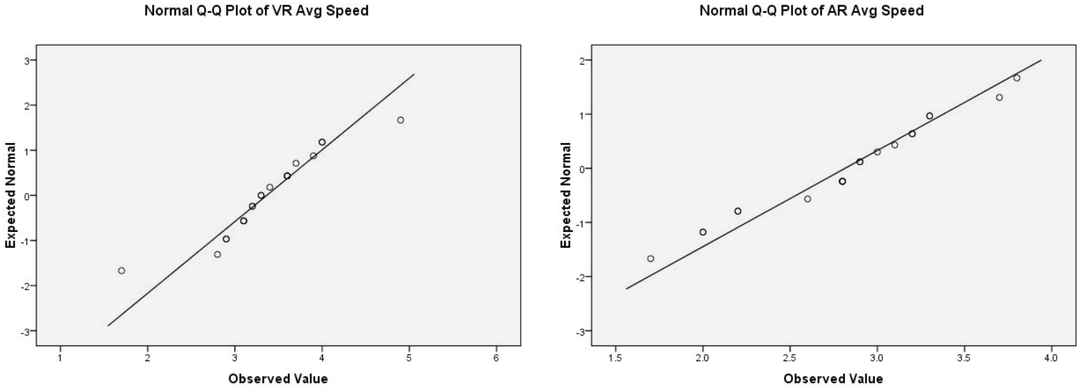


Figure 7-28: Normal Q-Q Plots for the Average Speed with the a) VR interface and b) AR interface

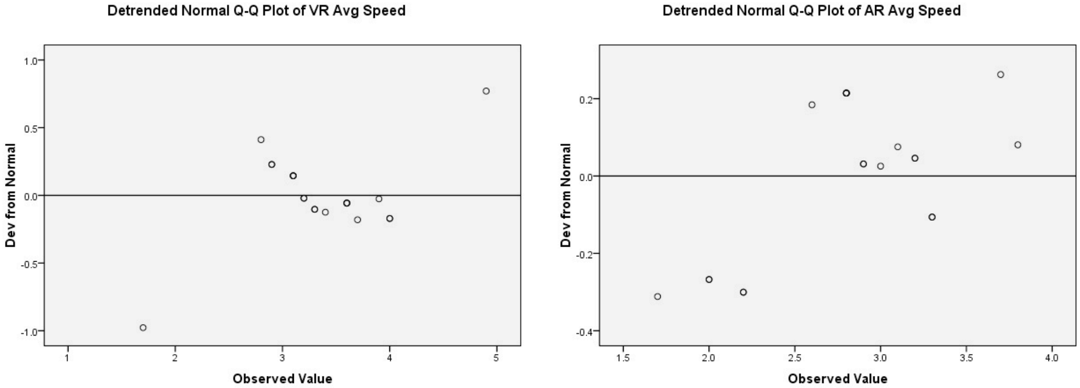


Figure 7-29: Detrended Q-Q Plots for the Average Speed with the a) VR interface and b) AR interface

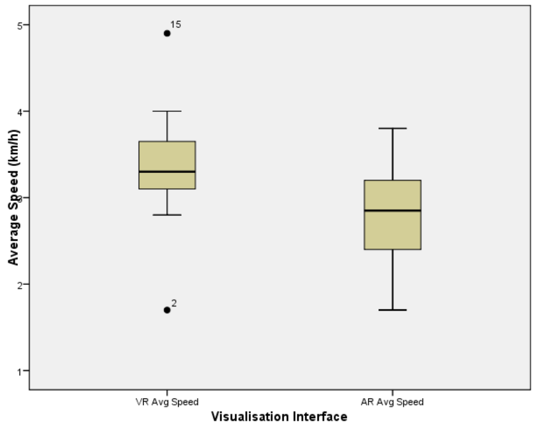


Figure 7-30: Box plot for the Average Speed with each interface

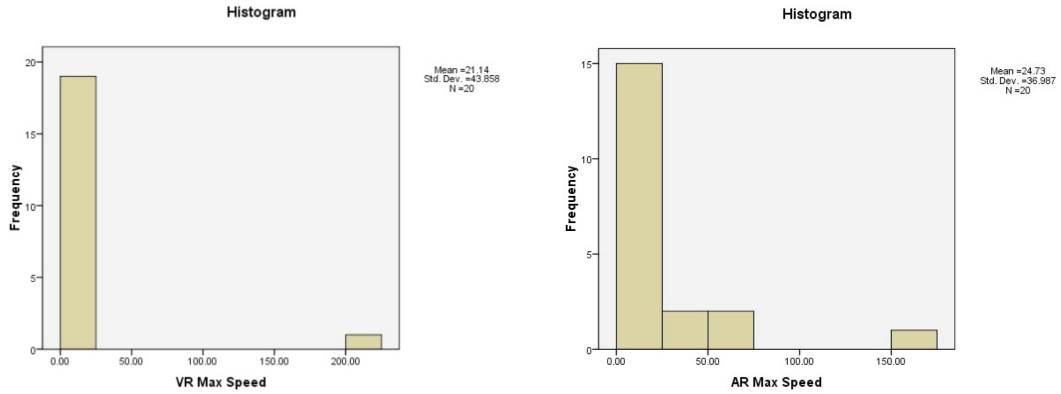


Figure 7-31: Histograms for the Maximum Speed with the a) VR interface and b) AR interface

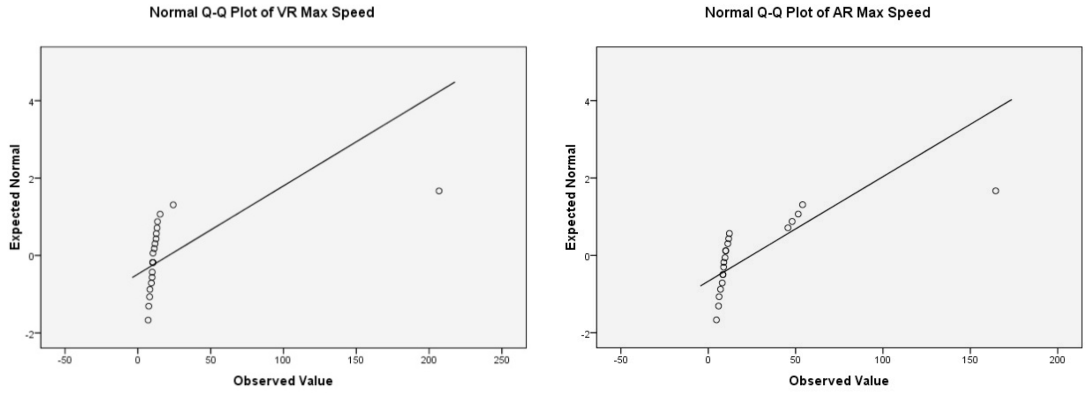


Figure 7-32: Normal Q-Q Plots for the Maximum Speed with the a) VR and b) AR interface

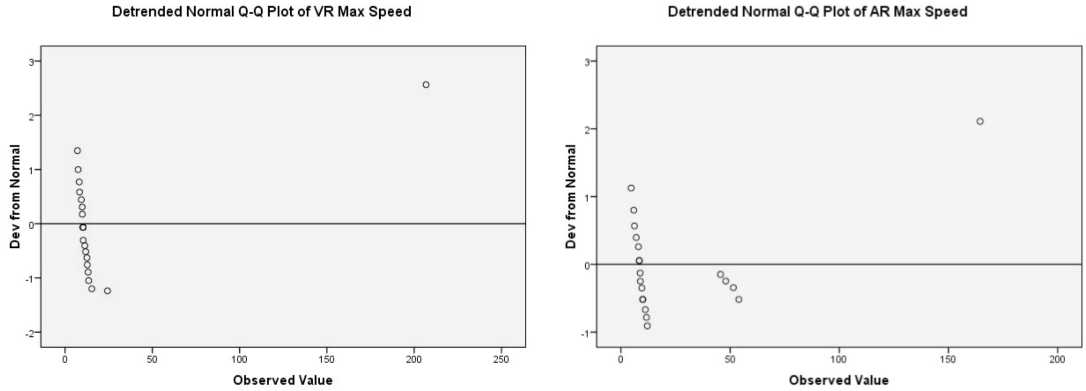


Figure 7-33: Detrended Q-Q Plots for the Maximum Speed with the a) VR interface and b) AR interface

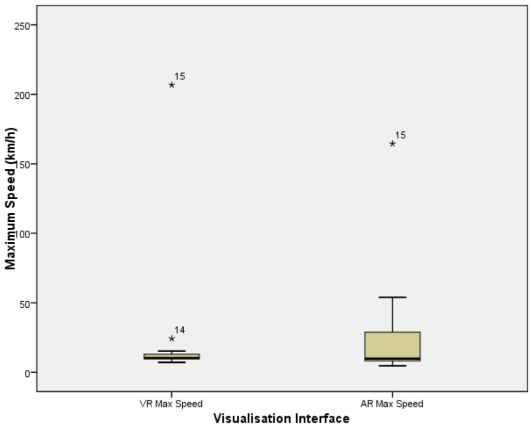


Figure 7-34: for the Maximum Speed with the a) VR interface and b) AR interface

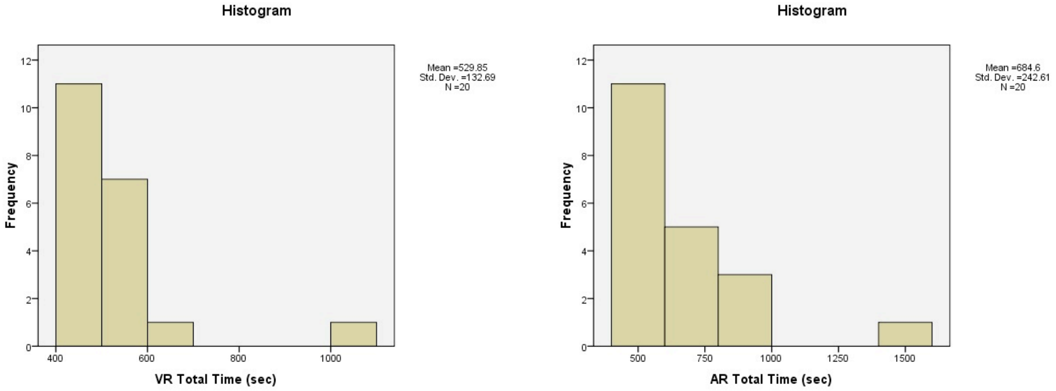


Figure 7-35: Histograms for the required Total Time with the a) VR interface and b) AR interface

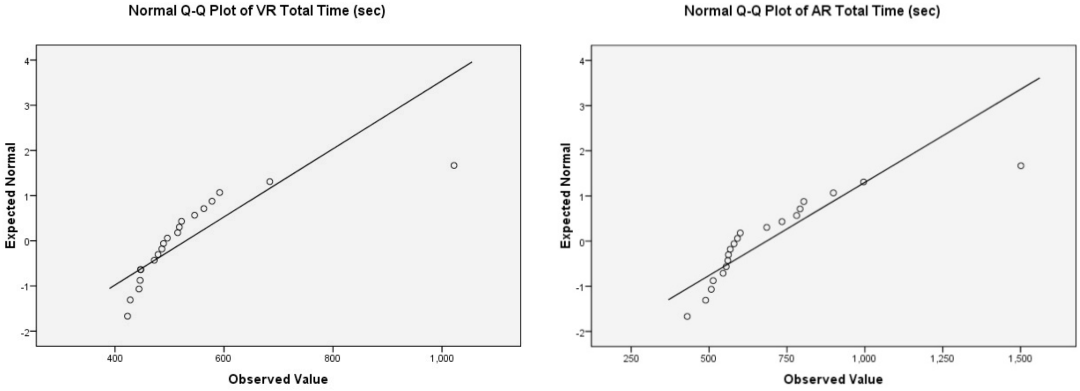


Figure 7-36: Normal Q-Q Plots for the required Total Time with the a) VR and b) AR interface

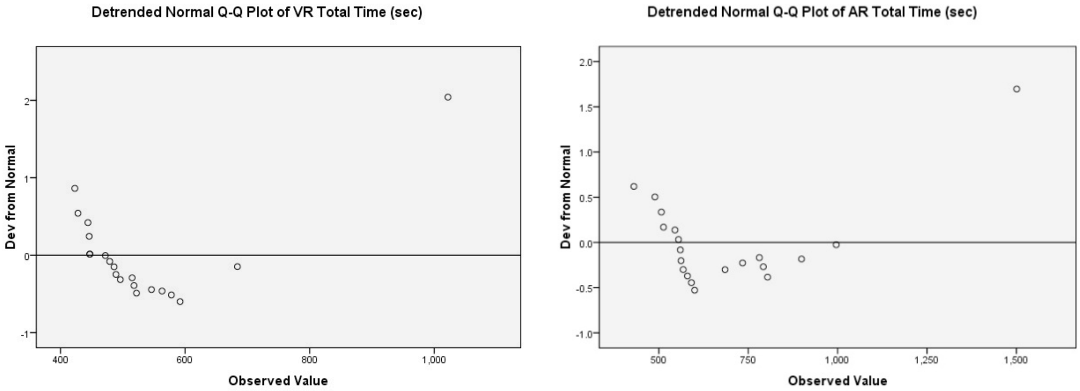


Figure 7-37: Detrended Q-Q Plots for the required Total Time with the a) VR interface and b) AR interface

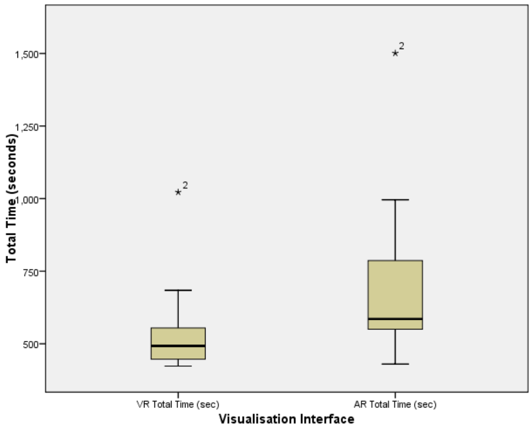


Figure 7-38: Box plot for the required Total Time with each interface

The histograms presented in Figure 7-19 show that the mean *Distance Covered* by 20 participants while using the VR interface was 486.244 metres and the standard deviation was 110.61208. In contrast, for the AR interface the mean *Distance Covered* was 514.1355 metres with standard deviation 117.36897. Following next, Figure 7-23 demonstrates that the mean *Minimum Speed* that the users reached was 0.35 kilometres per hour with standard deviation 0.29647 in VR, and 0.14 km/h, with standard deviation 0.25423 in AR. The first histogram in Figure 7-27 shows a mean *Average Speed* of 3.365 km/h with standard deviation 0.62935 for VR. The second histogram in the same figure shows that the mean *Average Speed* in AR was 2.8150 km/h with standard deviation 0.56314. For *Maximum Speed,* the mean for VR was 21.1350 km/h with standard deviation 43.85751, and for AR a mean of 24.7250 km/h with standard deviation 36.98674, as seen in Figure 7-31. The last variable that has been explored by the histograms in Figure 7-35 was the *Total Time* required by the participants to accomplish the wayfinding task. By using the VR interface, the mean time required was 529.85 seconds with standard deviation 132.690. In contrast, for AR the mean *Total Time* was 684.60 seconds with standard deviation 242.610.

The tests in Table 7-1 demonstrate that the deviation of our distribution is significant compared to a similar normal distribution, for most measurements. This can also be confirmed by the heavily skewed results of the histograms. The only variable, which was found normal, is the *Average Speed* maintained by the participants throughout each wayfinding task; in essence, by using both interfaces. The *Average Speed* in VR, $D(20)=0.93$, $p>0.05$ and in AR, $D(20)=0.957$, $p>0.05$ was in both cases normal. The rest of the cases were found significantly non-normal, for both interfaces. The results of the normality tests for the rest of the variables will not be described at this point, but can be analytically viewed in Table 7-1, as well as in the Appendices. A deviation from normality like that found in *Distance Covered*, *Minimum Speed*, *Maximum Speed* and *Total Time* shows that we cannot use a parametric technique because the assumption of normality is not tenable. For the *Average Speed,* though, the assumption of normality was found tenable.

In order to reduce the variations created by unknown factors (i.e. Unsystematic Variation) we had to counter balance the order in which a person participated in each wayfinding task. Making each person start the wayfinding task by using a different interface, reduced the risk of participants performing differently in one of the conditions due to familiarity with either the interface or the experimental task.

As we have seen in the previous paragraphs, there is one variable across both conditions, for which every assumption of parametric techniques has been validated. The variable is *Average Speed*. Therefore, a parametric test can be applied for verifying whether there is a statistical difference in the participants' mean *Average Speed* while accomplishing the wayfinding task, under both conditions (i.e. VR and AR). The most suitable parametric technique was found to be a *Paired-Sample T-Test*. But before accomplishing the T-Test to examine for statistical significance, the following error bars can provide an indication of the results for the examined variable.

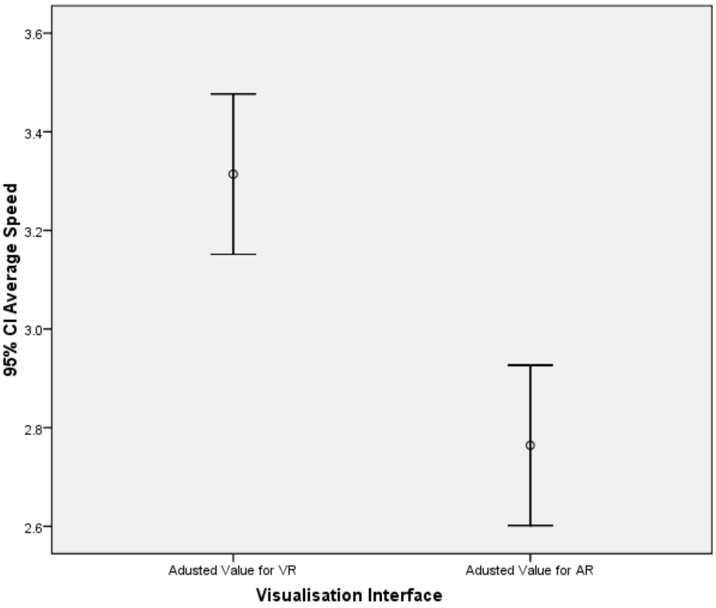


Figure 7-39: Error bars for Average Speed comparison

The production of the previous error bar involved a 4-step process (Field, 2005). Initially, the *Mean* for each participant in both conditions was calculated. Then the *Grand Mean* was produced, by averaging all the observations regardless of the group to which they belong (*Grand Mean = Mean / Number of participants)*. In the next step, the *Adjustment Factor* was found by subtracting the *Mean* from the *Grand Mean* for each individual, in both conditions. Finally, we had to create the *Adjusted Values* for VR and AR by adding the *Adjustment Factor* in the scores of each variable. The error bars indicate that there may be a significant difference between the two conditions because the bars do not overlap. Nevertheless, whether a significant difference between the two interfaces exists can be more accurately observed by the results of the *Dependent T-Test*.

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	VR Avg Speed	3.3650	20	.62935	.14073
	AR Avg Speed	2.8150	20	.56314	.12592

Table 7-2: Summary statistics for Average Speed in both interfaces

Table 7-2 presents some descriptive statistics about the *Average Speed* for both conditions. The only value, which has not been explored, is the standard error mean, which is calculated by dividing the standard deviation by the square root of the participant count.

Paired Samples Correlations

		N	Correlation	Sig.
Pair 1	VR Avg Speed & AR Avg Speed	20	.325	.162

Table 7-3: Pearson's Correlation and the two-tailed Significance for Average Speed

Table 7-3 presents the correlation, which explores the relationship strength between the two variables. So, the relationship between the participants' *Average Speed* when using the VR and the AR interface was investigated using Pearson product-moment correlation coefficient. There was a non-significant, medium, positive correlation between the two variables, $r=0.325$, $N=20$, $p=0.162$ (2-tailed), with high *Average Speed* in VR associated with high *Average Speed* in AR. The literature (Pallant, 2007) maintains that with a small sample (i.e. $N=20$) there can be moderate correlations that do not reach statistical significance at the conventional level ($p<0.05$).

Paired Samples Test

	Paired Differences							
				95% Confidence Interval of the Difference				
	Mean	Std. Deviation	Std. Error Mean	Lower	Upper	t	df	Sig. (2-tailed)
Pair 1 VR Avg Speed - AR Avg Speed	.55000	.69472	.15534	.22486	.87514	3.541	19	.002

Table 7-4: Paired-Samples T-Test output for Average Speed comparison

Table 7-4 illustrates the results of the repeated measures T-Test, which was conducted to compare the difference in *Average Speed* while using the two interfaces. The positive value of *t* shows that the mean *Average Speed* in VR is greater than the mean *Average Speed* in AR. Therefore, we can conclude that on average the participants completed the wayfinding task with significantly higher *Average Speed*, while using the VR interface, in comparison to the AR interface, $t(19)=3.541$, $p< 0.05$. The mean difference in *Average Speed* was 0.55 km/h with 95% confidence interval ranging from 0.22486 to 0.87514. The error bars found in Figure 7-39 verify this result. In conclusion, $r=0.63$ indicated a large effect size, with a substantial difference in the *Average Speed* between the VR and AR interfaces. The effect size was calculated by using the formula found in (Roseenthal, 1991), which was computed from *t* value of the T-Test measuring the differences between the two groups.

Until this point, this subchapter explored if the assumptions of parametric tests were tenable. From the 5 objective variables, which have been analysed, only one (i.e. *Average Speed*) validated all assumptions. For that reason, we performed the parametric *Dependent T-Test* on that variable and presented the results in the previous paragraph. For the other 4 variables, which did not validate the assumptions of parametric tests, an alternative solution was required, in order to compare the performance of the participants. The solution was found in the use of the *Wilcoxon Signed-Rank Test*. Similarly to the *Paired-Samples T-Test*, the *Wilcoxon Signed-Rank Test* is employed in order to compare two related conditions, but it does not make strict assumptions about the population from which the sample was drawn from (e.g. normally distributed data), as the former technique does. In the following paragraphs of this chapter, the reader can find the results of the statistical analysis accomplished by using the *Wilcoxon Signed-*

Rank applied on the *Distance Covered*, *Minimum Speed*, *Maximum Speed* and *Total Time* in both experimental conditions (i.e. VR and AR wayfinding).

Ranks

		N	Mean Rank	Sum of Ranks
AR Distance Covered - VR Distance Covered	Negative Ranks	8[a]	9.38	75.00
	Positive Ranks	12[b]	11.25	135.00
	Ties	0[c]		
	Total	20		
AR Min Speed - VR Min Speed	Negative Ranks	10[g]	5.55	55.50
	Positive Ranks	1[h]	10.50	10.50
	Ties	9[i]		
	Total	20		
AR Max Speed - VR Max Speed	Negative Ranks	13[m]	9.92	129.00
	Positive Ranks	7[n]	11.57	81.00
	Ties	0[o]		
	Total	20		
AR Total Time (sec) - VR Total Time (sec)	Negative Ranks	3[p]	4.33	13.00
	Positive Ranks	17[q]	11.59	197.00
	Ties	0[r]		
	Total	20		

a. AR Distance Covered < VR Distance Covered
b. AR Distance Covered > VR Distance Covered
c. AR Distance Covered = VR Distance Covered
g. AR Min Speed < VR Min Speed
h. AR Min Speed > VR Min Speed
i. AR Min Speed = VR Min Speed
jm. AR Max Speed < VR Max Speed
n. AR Max Speed > VR Max Speed
o. AR Max Speed = VR Max Speed
p. AR Total Time (sec) < VR Total Time (sec)
q. AR Total Time (sec) > VR Total Time (sec)
r. AR Total Time (sec) = VR Total Time (sec)

Table 7-5: Wilcoxon ranked scores for the paired comparison of the 4 objective variables

Test Statistics[c]

	AR Distance Covered - VR Distance Covered	AR Min Speed - VR Min Speed	AR Max Speed - VR Max Speed	AR Total Time (sec) - VR Total Time (sec)
Z	-1.120[a]	-2.015[b]	-.896[b]	-3.435[a]
Asymp. Sig. (2-tailed)	.263	.044	.370	.001

a. Based on negative ranks.

b. Based on positive ranks.

c. Wilcoxon Signed Ranks Test

Table 7-6: Wilcoxon test results for the paired comparison of the 4 objective variables

The results in the previous tables demonstrate that there was an insignificant change in the *Distance Covered* by the users, while using the VR interface, *Mdn*=453.81 metres, compared to the AR alternative, *Mdn*=464.915. The statistic results report T=75, Z=-1.12 and p>0.05, with a small effect size, r=-0.18 (Roseenthal, 1991). The second insignificant result that was observed is the difference of the *Maximum Speed* reached by the users. At this point we have to underline that in this test there was a single case (i.e. Participant 15), which produced extreme values in both tasks, as seen on the box plot in Figure 7-34. Therefore, it would be safe to exclude this case from the analysis. After running the *Shapiro-Wilk* and the *Wilcoxon* tests without that case, the assumption of normality was neither validated, nor was a significant result produced. So, the results that include case 15 will be presented because they take into consideration every valid pair of cases from our sample. The change in the *Maximum Speed* that the participants reached in the VR scenario, *Mdn*=10.35 km/h, was insignificantly different compared to the AR scenario, *Mdn*=9.8 km/h. The test results presented T=81, Z=-0.896 and p>0.05, with a small effect size, r=-0.14. Following next, the *Wilcoxon* test demonstrated that there was a significant difference in the *Minimum Speed* of the participants. The median for VR was 0.5 km/h, whereas for AR the median was 0 km/h. The results of the test demonstrate T=10.5, Z=-2.015 and p<0.05, with a small to medium effect size, r=-0.32. Furthermore, the *Total Time* required to complete each task was significantly lower while using the VR interface, *Mdn*=492.5 seconds, compared to the AR interface, *Mdn*=585.5. The results of the test showed T=13, Z=-3.435 and p<0.05, with a medium to large effect size, r=-0.54. A further discussion of the objective results is provided in Chapter 8.2 of this report.

7.7 Aura Usability Results

In this part of the report the results of the subjective measurements of the evaluation are presented. The data has been collected with the help of a questionnaire. The questionnaire included 4 sections, which tried to collect information about the *usefulness*, *ease of use*, *ease of learning*, *satisfaction* and *system performance* of each interface. The design of the questionnaire was influenced by the literature and especially from other projects that conducted similar research, such as LOCUS. The Questionnaire is presented in Appendix XIII whereas the collected data can be found in Appendix XIV. In this report, several publications have been identified which conducted subjective end-user evaluation or proposed certain techniques. Very selectively and without intending to mention all relevant references, the following texts were found particularly beneficial. (Witmer and Singer, 1998) (Preece et al., 2002) (Dix et al., 2003) (Hix et al., 2004) (Swan and Gabbard, 2005) (Benford et al., 2005) (Pingel and Clarke, 2005) (Kjeldskov et al., 2005) (Bertini et al., 2006) (Dünser et al., 2008) (Slater et al., 2009a) The participants had to answer these questions after completing each wayfinding task. The *usefulness* section consisted of 13 questions, the *ease of use* had 9 questions and the *ease of learning* included 2 questions. Furthermore, in the *satisfaction* and *system performance* sections, there were 10 and 9 questions, respectively. The last part of the survey included 6 questions, in which the participants could provide their feedback on their overall experience with the system, as well as a subjective comparison of the user interfaces and the visualisation perspectives that they preferred. In total, the questionnaire was composed out of 49 questions, 43 divided in the aforementioned categories and the last 6, which needed to be answered only once, after accomplishing both tasks. The majority of questions conformed to a 5-point Likert scale (Likert, 1932) format. The literature on questionnaire design (Bell, 1999) suggested not inviting the participants to respond to ambiguous, double, presuming, leading, hypothetical, sensitive and negative questions because it could lead to errors. Our questionnaire did not follow this rule and 5 out of the 43 questions for each interface were negative. Although we intended to avoid this practice and after consulting both supervisors of the research about the questionnaire content, we took the informed decision to leave these questions in the current format. The reason was that every other question conformed to same scale, having the worse answers on the left and gradually progressing towards the best replies which were positioned on the right side of the page. Therefore, we tried to make the negative statement more evident by underlying the negative word. After the evaluation has completed the issue was further discussed and the author realised that this could be avoided by simply swapping the answers. The data type of the scale questions is considered as ordinal. For that reason, the main part of the analysis has been based on this kind of variables. The main part of the questionnaire also had 2 nominal variables and 2 open-ended questions. In addition, the last part of the questionnaire consisted of 3 nominal and 2 open-ended questions. The questionnaire and the analysis of every produced result can be found in the Appendices of this report.

7.7.1 Statistical Techniques For Both Tasks

The objective of this part of the evaluation has been to assess the usability of *Aura* as a system, but in the context of a wayfinding scenario. In the wayfinding task that the participants had to accomplish, some of Aura's functionalities were not exploited. Thus, a focused usability evaluation about the level of assistance its visualisation interfaces offered, and about selected functionalities, has been provided. Due to the fact that the system was evaluated in two conditions (i.e. VR & AR), it was found appropriate to compare the functionalities of the presented interfaces and consider the best features for the task which was under investigation. Thus, we are going to examine the difference of the subjective scores in every condition and report those that demonstrate a significance change. Consequently, we will retrieve information about the suitability of each implemented feature of the wayfinding application and whether it needs to be adjusted in future releases.

Because the nature of the data is ordinal or even nominal in two cases, one of the assumptions of parametric tests has already been violated. This means that parametric tests cannot be used for analysing the data. Thus, a non-parametric test, which will convert individual scores to ranks and compare them for both conditions, will be used. Similarly, with the analysis of the objective measurements, the same participants have been employed to test both conditions (i.e. repeated measures). The answers to most questions belong to the *Strongly Disagree - Strongly Agree* domain and they have been interpreted to the corresponding integer scale, *1 - 5* respectively. Due to the fact that parametric tests are not available, there is no need to perform individual tests in order to explore the tenability of the other parametric assumptions. Nevertheless, in the following part of the report a visual exploration of the data will be provided, as well as the results of the statistical techniques that have been applied.

The most suitable test satisfying our requirements is the *Wilcoxon Signed-Rank Test*, likewise to the objective measurements. It is a non-parametric test, equivalent to the parametric *Paired-Samples T-Test*, which makes fewer assumptions about the nature of the data and the distribution of the sample. Further information about this test has been provided in Chapter 7.6.1. Only the most significant results will be presented, mainly due to space limitations. The rest of them can be found in Appendix XV.

7.7.2 Statistical Results For Both Tasks (S)

Out of the 43-paired variables that have been measured, only 7 produced a significant difference between the two interfaces. In ascending order, these variables are: S1, S5, S10, S31, S32, S38 and S42. The letter *S* preceding each variable number stands for *Subjective* whereas *Q*, which we will come across later, stands for *Question*. Both terms are interchangeable. The following list shows which was the corresponding question that triggered the responses for each interface and the category which the question belongs to. The full set of questions can be found in Appendix XIII.

Usefulness

Q1. *This interface provided effective support for my wayfinding task.*

(Strongly Disagree - Strongly Agree)

Q5. *With this interface, I can find my way to an unknown place effectively.*

(Strongly Disagree - Strongly Agree)

Q10. *The user interface was helpful in informing me of my current task.*

(Strongly Disagree - Strongly Agree)

Satisfaction

Q31. *I would like to use this interface in contexts other than my profession.*

(Strongly Disagree - Strongly Agree)

Q32. *Rate the realism level of this interface.*

(Very Low – Very High)

System Performance

Q38. *The response of the interface to my manual input (touch screen) was prompt.*

(Strongly Disagree - Strongly Agree)

Q42. *The text style (colour and size) was easy to read.*

(Strongly Disagree - Strongly Agree)

The following figures present an exploratory summary of the results produced by the 23 participants. In each figure, a box plot is illustrated. The box plot presents the result obtained for both interfaces so that a preliminary visual comparison can be accomplished.

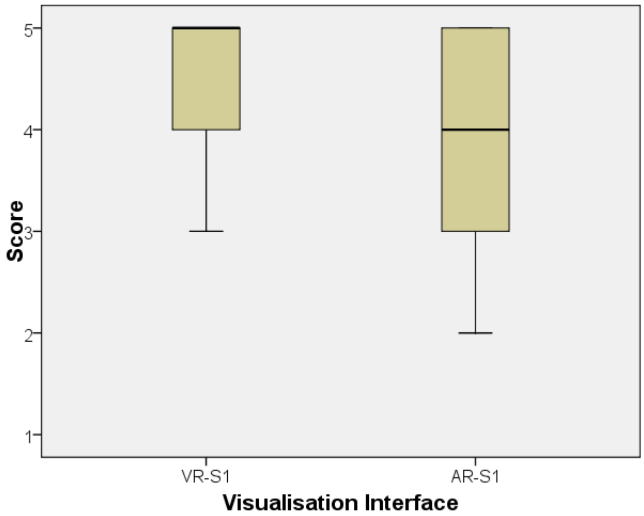


Figure 7-40: Box plot describing the answers provided for Q1, for each interface

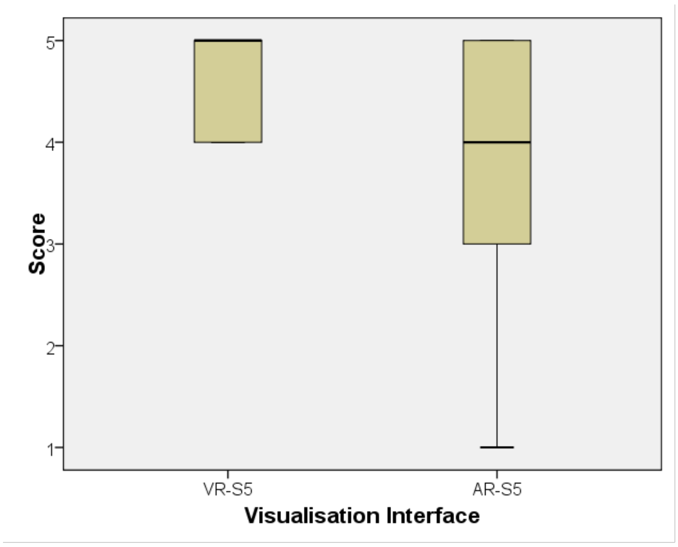


Figure 7-41: Box plot describing the answers provided for Q5, for each interface

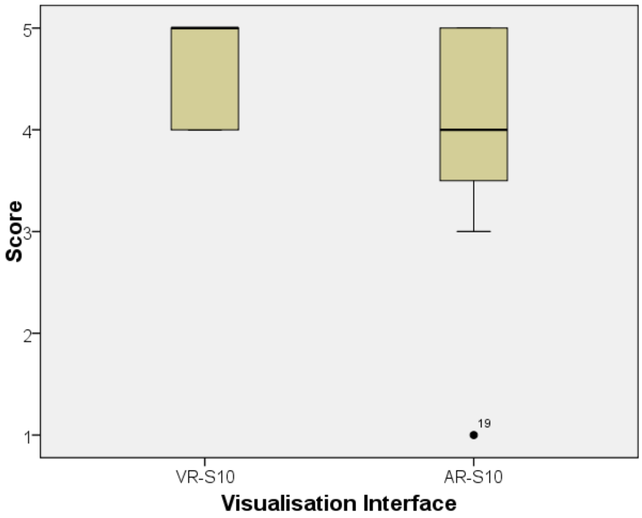


Figure 7-42: Box plot describing the answers provided for Q10, for each interface

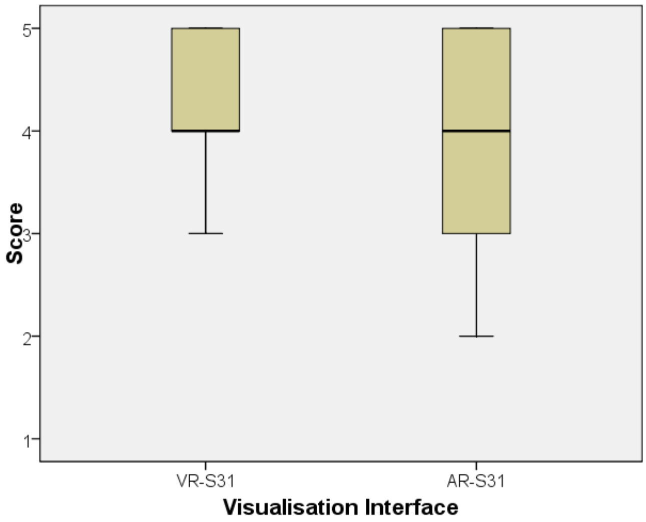


Figure 7-43: Box plot describing the answers provided for Q31, for each interface

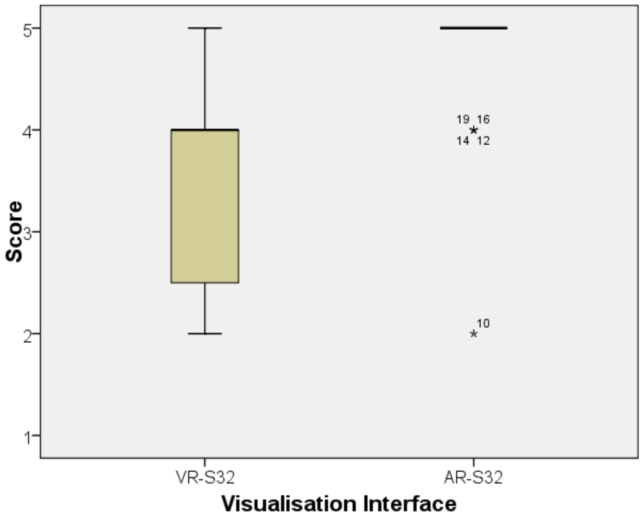


Figure 7-44: Box plot describing the answers provided for Q32, for each interface

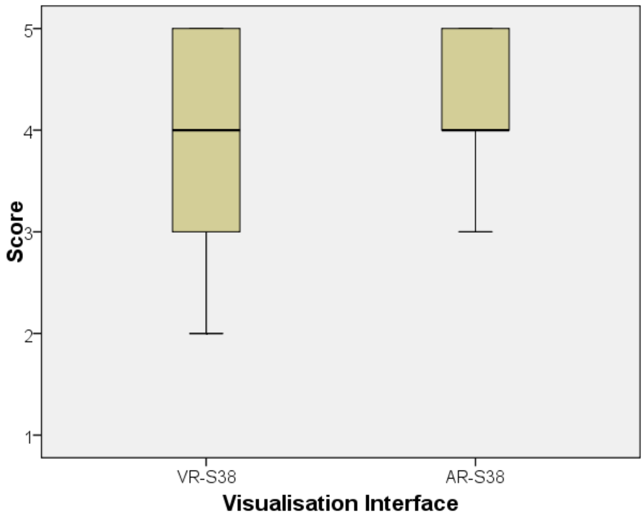


Figure 7-45: Box plot describing the answers provided for Q38, for each interface

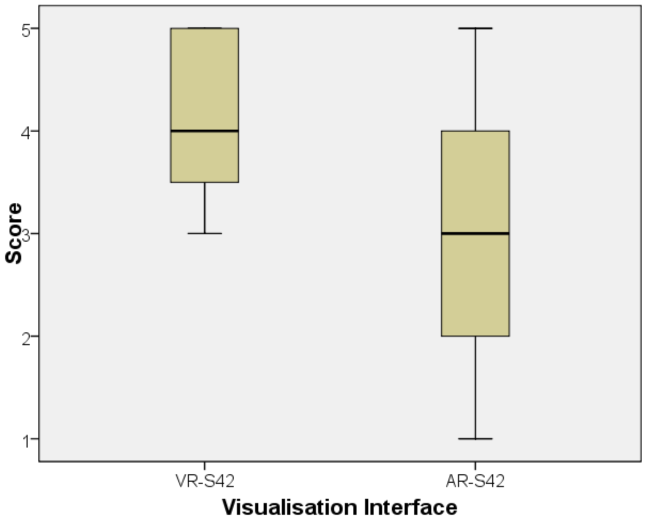


Figure 7-46: Box plot describing the answers provided for Q42, for each interface

The following tables present the statistics describing every group of answers. The most meaningful information is that 23 participants formed the sample, as well as the *Median*, *Minimum* and *Maximum* value for every case. These values are also illustrated in the previous box plots as the bold horizontal line and as the end of the two whiskers, respectively. Furthermore, Table 7-8 demonstrates the 25th and 75th percentiles, which are also represented on the corresponding box plots as the end of the length of each box.

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum
VR-S1	23	4.52	.593	3	5
VR-S5	23	4.57	.507	4	5
VR-S10	23	4.61	.499	4	5
VR-S31	23	4.30	.765	3	5
VR-S32	23	3.35	.982	2	5
VR-S38	23	3.87	1.014	2	5
VR-S42	23	4.09	.793	3	5
AR-S1	23	3.87	1.140	2	5
AR-S5	23	3.87	1.290	1	5
AR-S10	23	4.13	1.058	1	5
AR-S31	23	3.74	1.096	2	5
AR-S32	23	4.70	.703	2	5
AR-S38	23	4.43	.590	3	5
AR-S42	23	3.00	1.314	1	5

Table 7-7: Descriptive statistics (I) for the 7 subjective variables

Descriptive Statistics

	Percentiles		
	25th	50th (Median)	75th
VR-S1	4.00	5.00	5.00
VR-S5	4.00	5.00	5.00
VR-S10	4.00	5.00	5.00
VR-S31	4.00	4.00	5.00
VR-S32	2.00	4.00	4.00
VR-S38	3.00	4.00	5.00
VR-S42	3.00	4.00	5.00
AR-S1	3.00	4.00	5.00
AR-S5	3.00	4.00	5.00
AR-S10	3.00	4.00	5.00
AR-S31	3.00	4.00	5.00
AR-S32	5.00	5.00	5.00
AR-S38	4.00	4.00	5.00
AR-S42	2.00	3.00	4.00

Table 7-8: Descriptive statistics (II) for the 7 subjective variables

The following tables display the results produced after applying the *Wilcoxon Signed-Rank Test* on every pair of variables. The selected statistical technique was found more suitable compared to the alternatives, which are the *Sign Test*, the *McNemar Test* (McNemar, 1947) and the *Marginal Homogeneity Test* (Stuart, 1955) (Bhapkar, 1966) (Maxwell, 1970). Table 7-16 presents the level of statistical difference that exists between the compared interfaces, for each question, as well as the direction of change. Furthermore, the analysis of the results describes the effect size. The effect size was calculated by dividing the *Z*-score of each test by the square root of the number of participants, which is identical (i.e. *N*=23) for every condition (Roseenthal, 1991).

Ranks

		N	Mean Rank	Sum of Ranks
AR-S1 - VR-S1	Negative Ranks	10[a]	7.90	79.00
	Positive Ranks	3[b]	4.00	12.00
	Ties	10[c]		
	Total	23		

a. AR-S1 < VR-S1

b. AR-S1 > VR-S1

c. AR-S1 = VR-S1

Table 7-9: Wilcoxon ranked scores for the paired comparison of Q1 answers

Ranks

		N	Mean Rank	Sum of Ranks
AR-S5 - VR-S5	Negative Ranks	11[a]	8.45	93.00
	Positive Ranks	3[b]	4.00	12.00
	Ties	9[c]		
	Total	23		

a. AR-S5 < VR-S5

b. AR-S5 > VR-S5

c. AR-S5 = VR-S5

Table 7-10: Wilcoxon ranked scores for the paired comparison of Q5 answers

Ranks

		N	Mean Rank	Sum of Ranks
AR-S10 - VR-S10	Negative Ranks	9[a]	6.33	57.00
	Positive Ranks	2[b]	4.50	9.00
	Ties	12[c]		
	Total	23		

a. AR-S10 < VR-S10

b. AR-S10 > VR-S10

c. AR-S10 = VR-S10

Table 7-11: Wilcoxon ranked scores for the paired comparison of Q10 answers

Ranks

		N	Mean Rank	Sum of Ranks
AR-S31 - VR-S31	Negative Ranks	12[a]	7.75	93.00
	Positive Ranks	2[b]	6.00	12.00
	Ties	9[c]		
	Total	23		

a. AR-S31 < VR-S31

b. AR-S31 > VR-S31

c. AR-S31 = VR-S31

Table 7-12: Wilcoxon ranked scores for the paired comparison of Q31 answers

Ranks

		N	Mean Rank	Sum of Ranks
AR-S32 - VR-S32	Negative Ranks	1[a]	13.00	13.00
	Positive Ranks	19[b]	10.37	197.00
	Ties	3[c]		
	Total	23		

a. AR-S32 < VR-S32

b. AR-S32 > VR-S32

c. AR-S32 = VR-S32

Table 7-13: Wilcoxon ranked scores for the paired comparison of Q32 answers

Ranks

		N	Mean Rank	Sum of Ranks
AR-S38 - VR-S38	Negative Ranks	2[a]	3.50	7.00
	Positive Ranks	9[b]	6.56	59.00
	Ties	12[c]		
	Total	23		

a. AR-S38 < VR-S38

b. AR-S38 > VR-S38

c. AR-S38 = VR-S38

Table 7-14: Wilcoxon ranked scores for the paired comparison of Q38 answers

Ranks

		N	Mean Rank	Sum of Ranks
AR-S42 - VR-S42	Negative Ranks	14[a]	7.50	105.00
	Positive Ranks	0[b]	.00	.00
	Ties	9[c]		
	Total	23		

a. AR-S42 < VR-S42

b. AR-S42 > VR-S42

c. AR-S42 = VR-S42

Table 7-15: Wilcoxon ranked scores for the paired comparison of Q42 answers

Test Statistics[c]

	AR-S1 - VR-S1	AR-S5 - VR-S5	AR-S10 - VR-S10	AR-S31 - VR-S31	AR-S32 - VR-S32	AR-S38 - VR-S38	AR-S42 - VR-S42
Z	-2.391[a]	-2.601[a]	-2.230[a]	-2.696[a]	-3.498[b]	-2.365[b]	-3.370[a]
Asymp. Sig. (2-tailed)	.017	.009	.026	.007	.000	.018	.001

a. Based on positive ranks.

b. Based on negative ranks.

c. Wilcoxon Signed Ranks Test

Table 7-16: Wilcoxon test results for the paired comparison of 8 subjective variables

The results presented in the previous tables show that there was a significant difference between the levels of support that the users believed each interface provided. It was found that the wayfinding task was better supported in VR, $Mdn=5$ (i.e. *Strongly Agree*) rather than in AR, $Mdn=4$ (i.e. *Agree*). The results of the test demonstrate $T=12$, $Z=-2.391$ and $p<0.05$, with a large effect size, $r=-0.5$.

The following question was interested in whether the participants could find their way to an unknown place effectively by using each interface. Likewise, a significant change between the two groups of answers has been noticed, with higher scores for the simulated environment. For VR, *Mdn*=5, whereas for AR *Mdn*=4. The test produced $T=12$, $Z=-2.601$ and $p<0.05$, with a large effect size, $r=-0.54$.

Question 10 tried to examine whether each interface was helpful in informing the user about the task in progress. There was a significant difference in the answers provided for each task, with *Mdn*=5 for VR and *Mdn*=4 for AR. The produced results showed that $T=9$, $Z=-2.23$ and $p<0.05$, with a large effect size, $r=-0.46$. Therefore, the VR interface was found to be more helpful than the AR alternative, in informing the user about the current task.

The results of the following two questions compare user satisfaction between the two interfaces. In more detail, there was a significant difference about whether the participants would use each interface in any context other than their profession. Although the answers for both interfaces had the same median, *Mdn*=4, the test results demonstrated $T=12$, $Z=-2.696$ and $p<0.05$, with a large effect size, $r=-0.56$. Consequently, the users are more interested to see the VR environment operating in applications not related to their profession compared to the AR.

Question 32 examined the realism offered by each interface. Likewise, there was a significant difference between the responses provided by the participants. The median for VR and AR was *Mdn*=4 (i.e. *High*) and *Mdn*=5 (i.e. *Very High*), respectively. The result of the *Wilcoxon Signed-Rank Test* was $T=13$, $Z=-3.498$ and $p=0.0$, with a large effect size, $r=-0.72$. In contrast to the former tests, in this question the scores of AR were higher than those of VR. This was not unexpected because the AR interface provided a photorealistic feedback to the user, whereas VR offered a simulated environment with average quality textures.

The last two questions compared issues regarding the performance of the system. The test applied on Question 38 examined if there was a significant difference between the promptness of each interface, to the manual (i.e. touch screen) input of the user. The median for both groups of answers was *Mdn*=4. The reported result of the test was $T=7$, *Z*=-2.365 and *p*<0.05, with a large effect size, *r*=-0.49. Similarly to the previous question, users found that the response of the AR interface was prompter than the response of the VR, to their manual input.

The final question examined in this part of the report assessed whether the text style (i.e. colour and size), which was applied on each interface, was easy to read. For the VR interface, the median was *Mdn*=4 and for AR it was *Mdn*=3. Thus, the readability of the text presented in VR was significantly higher compared to the AR interface. The test results presented *T*=0, *Z*=-3.37 and *p*<0.05, with a large effect size, *r*=-0.7.

7.7.3 Results After Both Tasks (ABT)

This section presents 4 out of 6 measurements from the last part of the questionnaire. This part was designed to collect the answers, which were provided after the participants interacted with both interfaces of the application. All measurements are subjective and can be found in the Appendices. This section presents only 4 categorical results, as the other 2 are purely qualitative. The qualitative results are explored in Chapter 8.

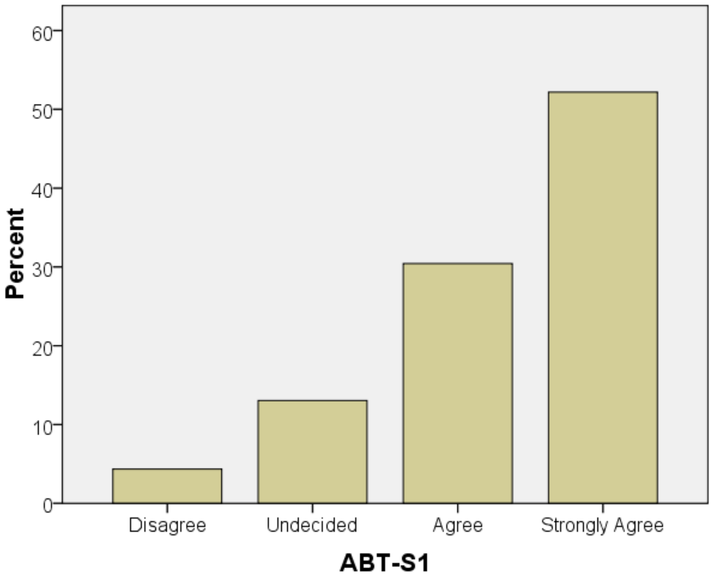


Figure 7-47: Bar chart presenting the answers provided in ABT-S1

The bar chart on Figure 7-47 illustrates that from the 23 users that answered the question, 82.6% were more motivated than usual to accomplish the wayfinding task because they were using a Smartphone. The rest were either averagely motivated (17.4%), or very little motivated (4.3%).

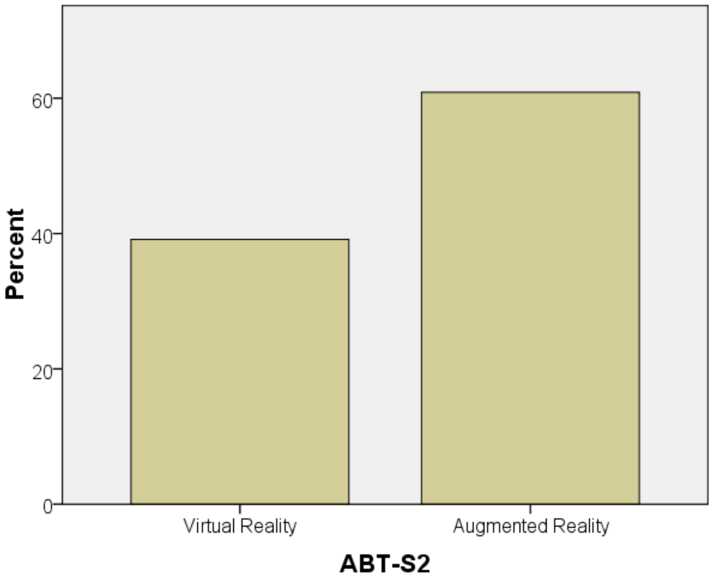


Figure 7-48: Bar chart presenting the answers provided in ABT-S2

In question ABT-S2, the participants had to select which egocentric, visualisation perspective they overall preferred. The available options are two; the one found in the VR interface or the one in AR. The difference between the two answers is 21.8%. The interface that got the most answers is AR (60.9%), mainly because of the photorealistic representation of the environment that it provides.

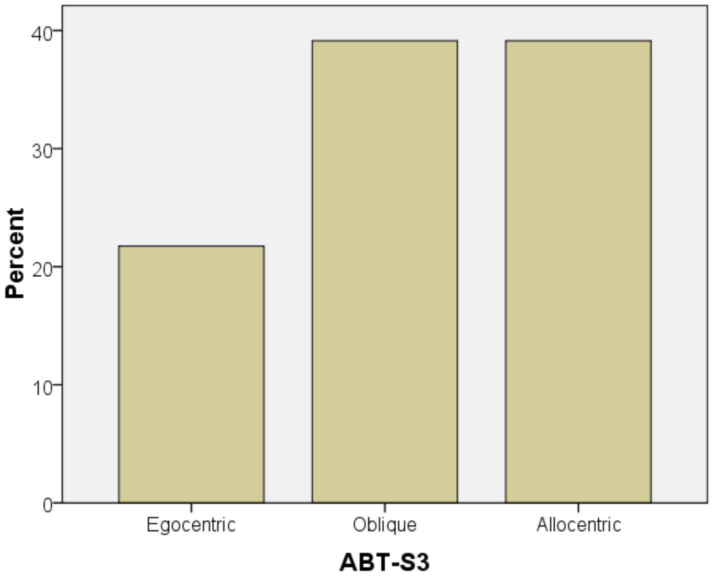


Figure 7-49: Bar chart presenting the answers provided in ABT-S3

The bar chart in Figure 7-49 shows the three available answers and the scores that they received. This question examined which visualisation perspective was the user's favourite. The egocentric perspective has been implemented on both user interfaces, whereas the allocentric oblique and the allocentric plan views have been implemented only in VR. The latter two perspectives acquired the same number of responses (39.1%). In contrast, the egocentric perspective got fewer responses (21.8%), which shows that it is not the primary user choice.

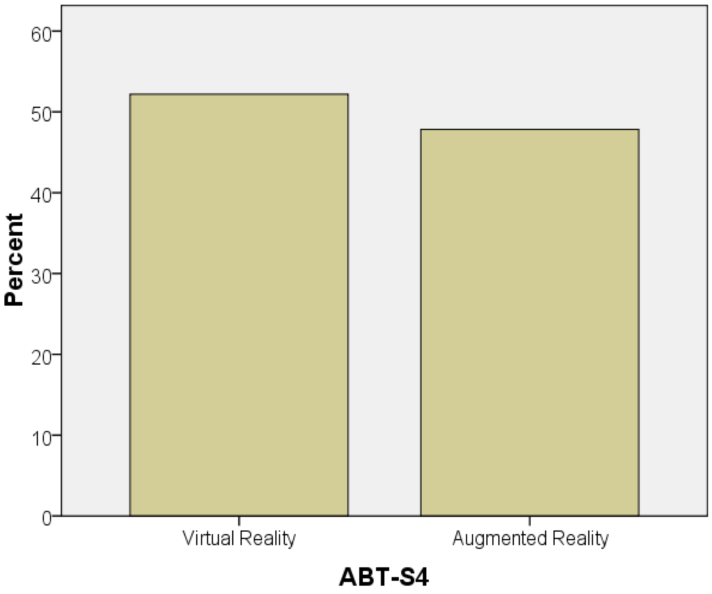


Figure 7-50: Bar chart presenting the answers provided in ABT-S4

From the results of ABT-S3, we could assume that the preferred interface for accomplishing a wayfinding task would be VR, because the egocentric perspective did not receive a high score. The results of question ABT-S4 confirm this assumption, but with a lot less difference than expected. The difference between both groups of answers is only 4.4%, which is translated to a single reply from the total number of 23 participants. A further discussion of the subjective results is provided in Chapter 8.2.

8 Discussion of Evaluation Results

This chapter provides a discussion of the topics, which have been explored during this research project. The chapter commences with a presentation of the results obtained by the preliminary evaluation cycles. Following next, the reader will find a discussion of the extensive assessment's results. The discussion of the results that have been produced by both evaluation processes is presented in the context of the framework's usability and functionality features.

Due to the nature of the methodology that was selected to develop the software product of the research, several limited evaluations of the system have taken place. Most of them have been accomplished by examining the behaviour and interactions of some expert users. Despite that, at certain points the researcher needed to establish solid results about the progress of the development and about meeting the requirements, which were initially identified. Consequently, two major evaluation iterations have taken place. These assessments and their most significant results have been analytically presented in Chapter 7 of this report – Chapter 7.2 and 7.3 for the Preliminary Evaluation and Chapter 7.4 to 7.7 for the Extensive Evaluation. The following sections summarise and discuss the results of both major evaluation tasks.

One of the primary objectives of this research has been to establish a technological framework capable of acquiring real-time information relevant to the user and presenting it in a way that can assist him or her to make better decisions regarding the task that they want to accomplish. Due to the fact that the user is mobile, the presentation of the information should include elements of the immediate environment and highlight certain entities that may be the source of the required context. This way, the user will be able to interact with these entities in the digital world. Furthermore, if these entities are significant for accomplishing a task, the user may select to interact with them in the real world as well. In order to visualise and interact with the immediate environment and its unique features efficiently, the use of an advanced user interface is required. This UI should also be capable of supporting several user activities and complement the user experience. Thus, the selected UI must simulate the natural behaviour of the user. For this project, we implemented and examined two distinct UIs that can depict the environment, as well as simulate user gestures and interactions. These interfaces are VR and AR. Both interfaces belong to the virtuality continuum (Milgram and Kishino, 1994) and are treated according to the geo-referenced virtual environment specifications (MacEachren et al., 1999).

8.1 Preliminary Evaluation Results

The main objective of the Preliminary Evaluation explored in Chapters 7.2 and 7.3 was to determine if the applied UI (i.e. VR, as AR was not fully developed at that point) could satisfy the requirements of the geo-referenced virtual environments and, subsequently, if it could become beneficial for a user. Thus, certain aspects have been examined, which present the accuracy and usefulness of the developed system in certain scenarios that took place in a simulated world. Investigating these factors should have been accomplished under the guidance of certain visualisation and interaction rules. A fundamental scenario, which has been very relevant to our needs, is wayfinding. That is because it can take place both in the real and in the virtual world. Wayfinding can be the core of any further potential scenarios because the actions that take place in its boundaries can be found in many applications that process real-time spatiotemporal context in simulated environments. As a result, the three key points, which have been examined in the context of mobile wayfinding are: (i) registration in the environment, (i) user movement and (iii) activity at decision points. Furthermore, the performance of the underlying mobile platform has been observed because at the time that the Preliminary Evaluation took place, mobile devices had only restricted functionalities and a reduced set of features available. This section highlights the results generated by the expert participants of the subjective Preliminary Evaluation.

8.1.1 Mobile Device Performance

Because there are several variables, which must be processed in real-time, the design of the system architecture should also be supported by the underlying hardware. The main functionality of the system includes the acquisition of context from various sources, both internal and external, and their presentation on the mobile display’s real estate. To accomplish this process:

- The framework should query the local sensors, the storage component and any remote devices for the latest updates in real time;
- The visualisation interface should be apparent and all virtual resources should be loaded and continuously updated;
- The system should process and record new information very frequently;
- The input from every source of context should be synchronised and presented on the device as soon as it occurs;
- The system should process several tasks in a seemingly concurrent manner.

Although certain software techniques can be utilised to accomplish that in order to achieve the best results:

- The mobile device should embed a fast CPU capable of multitasking.

Apart from the specifications of the utilised mobile device, the specifications of the sensor equipment should also be taken into consideration.

- The available connectivity options, the data communication protocols and their update frequency are the sensor parameters that could affect the operation of the system.
- An efficient system design should manage to minimise potential errors produced by the attached sensors;
- The environmental conditions could severely affect the accuracy of the sensors.

Another essential hardware component that Aura makes heavy use of is the embedded GPU. Due to the high intensity of graphics, both in VR and AR, it was found that:

- A dedicated graphics processor could prove beneficial for the system functionality;
- Having embedded hardware graphics acceleration, either in the form of a dedicated GPU or as CPU instruction sets, the stability and the presentation of the visualisations was improved, compared to the results produced by software-rendering techniques.

8.1.2 Registering Initial Location and Orientation

The second domain, which has been examined in the Preliminary Evaluation, was matching the user registration in both environments, the virtual and the real.

- The framework should support accurate registration of the user position and orientation in both environments.

Successful registration in an environment requires prior tracking of the use's latest context.

- The framework should offer features and functionalities that will assist a user to recognise the currently occupied location.

These features must support the matching of the user's cognitive frames of reference with the real world features. As a result, the user will become aware of the current situation and the potential interaction elements that may exist in the immediate environment. In order to successfully register the user in the environment, certain context variables must be acquired and certain environmental characteristics must be visualised. When the Preliminary Evaluation took place, the only context variable, which was being processed by the prototype, was location in 3D space. At that time, the orientation of the user was being calculated by processing changes generated by the user's location.

- User's orientation derived from position information is sufficiently accurate for the registration process when the user is in motion;
- User's orientation derived from position information has not proven beneficial for absolute registration purposes.

The reason is that the user cannot match the current pose when he or she uses the application for the first time or when standing still. This was a great disadvantage that required the introduction of a unique context variable in order to overcome it.

- Real-time orientation information can improve the user's registration process when a dedicated sensor module generates the information.

Thus, the preliminary research pointed out that for enhancing the user's registration process, we should attach a digital compass that can immediately reflect any changes when altering the current pose. Furthermore, the developed solution (i.e. acquiring orientation through location information) became obsolete and it was used only as a backup mechanism, in cases that a dedicated sensor was not available. At that point in time, orientation sensors were not usually embedded in mobile devices and even GPS sensors were scarcely attached to widely available commercial solutions.

- For certain applications (e.g. navigation), orientation context is as important as location context.

Apart from the context variables, which are bound to a specific user, there is more information that can be accumulated in order to assist the registration process.

- Temporal information can assist the user registration process.

The date and time can be translated to the position of the sun on the virtual sky. As a result, the users can use the source of the 3D light to align their current stance according to the time of day.

Different visualisation perspectives are required in order to match the cognitive spatial environment with the real.

- Several customisable visualisation perspectives support the user's registration process more effectively;
- Multiple visualisation perspectives provide better assistance to the users in matching their cognitive map with the real environment.

The implemented solution offers three distinct perspectives, which can be effortlessly altered according to the user preferences: the egocentric, the allocentric oblique and the allocentric plan views. Therefore, the user can visualise the immediate environment and its features according to the current needs. Although the three different views depict the same surroundings, the volume of information is not the same.

- In some cases (e.g. exploration) the users may want to visualise all features in every direction around them (i.e. allocentric view);
- In some other cases (e.g. primed search) it is more important to visualise only what is in front of the user (i.e. egocentric view).
- The registration process becomes easier when the initial visualisation perspective is aligned to the user's physical behaviour.

Despite that, under certain circumstances the users may not be able to observe every feature of the environment that lies in front of them. This can happen when he stands very close to an object, which occupies most of the screen real estate. As a result, the user must either move or change direction, in order to register successfully. In that case, a new solution is required to overcome the issue.

- A customisable field-of-view helps the users register more effectively in the environment.

By customising the current FOV according to their preferences or according to the proximity of an object, the users possess the ability to visualise a wider or narrower segment of the environment.

- A customisable field-of-view can reduce information overload.

That is important in cases where the users may want to discard any irrelevant features after they have focused on a specific target.

Another factor that has also been verified by other research projects examined in the Literature Review is:

- The presentation of important natural landmarks in the virtual environment can assist the user's registration process.

The users become aware of their actual position, by establishing a connection with the landmark position in the virtual and the real world. There are a variety of places (e.g. park), buildings (e.g. tower) or objects (e.g. statue) that can be considered as landmarks, which may transform with the passing of time. Apart from the exterior design changes that may occur to the shape (e.g. renovation) of the landmarks, their façade and texture (e.g. colour) may also change. For these reasons, it is very difficult to accurately reconstruct every available landmark in the VR environment on the fly.

- Variation in building heights is useful for establishing links between real and virtual worlds.

Furthermore, it has not been easy to maintain a recent photorealistic impression that can be used when texturing the 3D objects. The solution, which came up during the Preliminary Evaluation, demonstrated that:

- Descriptive annotations about the object's identity and its primary function must be overlaid on top of the landmarks when high quality 3D graphics is not available.

These annotations can be considered as a replacement to the photorealistic outlook. Although the reconstruction of complete virtual worlds is less expensive and time-consuming than it was a few years ago, it is still very difficult to automatically develop virtual environments of high verisimilitude. As a result, landmark representation has been one of the reasons that have directed the researcher to develop a new visualisation interface (i.e. AR) capable of supporting photorealistic features.

The Preliminary Evaluation also demonstrated that at the beginning of each interaction with the system, the users felt overwhelmed because they had to accomplish a specific task, but without having a priori knowledge (i.e. naïve search) about the environment.

- A *manual* interaction mode with the virtual environment helps the users to form a better understanding of the wider space that surrounds them before initiating their task.

Apart from the sensor-bound mode, the following versions of *Aura* established the manual interaction mode. In the *sensor mode*, position and orientation is inherited from the values generated by the GPS and compass receivers respectively. In the *manual mode*, which can be selected at any time, the users are free to navigate in the virtual environment and collect information, which can assist their decision-making process. This is particularly useful when there are several options (e.g. paths) that could be selected for accomplishing a task.

- The time required to explore the virtual environment is significantly reduced, by introducing both a sensor and a manual modes of operation.

The last element, which assists the registration process, is the direct presentation of context descriptions to the user.

- Representing real-time context in textual or graphical forms to the user assists the registration process.

Although the presentation of numerical variables, which describe the position coordinates of the user on the earth, may not appear very supportive, the presentation of the current orientation parameters is more reasonable. Apart from numerical values, the current orientation can be illustrated in the form of additional visual aids. This can be either a mini map, which constantly changes direction to represent the real behaviour, or a compass-like reference. As a result, certain users who have superior sense-of-direction skills may effortlessly register to the environment by observing such aids.

8.1.3 Movement with the Device

The third aspect, which was found very important and was explored in the Preliminary Evaluation, is interaction in space. The most frequent type of interaction in the context of this research project is movement.

- The framework should offer adequate support in simulating and assisting the user's movement while trying to achieve a goal.

The accurate representation of movement in a VE depends on two factors.

- The quality of sensor data influences the representation of movement in a virtual environment;
- The quality of the 3D entities influences the representation of movement in a virtual environment.

These entities can either be the digital world (i.e. 3D model of the surrounding area) or any other 3D element that represents contextual elements of the natural world (e.g. a remote user). In addition to these entities, which can be queried for acquiring their details, there is supplementary content that should be apparent to the users in order to assist them with their task. This 3D content can be considered as visual aids, which inform the user about the current situation or about subsequent interactions. Furthermore, these visual aids must be produced and amended on the fly, according to real-time user input. Apart from the quality of sensor data, the environmental representations, the modelled remote entities and the visual aids, the system must offer certain functionalities, which will complement the user behaviour. All of these consist a group of features, which must be taken into consideration when concurrent interaction in both worlds takes place.

- Collision detection between the user's representation and buildings in the VR scene is required to avoid occlusions between camera position and user position in real world scene.

During the expert evaluation we found out that:

- Real-time orientation information is required by most scenarios in order to have a satisfactory simulation of movement.

Apart from heading, which is an important variable for navigation, a digital orientation sensor can produce data about the other two axes; pitch and roll. This supplementary data can be used according to the requirements of the scenario.

- Real-time pitch and roll information can describe movement in 3D space (i.e. 6-DOF);
- Real-time pitch and roll information can be translated to meaningful actions in the virtual environment.

For instance, in a navigation application, location (i.e. position coordinates and altitude) and orientation (i.e. heading, pitch and roll) information can fully describe the activity of the user. On the other hand, in an entertainment scenario, pitch and roll can be used to interact with a remote entity, in the same way that people use gestures to communicate. Furthermore, examining these variables may trigger non-trivial in-application functionality, such as altering the bound perspective according to the way that a user is holding the device.

- A digital orientation sensor is required to support advanced user interactions or gestures;
- Position information should be used to derive heading and pitch if a digital orientation sensor is not available.

Although accuracy will be sacrificed, *Aura* can derive information about heading and pitch by examining changes in location, which could effectively trigger alternative functionality.

A further objective of the Preliminary Evaluation was to verify that the implemented functionality assisted the user while moving in space. The first examined feature was a virtual line (i.e. red carpet), which was superimposed on the path that the user had to follow. Although that is an invaluable feature that has been implemented into several location-based applications, the results of our tests showed that it needed to be partially amended in order to reflect its full effectiveness.

- The route line navigational aid should be evident at any time and from every visualisation perspective;
- The route line should be dynamically altered according to the bound perspective;
- The colour of the route line should vary according to the surface that it is overlaid on.

Therefore, in further implementations the route line was multicoloured.

- Just a virtual route line is not adequate for effective navigation support;

- Distinct 3D elements that describe a point of interest should be apparent so that the users can easily comprehend that they have reached a decision point and that they have to react accordingly, if at all.

The position of the 3D content is important because it connects the current behaviour or movement with the subsequent. Thus, it becomes easier for the user to make a decision, which will bring him or her closer to the ultimate goal.

Another feature, which was found particularly useful, is:

- The users should be able to visualise the previously occupied route.

The followed route should be available to the users in cases where they need to review their behaviour and find out about their progress. To implement this feature, the application needs to examine the GPS track logs so that it can reproduce a virtual path according to the spatial, as well as the temporal information. This way the actor can examine the preceding behaviour and put it in the context of the whole task in progress. The result should be a simulation of earlier interactions between the user and the system. Although simulating real-world behaviour in a virtual environment has proven very useful in most cases, there is another factor, which must be taken under consideration.

- The latency and update frequency of the on-device visualisations influences the simulation of user movement.

It has been found that under certain circumstances real-time simulation of the user behaviour might prove distracting. The reason is that very frequent screen updates affect the processing performance of the device, which might skip certain playback frames to compensate. Consequently, the visual output does not appear to be smooth and may confuse the user. Because graphics latency is important, it needs to be managed according to the application requirements and according to the device specifications.

- If the device hardware permits the screen to be redrawn in real time, then it should become the default behaviour;
- If the device capabilities are not high, a hardcoded latency variable should be established.

The last feature, which affected movement in the environment, is the informed selection of a visualisation perspective. Although this feature was not implemented in the version that was used in the Preliminary Evaluation, the user's feedback pointed out that it could become beneficial for subsequent versions. Certain advantages can be observed by automatically altering the visualisation perspective according to the user speed. The proposed solution selects the egocentric perspective when the user is moving slowly and the allocentric oblique and allocentric plan views when the user is moving with average or high velocity, respectively. We mentioned earlier that during the Preliminary Evaluation the automatic selection of a different visualisation perspective according to the subject's speed of movement was not exposed to the participants of the study. In contrast, the participants were allowed to alter the bound perspective through the user interface controls of the application. To begin with the advantages of this solution, several operational resources of the mobile device are released because there is no need to render every single feature of the environment in high quality when the user moves above a specific threshold.

- An abstract (i.e. allocentric view) representation of the environment is more suitable when the movement speed is fast.

Alternatively, when the users are searching for something in particular (i.e. primed search) the focus must be drawn on what is ahead of them (i.e. egocentric perspective).

- A detailed (i.e. egocentric view) representation of the environment is more suitable when the movement speed is slow.

Another reason for establishing this feature is that, when using certain means of transportation (e.g. train), it makes more sense to have a more abstract but also complete view of the surroundings, in contrast to the view that should be obtained while walking. The users, then, can proficiently appreciate the environment and their actual proximity to the target or any other objects.

8.1.4 Decision Points en Route

The last element, which the Preliminary Evaluation intended to examine, is the level of assistance that *Aura* offers to its users when they reach a decision point. Decision points are specific locations or waypoints in the world where the user is required to change behaviour. They can either be a target location or a means to reach the target location. When a dedicated scenario has been applied, a decision point is in effect a POI, which is relevant to the task that the user is involved in. In verbal communication, directions are offered by mentioning distinct elements of the environment, such as a corner of a road or a landmark. Directing a person through the use of a digital medium inherits similar characteristics.

- Landmarks should be apparent when users reach decision points.

The cognitive map of the user is enhanced by relating the position of a decision point to the position of a landmark. The fact that some users would have preferred textual directions at decision points, suggested that an AR solution may have been preferable.

- Accurate illustration and overlaying additional information on landmarks in real-time requires the use of a dynamic photorealistic interfaces (e.g. AR).

With a fully functional AR interface the decision point can be examined from any direction. This has been found particularly useful in cases where the decision point is not directly visible by the user (e.g. the entrance of a shop is on the other side of the building). This observation extends the following one as well.

- A representation of the current target should be always visible on the device screen.

This way the user is always aware of the task that needs to be accomplished. While interacting with the system, if the following decision point is not apparent, there will be a delay until the user processes the presented information. This is exactly the opposite outcome of what our application is aiming for, especially while using a mobile device. That is because interactions with the mobile device are more frequent but less time-consuming compared to the interaction with desktop computers. Consequently, the selected mobile interface should always present the subsequent decision points as well as other visualisation aids, which will assist the users to reach their destination. Furthermore, an oblique angle of 45° can lead to the user not being able to see the decision point that they are navigating. Using a more planimetric view, a wider field-of-view, including an avatar to represent user position, or allowing the user to control these factors through screen interaction could prevent this problem from occurring.

- An allocentric plan view of the environment can assist the users in finding their target location;
- A wide field-of-view can assist the user in recognising a decision point.

8.2 Extensive Evaluation Results

The Extensive Evaluation focused on assessing the effectiveness of the developed framework as well as its usability features. The application that has been tested is the latest version of *Aura*, which implemented every requirement that was found crucial for the operation of a context-sensitive mobile system. This version included two distinct interfaces (i.e. VR and AR), which offered a different approach for the users in order to accomplish their task. Different types of information (i.e. numeric/text or graphics) were presented to the user according to the particular interface. *Aura* was evaluated in a navigation scenario. The reasons that a navigation scenario was selected are presented in Chapter 7.1.2 of the report. Each user interface provided different functionalities and level of assistance to the users. The Extensive Evaluation produced invaluable results for the operational features of each interface. The structure of the Questionnaire is presented in Appendix XIII. The results were not individually analysed because we chose to assess the solution as a distinct whole. The collected data has been attached to Appendix XIV. It has been very interesting to examine the reaction of the participants while using their real-time context as a way to interact with the system. Furthermore, we have assessed the unique features of each interface, by directly comparing their functionalities. Consequently, we have verified that the objectives of this evaluation task have been successfully met. Chapter 8.2.1 presents the results from the objective tests, whereas 8.2.2 and 8.2.3 present the results of the subjective assessments.

8.2.1 Effectiveness

The first domain, which was examined, observes the effectiveness of the framework. In more detail, the first objective was to compare the task performance of the users while consulting each visualisation interface. In order to verify which interface was more useful for the navigation task, we needed to study the results of the objective measurements of the evaluation. The results have been analytically documented in Chapter 7.6.2. The measurements that produced significant results, which were found relevant to the user's task performance, are *Total Time*, *Minimum Speed* and *Average Speed*. It has been found that:

- Users spent significantly less time to complete the wayfinding task while using the VR interface compared to AR;
- The users' *Minimum Speed* was significantly higher in VR rather than in AR;

- The users' *Average Speed* was significantly higher in VR rather than in AR.

Although these results are representative of the effectiveness of each interface, there are some constraints that must also be explored. Chapter 7.5 showed that:

- A large proportion of the participants had never used AR prior to our test. In contrast, the use of VR was far more prevalent.

The lack of expertise with AR systems may have been the reason, which partially influenced the outcome. Considering the number of mobile AR applications that have started flooding the market, we can assume that in the immediate future more people will be exposed to AR systems. Consequently, although VR users performed faster compared to AR users, we expect that with the passing of time the significance of these measurements will drop. Another issue that affected the results of *Minimum Speed* was that VR users did not have to stop moving when they reached a decision point, because the navigational aids were always visible on the device display. In contrast, the likelihood for AR users to stop completely when they reached a decision point was higher, because they needed to pan the device around them, in order to locate the following waypoint. This is one of the disadvantages of the AR prototype, which can be rectified in a following version of the interface, by introducing further features capable of overcoming this issue.

The second objective in the domain of system effectiveness, which was investigated during the Extensive Evaluation, tried to explore if the use of the system architecture and its implemented features helped a user to make better decisions regarding the applied wayfinding task. Similarly to the previous objective, we compared the two available interfaces of *Aura,* in order to measure which one was found more useful by the users. For the purpose of this objective, we examined the subjective responses of the participants, which yielded significant results. Therefore, the questions that were found relevant to this objective are S1 and S5. The responses of S1 demonstrated that:

- The VR interface offered more effective support for the wayfinding task in comparison to AR.

The *Median* of the responses provided for VR (i.e. 5 out of 5) shows that the users found the VR interface very useful for accomplishing their task. Although the *Median* for AR was lower than VR (i.e. 4 out of 5), it shows that certain users believed that the level of assistance provided was adequate, or better, for the applied tasks. Similarly to S1, the results of S5 depict that:

- The users can find their way in an unknown place more effectively when they use VR rather than AR.

The results of S5 are more absolute than those of S1, although the *Median* values for both are identical. In S5, the minimum value for VR was 4 out of 5, whereas for AR it was 1 out of 5. The results of both questions demonstrate that the implemented system's features offer valuable feedback to the users, which help them facilitate informed decisions for accomplishing a wayfinding task.

The third objective about the effectiveness of the system architecture compares the two available geo-referenced interfaces in terms of their ability to represent the environment and its contents, so that a user becomes aware of the task that he is involved in, and how to accomplish it. In order to make a comparison between the two alternative interfaces, we needed to describe the same area, which should include the same number of entity representations (e.g. POIs). This way the benchmark is identical for both UIs In contrast, each interface utilises unique means in order to present information about each entity, and provides independent support on how to reach or interact with it. The subjective results of S5 illustrate that the VR interface is more useful for a user that is trying to find his or her way in an unknown environment in comparison to AR. This means that a 3D representation of the world (i.e. all perspectives) makes the users comprehend the layout of the surroundings faster, so that they may proceed pursuing the task at hand. Furthermore, posing a more comprehensive question about which interface was more helpful in informing the participants about their current activity, we managed to get significant results relevant to this objective. The results of subjective question S10 demonstrated that:

- The VR interface and its features were more helpful in informing the user about the current task than those of AR.

Consequently, we can deduce that the sum of implemented features in VR represent information about a specific area in a more constructive way. In addition, tracking and directing further user activity is handled better by this interface, mainly because of the visual aids that have been applied. As a result, the use of VR makes a user form a more complete cognitive map about the surrounding environment as long as the represented scene describes the important physical world entities accurately. Gesturing was not found to be the most suitable solution for retrieving activity information because it seems that most users are bound to traditional alternatives (e.g. manual input) due to their familiar behaviour. It should be acknowledged though, that after several interactions with the AR interface, gesture-based activity querying may score better in the user preference scale.

8.2.2 Usability

The second domain that was examined during the course of the Extensive Evaluation tried to identify which features of the framework positively influenced the usability expectations of the users. The first objective of the usability domain intended to examine, which was the users' preferred virtual environment or technology while they were moving on foot. The answer to that question is particularly influenced by the subjective responses, which were documented by the last part of the Extensive Evaluation questionnaire. In more detail, question ABT-S4 collected responses about the interface that was mostly preferred by the users for accomplishing their wayfinding tasks. The responses to this question were recorded after each participant experienced both conditions. As a result, ABT-S4 should offer a good indication about the overall user preference. Although the performance measurements demonstrated that the VR interface was significantly more effective for wayfinding, the results of this question were ambiguous. The reason is that 12 out of 23 participants preferred VR for wayfinding, whereas the rest leaned towards AR. The evidence point out that the opinion of only one person (i.e. 4.4%) is not adequate for reaching a final verdict on this issue and get a significant outcome. Conversely, the results have shown that:

- Even though most users performed faster with VR, a large proportion still prefers to use AR.

This means that the AR interface has several interesting features, which positively affected the users. The enhancement of these features in subsequent versions of *Aura* may prove beneficial and render better results for the AR interface. Because this was a subjective assessment, another important factor that influenced the responses of the users, must have been the level of satisfaction and excitement that they experienced while interacting with each interface. The subjective responses that have been recorded for questions S28 and S29 support the findings of ABT-S4. In more detail, question S28 measured the enjoyment while using each interface and S29 measured the excitement or entertainment that each interface offered. Because the results of these questions are not significantly different, we have chosen not to present them in detail in the Evaluation chapter. The *Median* for S28 and S29 in both interfaces is the same, 5.0 and 4.0 respectively. This means that:

- On average, each interface provides the same level of joy and excitement to the user.

This observation proves that the result of ABT-S4 has not been randomly generated because it is also supported by S28 and S29. In conclusion, the final UI preference is not affected only by the performance measurements, but it is counter-balanced by the level of joy and excitement that it offers to the users. Thus, we believe that the pedestrians involved in our experiment marginally favour VR for accomplishing a wayfinding task, primarily due to the number of implemented features and their background expertise.

The second objective of the usability domain intended to identify which is the favoured visualisation perspective of the users while walking. The developed VR interface offers three options (i.e. egocentric, allocentric oblique and allocentric plan views), whereas the AR interface offers only one (i.e. egocentric). Similarly to the first usability objective, the result of the second objective has also been influenced by the subjective responses, which have been provided to question ABT-S3. This question aimed to identify the overall favourite visualisation perspective and it can be therefore used to directly influence the results of the second usability objective.

- Users preferred the allocentric oblique and allocentric plan views equally (i.e. 39.1%), whereas the egocentric perspective received fewer responses (i.e. 21.8%).

From this question, we can assume that a map-like interface (i.e. allocentric plan) is equally favoured to a pure 3D interface (i.e. allocentric oblique) because the vertical view can only be established in these two alternatives. Thus, VR is more resourceful and convenient to use because it offers the option to switch between the available alternatives. Although the egocentric perspective received the fewest responses in question ABT-S3, question ABT-S2 tried to identify which is the preferred interface for depicting an egocentric view of the environment. The results of ABT-S2 show that:

- An egocentric AR view (i.e. 60.9%) is favoured by the users to an egocentric VR view (i.e. 39.1%).

The principal reason that influenced this outcome is the realism that is offered by this interface, which is explored by question S32 and discussed in the following paragraph. Furthermore, the substantial difference between the responses in ABT-S2 confirms that when an egocentric representation of the environment is required, AR can prove more effective than VR. If we consider the significantly higher cost and effort that is essential in order to represent the surroundings in a simulated rather than in a see-through system, we can deduce that the egocentric AR view offers a valuable alternative to the preferred visualisation perspective. A reason that influenced the answers of the participants in ABT-S3 is that the requirements of the scenario instructed them to interact only with the environment and not with another entity (e.g. POI or remote user). As a result, the full potential of the egocentric perspective was not exposed. This expert observation shows that:

- The allocentric oblique and allocentric plan perspectives are more valuable when a user interacts with the surrounding environment;
- The egocentric perspective is better when the user needs to interact with a single entity of the real world that is directly visible by the actor.

Another observation is that the purpose for introducing a multi-view framework has been achieved. The applied scenario instructed the user to accomplish a naïve search of the environment followed by a primed search. Exploration was not included as a wayfinding task of the evaluated scenario.

- For accomplishing the naïve search task, most users referenced the allocentric plan view;
- For the primed search, the allocentric oblique perspective was favoured when they had to locate and follow the route points.

The last objective of the usability domain aimed to identify if the use of AR and VR interfaces could enhance the enjoyment that a mobile context-aware service offers. The analysis found in Chapter 7.7.2, provided significant results on two questions, which can influence this objective. In more detail, question S31 collected responses on whether the users would like to use each interface in contexts other than their profession. The results illustrated that:

- Most users found VR to be more satisfying than AR and they would like to find more applications or services that make use of it.

Following next, S32 compared the realism level of each interface.

- The represented scene realism was found to be significantly higher in AR compared to VR.

This outcome was not unexpected because AR offers a real-time photorealistic representation of the environment, whereas VR offers a simulated representation, which is produced out of spatial data that has been collected at random intervals. Even though all effort has been made to develop a realistic model of the environment, it was impossible to simulate absolute reality in real-time for the purpose of this project. Thus, a significant difference between the realism of the two interfaces has been identified. Another two questions, which did not offer significant results while comparing AR with VR but can directly influence this objective, are S27 and S33. Question S27 demonstrated that the participants would like to use the VR interface (*Mdn*=5) a bit more than AR (*Mdn*=5). Conversely, question S33 shows that the users had been more easily immersed in the augmented environment (*Mdn*=4) rather than in the virtual alternative (*Mdn*=4). Based on the significant and non-significant results of the previous questions, we can conclude that:

- The level of joy that each interface provided to the users, is not significantly different.

Both interfaces were found enjoyable, which explains that attaching either of them to a mobile context-aware service can improve the users' satisfaction, with variable efficiency according to the underlying application purposes.

8.2.3 Technical

The third domain that was examined during the course of the Extensive Evaluation considered the performance issues that may have affected the user experience. The first objective measured if the accuracy of the location sensor (i.e. GPS) has proven satisfactory when used for real-time positioning in both interfaces. Although the underlying software architecture and the polling interval was identically set up for both interfaces, this question aimed to investigate if the visual feedback was satisfactory for the user. The reason is that, although every update had been recorded when it took place, there was a minor hardcoded delay attached. The first reason for introducing the delay constant was to minimise the rendering operations, which took place on the device screen because they stress the hardware resources. The second reason was to reduce the continuous updates, which are required to represent the user's position and orientation adjustments. In more detail, if the image got redrawn every time that a new reading became available, then the graphics would seem to flicker. That is because the sensors offer several measurements per second, which are translated to several adjustments in the selected interface. The comparison of the positioning accuracy in both interfaces did not produce significant differences. Question S35 of the subjective assessment demonstrates that:

- The accuracy of the positioning sensor was considered adequate for both interfaces in the navigation task.

The Median for both VR and AR was found to be 4 out of 5, which shows that there was no negative effect on the user activity. The second objective of the technical domain measured the accuracy of the orientation sensor (i.e. digital compass). Similarly to the previous objective, a significant difference in the accuracy of the orientation sensor between the two interfaces was not observed. Question S36 measured how accurately the physical orientation was depicted in the virtual environments. Although the AR scores (*Mdn*=5) were a bit higher than those of VR (*Mdn*=4), the difference was not significant. From these scores, we can deduce that:

- The accuracy of the underlying subsystem is adequate for representing the physical orientation of the user in both interfaces in real time.

To summarise, the accuracy and performance of the context acquisition entity (i.e. CMS) of *Aura* has been rendered sufficient for use in a context-sensitive service applied in a wayfinding scenario.

The last objective of the technical domain tried to identify how the performance of the developed system architecture (i.e. IPS) affected the information visualisation and interface interaction requirements of the user. For this objective, two questions produced significant differences between the utilised interfaces. Question S38 measured how prompt the response of each interface was to the manual input of the user. It was found that:

- The AR interface responded significantly faster to the manual input of the user compared to the VR interface.

A possible interpretation could be that the VR interface is heavily utilising the graphics components of the device, which means that a slight latency between the user action and the resulting feedback may occur. This happens because the graphics pipeline must render every 3D object of the surrounding environment as well as process the required information output. Following next, question S42 measured the quality of the textual information that was presented on each interface. This question produced substantial results as well. It was found that:

- The text style (i.e. colour and size) was easier to read in VR rather than in AR.

The main reason that triggered these responses could be that the software version that has been employed for the experiment, did not implement a dedicated algorithm in order to recognise the background pixel colours so that it could adjust the colour of textual output accordingly. From the participant responses, we can deduce that this is essential functionality for the AR interface, which should be implemented in a following version of *Aura*. Useful information on how to further explore this issue can be found in the following publications (Leykin and Tuceryan, 2004) (Gabbard et al., 2005) (Gabbard et al., 2007). The responses collected from the other *System Performance* questions of the subjective assessment, verify that the participants found every aspect relevant to the performance of the IPS entity of *Aura* satisfactory. The *Median* produced by each question, which is equal to or more than 4.0 out of 5.0, supports this expert observation.

Further Investigations

The final domain of the evaluation has two distinct objectives. The survey included several open-ended questions for examining these issues. The first one intended to discover new potential applications, which can be supported by the framework. For the first objective, the participants provided answers to a single question for each visualisation interface. The purpose of question S13 was to explore what kind of mobile applications may be suitable for each interface. For VR, the following responses have been documented. Due to the fact that the evaluation scenario focused on wayfinding tasks, 9 participants believed that a complete satellite navigation solution would expose the best features of the framework. Furthermore, 7 participants believed that a POI locator, which would also offer further information about landmarks or other users, could expose valuable services. Following next, 10 participants mentioned that a mobile gaming solution would be commercially viable. The main reason for receiving so many replies relevant to the entertainment domain could be that most people have experienced VR in a gaming scenario. The last group of answers, acknowledged by 3 participants, involved a mobile advertising or marketing solution. Respondents also noted that the use of *Aura* in applications such as traffic management, transport services or more specialised confined space solutions could prove beneficial. For the AR interface, the participants believed that the following applications might be significant. Satellite navigation received 6 replies, whereas 10 participants would like to query POIs by using AR. Furthermore, entertainment applications received 4 replies. A large proportion (i.e. 8 participants) believed that an advertising application would be quite beneficial and innovative, compared to the existing mobile marketing solutions. Finally, the last group of applications, which are worth mentioning, belong to the public transport and social networking domains. The recorded responses, especially for AR, support the commercialisation plan, which has been developed during the course of this project. The plan can be found in Appendix XVI of this report.

The last objective of the Extensive Evaluation intended to examine whether the framework can be commercially viable. Similarly to the previous objective, examining two open-ended questions of the survey has produced the answers. Question S34 investigated the amount of money, which the participants were willing to invest in a useful application that implemented each individual interface. On the other hand, the purpose of ABT-S5 was to identify the cost, which the participants were willing to pay, in order to receive an application that implemented both interfaces. The full set of responses can be found in Appendix XV. Although these two questions provided interesting results, further market analysis is required to determine a certain suggested price. Most participants added the prices, which they had provided to question S34 for each interface, in order to get the total for ABT-S5, including a minor discount in some cases. We also noticed that some participants were willing to obtain a rolling subscription (e.g. monthly or annually) in order to receive the required services. Generally, the feedback to these questions was very informative because it depicted very satisfying values, which were higher than our initial estimations. Thus, it is our belief that a well-tuned application that makes use of the features found in the developed framework can offer benefits to the users, as well as to the developers.

9 Conclusion

This chapter provides a summary of the topics that have been explored in this research project. The initial aims and objectives are examined, to verify how they have been satisfied for the purpose of the research. The sections that follow present the overall contributions made through this research as well as a critical analysis of the results, including the identified limitations and the recommendations for future work. These recommendations will allow overcoming the identified limitations and could assist in the production of a tangible commercial solution out of this research project.

9.1 Research Summary

This research project examined whether the representation of real-time contextual information on two distinct interface paradigms can prove beneficial for the information needs of mobile users. Although the use of sophisticated mobile devices that have embedded several types of context-sensitive sensors has increased in the past few years, end-users, still, cannot take full advantage of the available technologies. The main reason is the lack of certain methodologies that can enable visualisation and interaction with information objects in real-time from a mobile device that its location changes frequently. Innovative services can evolve, if the users are presented, in situ, with relevant information about themselves and/or about other entities that exist in the real world. The users are not only interested in the functionality provided by the mobile application, but also in the overall experience while operating it. Therefore, some mobile applications that represent the user's immediate environment offer advanced user interfaces that reference the real world. Connecting the real-world elements that are particularly relevant to the user's needs with the virtual environment can provide several information benefits on a wide range of application domains. One broad aim of this project is to examine the user requirements for a context-sensitive mixed reality system that can contribute to the satisfaction of their goals across several application domains.

One way of implementing the connection between the real and the virtual elements is by examining the contextual information of both the user and the selected real-world entity. Thus, the user can visualise the representation of the remote entity on his or her device and virtually interact with it from distance, through an established network infrastructure. But in order for this information communication to be both effective and pleasing for the user, certain visualisation, interaction and collaboration requirements must be satisfied. Another broad aim of the research is to identify these core user requirements for a system that processes context ubiquitously and in real time. The only interface that can offer truly ubiquitous operation without the need to model or intervene in any way to the real environment and has the potential to satisfy the aforementioned requirements is a context-sensitive augmented reality system.

The missing link that connects the satisfaction of the users' information needs with the already available technologies is a framework which can satisfy both types of user requirements that were mentioned earlier. The framework should acquire and distribute contextual information in real time and represent relevant to the user information ubiquitously through geo-referenced interfaces. The third broad aim of this project was to design and develop a general-purpose framework that accomplishes this goal. After specific customisations take place the framework should be able to serve several application domains. But in order to evaluate whether the framework satisfies both the high and the low-level requirements, a more focused implementation has been developed. The application domain that was selected is urban navigation and the main reasons were because it could present the ubiquitous operational capability of the system and also because it is a scenario rich in environmental information, as well as contextual information both from the user and the remote entities.

The research undertaken so far has demonstrated that a single platform, which enables interaction between users and selected environmental features, as well as collaboration between ubiquitous users on a mobile context-aware environment using both AR and VR as visualisation and interaction means, does not exist. Accurate acquisition of the requirements, analytic system design and precise implementation of the novel architecture have been some of the most challenging tasks of the research process. Each phase of the evolutionary rapid prototyping approach that has been followed intended to provide answers to the applied research questions. Initially, we organised a Requirement Acquisition Survey (Chapter 3.3.1) in order to collect the core requirements regarding the visualisation, interaction and collaboration features which the users expected a real-time, ubiquitous, context-sensitive system to offer. The answer to the 1st Research Question has been provided mainly by the results of this user survey. The requirements generated by the Requirement Acquisition Survey have been appended to requirement list presented in Chapter 4.1 and 4.2. Furthermore, these requirements have also been introduced to the framework design. After producing the first-cut prototype that facilitated context-sensitivity in VR, we decided that it was time to evaluate the developed functionality. Therefore, a Preliminary Evaluation task (Chapter 7.2 and 7.3) took place which intended to provide an understanding about the effectiveness of our solution in satisfying the users' main expectations in a wayfinding task. The results of this expert evaluation task influenced answering the 2nd Research Question and provided additional requirements for the 1st Research Question. Answering the 2nd Research Question also required further investigations, which are presented in detail in Chapters 3.3.2 to 3.3.6 of the report. After examining the results of the Preliminary Evaluation the requirements list took its final form. The full requirement specification influenced the overall design of the framework (Chapter 4.3) and provided hints that the introduction of a second interface paradigm was required to fully satisfy the real-time information needs of the users - augmented reality. The issues related with the application of real-time context on VR and AR interfaces, as well as every other implementation aspect related to Aura is analytically presented in Chapter 5. Therefore, the design and the implementation of our framework produced a technical specification which can be considered as the answer to the 3rd Research Question. Finally, in order to provide an answer for the 4th Research Question, the framework sustained further development so that it could take the form of a mobile guide that processed real-time context and could be used in an urban navigation scenario. Both interface paradigms have been objectively evaluated in this scenario for determining whether the decision-making process of the user, while being immersed in a virtual environment, had been improved by using the mobile client as a communication instrument. Furthermore, certain usability aspects that can affect the behaviour of Aura's users have also been subjectively examined in the Extensive Evaluation. The design of both assessments (i.e. objective and subjective) of the Extensive Evaluation, as well as the results can be found in Chapters 7.4 to 7.7.

9.2 Review of Research Aims

In this subchapter, a description of the aims that directed this project can be found, based on the full listing of the research aims provided in the Chapter 1.5. The description of each aim is followed by a description of the approach, which has been followed in order to satisfy that aim.

The initial aim of the research has been to propose a framework architecture and functionality that can support peer-to-peer interaction and context-sensitive information retrieval, using mobile devices and networks. For this aim, we needed to investigate the available sources of context, which could be queried for up-to-date information, by a mobile application. In more detail, every possible method of acquiring contextual information was examined, as well as its distinct characteristics. Depending on the type of each source, a new software component was developed. The sum of all components formed the Context Management System (CMS) of *Aura*. As a result, *Aura* has at least one paradigm implemented, which illustrates how each type of context source can be queried and processed. Due to the fact that real-time context exchange is required by several framework applications, a new networking layer was designed, developed and integrated. This layer is capable of transporting context as well as its metadata. The attachment of the networking layer to the CMS entity offered a new perspective, which rendered the collaboration of a group of actors feasible.

The second aim of this project has been to develop an application, which will combine geo-referenced 3D content with contextualised user information in real time, to promote collaboration and interaction between mobile clients. In order to achieve this aim, we needed to explore a set of visualisation and interaction interfaces, which could handle the presentation of geo-referenced 3D content. Initially, the characteristics of three interfaces were examined, namely the map, the VR and the AR interfaces. After gathering insightful results, the use of the map interface was discarded and the development efforts were focused on VR and AR, which have both proven particularly constructive for the purpose of this aim. Consequently, the resulting framework encompasses an implementation of each visualisation interface. Interaction with both user interfaces is governed by a single high-level entity of the framework; the Information Presentation System (IPS). The IPS, when coupled with the CMS, forms the core of the framework. The core can represent and direct user behaviour according to the information provided by each individual. Collaboration between individual users can take place in various forms including textual, graphical or activity-based.

The third aim of the research has been to examine the framework against the technical as well as the social aspects of what is expected from a ubiquitous context-aware application, using pervasive scenarios as a case study. For this aim, we needed to discover potential applications, which could present the advanced functionalities of the developed framework. Although the use of the framework is not limited to the conceived applications, the selection of a certain scenario offered a benchmark, which could be applied to measure the effectiveness of *Aura*. The most beneficial scenario was found to be a wayfinding application. Thus, the technological, cognitive and social aspects of the framework were tested several times in order to satisfy the requirements instructed by the scenario. This was accomplished during both evaluation cycles described in detail in Chapter 7, as well as during minor expert evaluations, which took place due to the selected development methodology. In order to fully meet the requirements of this aim, the developed system embraced certain user-centred functionalities such as personalisation and privacy mechanisms. Furthermore, the cognitive issues related to navigation were investigated, so that a point of reference would be set, which could be used to identify the effectiveness of the framework in the selected scenario.

The final aim of this research has been to evaluate the usability of this context-sensitive, integrated system for information retrieval and applied visualisation techniques with currently available location-based services, to identify possible prospects for commercialisation. This aim has been approached from the perspective of potential adoption of the system either by end-users or by stakeholders. An analysis of several potential applications, which could evolve from the customisation of the framework, has been accomplished. Furthermore, an analysis of the optimal strategy for bringing a commercial solution to life has been documented. Although the commercialisation plan may not be implemented exactly the same way as it is described due to market evolution issues, the details provided in this report may prove extremely beneficial in such efforts. The Extensive Evaluation, which was accomplished at the end of the project, produced significant results about the effectiveness and usability issues of the framework. These results offered very interesting directions, which could be followed to make the framework more attractive to the public. Finally, the extensive user assessment provided certain pointers about the applicability of both visualisation and interaction interfaces in the context of the ubiquitous service that was investigated.

9.3 Review of Research Objectives

In this part of the report, the full set of objectives, as presented in Chapter 1.5, are revised. This way, we can assess the degree to which each objective has been accomplished through meeting the detailed requirements set in the research.

1. To acquire high-level, user-related requirements through modelling user behaviour and to discover connectors with the technical aspects of the system.

For the first objective, we had to identify the underlying functional requirements that would direct further development of the framework. It was crucial to distinguish between the high-level functionality expected by the users and the low-level requirements that affected the implementation of the system architecture. In order to collect the high-level requirements, we initiated a qualitative *Requirements Acquisition Survey*. The questionnaire was distributed to a number of people who had average background experience on the topics explored by the project. Chapter 3.3.1 of this report presents the results of the survey. The questionnaire collected the participants' responses about information management, visualisation, interaction and collaboration issues. The results of this task encapsulated the features, which the participants required from a context-aware system. Following next, modelling the user environment in order to represent its features on an UI, capable of supporting advanced input and output options between the user and the physical entities, provided further instructions for the improvement of the framework. This was found particularly important in order to support the ubiquitous operation of the system. In addition, the cognitive issues related to wayfinding were examined, so that interaction with the virtual environments would provide valuable assistance to the users. These results have been published in the *International Conference on Computer Graphics Theory and Applications* (GRAPP) (Liarokapis et al., 2006b). The last set of high-level requirements was introduced by examining the details of other context-aware systems. The focus of this approach drew particularly on the fields of collaboration and entertainment. This way, we managed to collect knowledge, which renders the operation of the system not exclusive to a certain domain of applications.

2. To identify and include in the implementation low-level, technical requirements and specifications that can describe the full functionality.

For this objective we needed to research the available technologies that would be selected to develop the framework. The technologies were separated to hardware and software-based. The hardware-based included mobile devices, sensors and network infrastructure options, whereas the software-based consisted of the underlying development platform and 3rd party software libraries, which could enhance system operation. The accomplished research produced several non-functional requirements spanning across the following domains: *Device*, *Operational*, *Performance* and *Portability* requirements. An analytical description of these requirements can be found in Chapter 4.1. Apart from the non-functional requirements, several *Functional* and *Usability* requirements have been identified. These two sets established a connection with the higher-level requirements, analysed in the previous objective, and revealed some core usability issues, which were found indispensable for the users. The *Functional* and *Usability* requirements are described in detail in Chapter 4.2.

3. To design and develop a mobile data communication protocol for cellular and wireless networks that will be able to transfer user and location context in real time.

The first part of this objective was met by acquiring the requirements relevant to context and information exchange between actors. These requirements have been presented in Chapter 4.1.1.3. Taking into consideration the acquired requirements influenced the design of the networking component, which was the second part of this objective. The design of the framework’s networking component and its implemented features have been analytically presented in Chapter 5.6. The developed protocol is capable of exchanging textual information between users, as well as to transfer real-time context and the metadata describing it. It can work on several mediums because compatibility with current formats has been considered. This functionality promotes user collaboration relevant to the pursued task, within a single mobile application. The source code for the mobile data communication protocol can be found in Appendix IX.

4. To design and develop a flexible archive system, which will dynamically store the user's position, orientation and remote entity contextual variables. It should be concurrently accessible by several users.

Similarly to the previous objective, the requirement acquisition process found in Chapter 4.2.1 influenced the design of the archive subsystem. The design and implementation of this component can be found in Chapter 5.2.1. This component has been designed to store and retrieve information from generic GPX documents, as well as plain log files. Although the use of a database is considered the best practice, the exchange of information between clients becomes straightforward, by following the proposed technique. Furthermore, the introduction of a certain DBMS would reveal new requirements, which could render further use of the framework dependent on the database platform. The introduction of a centralised archiving architecture could present certain benefits such as those described in Chapter 9.5. The source code for the archiving component has been attached to Appendix IX.

5. To implement a robust context-aware, location model, which will be the fundamental element of the geographic component.

The development of a conceptual location model has been a core objective of this research endeavour. The reason is that every interaction relevant to geographic content must be supported by the system architecture. Some functionality supported by the developed model includes finding object positions in the real world, distance querying and orientation of local and remote entities, amongst other spatial functions. The use of a geometric location model has been selected for the purposes of this project. This selection was found suitable because it can support both operating modes of *Aura*. Further information on other types of location models, which may enhance the already achieved functionality, has been provided in Chapter 5.2.2. The implementation of the location model functionalities takes place in the CMS entity of the framework. The source code facilitating the location model functionalities can be found in the Appendices.

6. To develop user profiling and data management mechanisms, with emphasis on enabling peer-to-peer collaboration and interaction.

Due to the fact that the framework should support collaboration between users, a specific component that manages the user details evolved. The user is considered as an active source of context, which means that personal information and preferences must be recorded in order to provide effective assistance. Furthermore, the customisation of a user profile with details about several features of the framework became feasible by introducing this software component. The component allows the management of personal details, preferences and privacy options by every user. Despite that, the introduction of user management functionality enabled querying a remote profile by the local user. If the user is interested in the remote profile details, initiating peer-to-peer communication with the other party becomes straightforward. The *Requirements Acquisition Survey* provided valuable input about the information that users were willing to disclose, for various reasons. Chapter 5.1.4 presents the implementation details for managing the user context. The source code for the profile management component can be found in Appendix IX.

7. To apply privacy and security restrictions, which will govern user communications and exchange of information.

Another important feature, which is increasingly anticipated by end-users, is the privacy of sensitive information. The initial *Requirements Acquisition Survey* in Chapter 3.3.1 demonstrated that users expect an application or service to securely distribute personal information only to the parties that they have authorised, and for distinct reasons. Furthermore, several national and European legislations exist, which are concerned with the privacy restrictions governing information distribution and management. The information that is processed by *Aura* can be considered as sensitive because it includes personal details, as well as real-time context. Thus, a software layer was embedded in the framework, which is used to govern the exchange of information. In consequence, the users have the option to allow or deny the distribution of their personal details, as well as their track logs. Separating the recorded details to personal and location-based allows the application not to restrict the expected functionality, even when only one group has not been authorised. Privacy restrictions are explicitly handled through the user profile. Furthermore, due to the importance of this feature, the developed network protocol is overridden in order to safeguard the user's privacy rules. The privacy mechanisms have been included in the framework's CMS entity in order to meet the acquired requirements found in Chapter 4.2.1. Chapter 5.1.4 presents the implementation details for managing the available privacy restrictions.

8. To develop and combine various level-of-detail interfaces for supporting ubiquitous interaction with the environment and selected elements.

This objective has been comprehensively investigated in various sections of the report. Due to the fact that the framework must support ubiquitous operation, it was vital to develop and employ certain visualisation and interaction interfaces, which would sustain such functionality. Although there have been several applications in the market, which employ highly-interactive interfaces to represent the environment and its features, *Aura* allows its users to interact and visualise remote entities by blending the advantages of more than one UI. This is accomplished by having implemented several interfaces, which can support diverse user needs. Initially the efforts were focused on the development of three distinct UIs (i.e. 2D/Map, 3D/VR and AR) but further development on the Map-like interface was deemed redundant because VR can offer similar or better functionalities. The specific visualisation and interaction requirements, which must be met by each interface, are presented on Chapter 4.2.2 and Chapter 4.2.3, respectively. The implementation of the interface functionalities takes place in the IPS entity of the framework. Further details about the implementation of the VR interface have been provided in Chapter 5.4, whereas for AR, which was one of the main achievements of this project, in Chapter 5.5. The details of our innovative algorithm used for context-sensitive AR have been published in the *International Journal of Computer Graphics* (The Visual Computer) (Papakonstantinou and Brujic-Okretic, 2009a).

9. To enhance the information visualisation framework of a mobile device, in order to support collaboration between actors and stakeholders.

The focus was drawn on this objective after meeting the requirements of objectives 3 and 4. It was crucial to have the networking layer of the CMS as well as the IPS entities of the framework implemented for meeting the requirements of this objective. Communication with remote parties became feasible not only in textual forms but also in graphical terms, which could represent more information on the mobile device display. As a result, the users of the framework can visualise information not only generated by themselves but also from diverse sources. Collaboration between users is promoted by visualising real-time activity as well as personal information. Furthermore, interaction with stakeholders can be initiated either by the 3rd party or the local user. The local user may interact with remote entities bound to a specific stakeholder by querying an object in the virtual environment. Nevertheless, a stakeholder may initiate contact with a user by sending arbitrary information, which is presented on specific POIs. The stakeholders have the option to select which users to contact by examining their real-world behaviour (e.g. proximity to one of their establishments). Certain scenarios that we explored (e.g. m-Marketing) required the advanced visualisation features offered by *Aura* in order to positively affect the users. The detailed framework architecture, which enables collaboration with remote parties and offers advanced information visualisation and interaction opportunities to its users, was presented in the *International IEEE Conference in Serious Games and Virtual Worlds* (VS-Games) (Papakonstantinou and Brujic-Okretic, 2009b). In that publication, the focus was drawn on pervasive entertainment applications.

10. To formulate knowledge-based scenarios, which can be integrated to the software environment, to test user interactions.

The concept of a framework implies that several applications may evolve out of it by customising and enhancing it accordingly. Likewise, *Aura* was developed by taking this concept under consideration. In order to let new applications evolve, the design of the framework was loosely coupled with the implementation details. Several potential applications of the framework have been identified during the research and by exchanging ideas with other researchers and potential users. Furthermore, the commercialisation model, which has been conceptualised and presented on Appendix XVI, illustrates a number of scenarios, which *Aura* could effectively accommodate. One of the identified scenarios has been applied to extensively evaluate the effectiveness of the framework. The scenario was urban navigation and the details of the evaluation process are presented on Chapter 7. During the course of the research, the framework attracted attention from the research and the business community as well. It was an honour to gain an invitation to the *Idea to Product Global* (I2PG) competition, which took place in Austin, Texas, USA on November 2009. In that international competition our project represented City University London. The application that was selected for participation was a mobile Marketing solution (Papakonstantinou and Bhatia, 2009), described in more detail in Chapter 6.1.5.

11. To evaluate the usability aspects of the framework, especially in terms of information visualisation options, in order to enhance user decisions and their application in ubiquitous scenarios.

The last objective of the research focused on evaluating the effectiveness and usability features of the framework. In more detail, a specific application was developed and it was presented to the participants of this task. The application was a wayfinding tool tested in urban settings. The application could run in any environment including previously unexplored ones and it offered certain services to the users in real time. The quality of the provided services was measured against qualitative and quantitative criteria, which can be found in Chapter 7.6.1 and Chapter 7.7.1. In more detail, the effectiveness and usability of the developed application was examined in two conditions. The first one utilised an environment-simulating interface (i.e. VR) and the second utilised a video see-through interface (i.e. AR). Apart from measuring the effectiveness of the whole framework, the evaluation process was invaluable for informing us about the advantages and disadvantages of each interface, in the scenario under investigation. The results of the evaluation process are analytically presented on Chapter 7.6.2, Chapter 7.7.2 and Chapter 7.7.3. They have demonstrated that the use of the features of the developed framework proved useful for the task that the participants had to accomplish and validated the requirements, which had been previously set. The discussion of the obtained results can be found in Chapter 8. The results of the preliminary expert evaluation have been published in the *Journal of Virtual Reality and Broadcasting* (JVRB) (Liarokapis et al., 2006a).

9.4 Overall Contributions

The first significant contribution of this research project is the integration of a context-aware software system, two visualisation and interaction interfaces, a user privacy and personalisation scheme and a dedicated context-sharing communication protocol into a distinct mobile framework. The developed framework, *Aura*, identifies the boundaries for the research carried out in relation to the design, implementation and the individual user related issues. Research in the field of mobile computing and its related disciplines, such as context-aware services and information communication technologies, revealed that real-time context acquisition and management can prove valuable for the ever increasing information needs of mobile users. Due to their frequent mobility, processing spatiotemporal context variables ubiquitously can trigger valuable services for the users, not only by becoming aware of their personal situation, but also for interacting with other information objects that exist in their surroundings in real time and according to their proximity or relevance. Furthermore, due to the vast volume of available information, the framework facilitates the acquisition and processing of user-related context, such as their personal preferences according to the predefined profile. Therefore, the provided services can be personalised to fit the information needs of mobile individuals. The framework does not only offer an intelligent information retrieval engine for exposing to the user the optimal volume of data at any time, but also an information-rich content space to enhance the user experience. The virtual environment can enable collaboration between individuals, based on spatiotemporal information (e.g. proximity) and on similar personal preferences. Such collaboration activity, in addition to immersion in a digital environment, produces interesting communication patterns, which can be further explored to assist the users in achieving their goals. The context variables processed by the framework (i.e. personal and spatiotemporal information) are characterised as sensitive information that can be used to identify an individual. Therefore, communication of such information is accomplished in privacy-enabled mobile network architecture, which conforms to the technical standards as well as to the national and European laws on data protection and privacy. Another issue that has been investigated is the ability for the framework to operate in unknown environments, which is a crucial challenge for every context-sensitive system that offers environmental representations. This challenge cannot be resolved by a unique approach; thus, an integrated approach has been applied in the design of the novel framework architecture, which introduces a visualisation and interaction solution that does not require additional environmental information. The embedded solution is a context-sensitive AR interface.

The second significant contribution of the research is the design of a distinct algorithm that solves the calibration, tracking and scene rendering problems found in traditional augmented reality systems. The main functionality of this algorithm is to superimpose descriptive information on precise positions on the image acquired by the mobile device camera and presented on its display. The on-screen position of the virtual overlays corresponds to a position found in the real world, in most cases, the position of an information object. Therefore, one of the unique advantages of an AR solution is that it superimposes digital representations of information on real-world entities, which provides the means to augment the natural world with potentially useful information for the users. This is a very intriguing concept because it offers means to ubiquitously connect the real with the virtual in real time. As a consequence, AR users can visually explore the contents of both worlds through a single interface and, by combining the output, they visualise real-time context, which otherwise would not be available in a straightforward manner. Another advantage of AR is that it does not depend on geo-referenced computer-generated environments like VR does, which significantly reduces the development and operation costs of such systems. Therefore, it does not require additional environmental information and can work in any environment without prior training. Augmented Reality functionality comprises several stages which are tracking, registering, camera modelling and scene rendering. Traditional AR systems utilise object recognition and pattern matching techniques to accomplish tracking and registration. In contrast, recent developments in mobile device functionalities, such as 3D positioning and orientation systems, digital cameras and networking capabilities enabled the realisation of our novel algorithm into a working software component. Precise position and orientation information, for both the device and the remote element, have superseded the need for object recognition and pattern matching, which was introduced by the early mobile AR systems. Therefore, our novel AR algorithm establishes a new approach for tracking, registering and scene rendering and makes use of a camera modelling technique that was proposed in 1982 (Hall et al., 1982). Nevertheless, AR - as a new technology in the mobile context, poses several challenges that influence the operation of such systems. During this project, we found that the technical issues involved in the implementation of a working AR solution are not trivial. The primary reason is the incompatibilities between the available mobile platforms, as well as the performance issues associated with each mobile device. Furthermore, context reaction greatly influences the usability of a mobile AR solution because there are several sources that produce real-time context, which have to be queried according to their type.

The third significant contribution resulting from this work is a comprehensive system development methodology that enhances the functionality and the mobile users' experience of a mixed reality system which processes context in real time and dynamically offers information services to its users in situ. Apart from the technical issues, the research needed to explore issues related to individual users and their potential interaction and collaboration activities. In order to manage such issues, we needed to research across several scientific fields; including human-computer interaction, cognitive, as well as the social philosophies. Although this is a research project, a distinct system development methodology has been proposed, consisting of several analysis, design, implementation and evaluation phases. Its uniqueness lies in the context that the resulting framework is general-purposed and customisable to support a variety of application domains and that it operates ubiquitously in environments without any preparation. One of the reasons that required a distinct development methodology was the lack of explicit requirements for the system. Initially, the only requirements that had been validated were those obtained through scientific publications. Furthermore, there were not any stakeholders to query for their preferences, nor were there any specific actors to consult them on how to improve their current behaviour. Consequently, our novel approach was based in the combination of two distinct system development methodologies; rapid-prototyping and the waterfall technique. Rapid-prototyping has been used for developing the overall system, whereas the waterfall approach has been followed during the development of specific system components. Another positive characteristic of our approach is that it required several interactions with potential end-users, a concept that contributed to the usability of our framework because we collected feedback on several instances and after implementing specific functionalities we could verify that served their purpose effectively. Initially, the Requirement Acquisition Survey collected feedback about the core user requirements for a mobile context-aware MR system. Following a development phase, the Preliminary Evaluation took place, which examined the first-cut VR prototype. After the Preliminary Evaluation, a second extensive development phase produced a full set of functionalities, which has been assessed in the Extensive Evaluation. It is worth noting that during the course of bringing the framework to life, several minor expert and empirical evaluations have been carried out.

This project has defined a novel framework architecture for ubiquitous, context-sensitive virtual environments. Although the architecture was intended to satisfy the information needs of mobile users across a wide range of application domains, it has been implemented in a distinct system architecture; an intuitive mobile urban navigation application. The novelty of the application is that it applies real-time contextual information on two distinct visualisation and interaction interfaces; VR and AR. Furthermore, the solution was developed with the minimum financial cost by integrating several components and by utilising mainstream hardware configurations. We consider the results of the research significant, as the project seeks to make a connection between the research fields involved and bring the applications based on the proposed framework closer and become more responsive to the user requirements. Finally, this customisation of the framework to a specific application may provide the necessary knowledge to other researchers, who wish to design and implement effective novel applications. The development of the proposed framework and one of its implementations was an ambitious venture especially during the time that this study commenced. The navigation application presents innovative features, by addressing the interdisciplinary challenges associated with the research context. The deployment of the framework architecture to a prototype application allowed us to evaluate the framework requirements, as well as to compare the differences between the applied interfaces. The difference between AR and VR for urban navigation has been examined in terms of user performance and in terms of usability. A selection of the most interesting results that have been produced after the Extensive Evaluation of the urban navigation application, are briefly presented at this point.

Effectiveness

- Users spent significantly less time to complete the wayfinding task while using the VR interface compared to AR;
- The *Minimum Speed* of the users was significantly higher in VR rather than in AR;
- The *Average Speed* of the users was significantly higher in VR rather than in AR;
- The VR interface offered more effective support for the wayfinding task in comparison to AR;
- The users can find their way in an unknown place more effectively when they use VR rather than AR;

- The VR interface and its features were more helpful in informing the user about the current task than those of AR.

Usability

- Even though most users performed faster with VR, a large proportion still prefers to use AR;
- Users preferred the allocentric oblique and allocentric plan views equally, whereas the egocentric perspective received fewer responses;
- An egocentric AR view is favoured by the users to an egocentric VR view;
- The allocentric oblique and allocentric plan perspectives are more valuable when a user interacts with the surrounding environment;
- The egocentric perspective is better when the user needs to interact with a single entity of the real world that is directly visible by the actor;
- For accomplishing the naïve search task, most users referenced the allocentric plan view;
- For the primed search, the allocentric oblique perspective was favoured when they had to locate and follow the route points;
- Most users found VR to be more satisfying than AR and they would like to find more applications or services that make use of it;
- The represented scene realism was found to be significantly higher in AR compared to VR;
- The level of joy that each interface provided to the users, is not significantly different.

Technical

- The accuracy of the positioning sensor was considered adequate for both interfaces in the navigation task;
- The accuracy of the underlying subsystem is adequate for representing the physical orientation of the user in both interfaces in real time;
- The AR interface responded significantly faster to the manual input of the user compared to the VR interface;
- The text style (i.e. colour and size) was easier to read in VR rather than in AR.

9.5 External Constraints & Further Work

Although we tried to be as exhaustive as possible while exploring the topics involved in the related research fields, there are some issues, which would be worth exploring further. These issues have been identified either by taking into consideration the advance of relevant technologies taking place during the course of the project, by investigating in more detail certain aspects of the study, or by coming across limitations of the selected approaches. In this section, the reader can find out about the limitations of this research, even though the majority has been mentioned in the relevant sections of the report. Furthermore, this section contains some suggestions on how to influence further research and development of the proposed framework.

When the project commenced, a decision needed to be made on the selection of the mobile platform that would accommodate the development efforts. At that time, the most advanced smartphone platform was Microsoft's Windows Mobile. Since then, several new platforms have evolved, which have provided similar or more advanced functionalities. This is reflected by the introduction of several new versions of Apple's iPhone and Google's Android, which are currently the most prevalent platforms. Apart from the platform specifications, the functionalities of the mobile devices have also improved. Currently, a large set of mobile devices embeds positioning sensors, orientation sensors, complex graphic pipelines, fast wireless networking and even dual-core processors, which reach and overcome the Gigahertz boundary. These new features would have been invaluable if they had been available at the beginning of this research because we needed to make several adjustments in order to accommodate issues that were partially implemented by the manufacturers. In more detail, the recently introduced sensors, which have been attached on the latest mobile devices, render the use of the system far more practical because there is no need for the user to hold more than one appliance at any time. Conversely, the most important drawback that we encountered while developing the framework was the lack of compatibility between the OpenGL ES API and Windows Mobile platform. This issue didn't allow employing OpenGL extensively in our system, which led into implementing the 3D interface on VRML. Furthermore, the AR interface was also affected, as it was not possible to render 3D elements on top of the streaming video. The accomplished research, in Chapters 5.4 and 5.5, took this issue under consideration and proposed a new way, by which the system functionality could be altered in order to utilise OpenGL ES for rendering 3D graphics. Thus, changing the 3D subsystem of the framework to support OpenGL ES will be the first step towards improving the existing solution.

Porting

Although the first version of the system has already been implemented on the Windows Mobile platform, it has become apparent that, in order to attract more interest and increase potential user adoption, it must be ported to additional platforms. *Aura* could cover a significant proportion of the smartphone market, therefore increasing the potential for generating revenue, by developing new versions for Google's and Apple's platforms. The next step of the plan is to develop a version for the iPhone platform, which is the most rapidly evolving trend in the mobile market. The market is considered big – in line with the size of the mobile services market. Upon preliminary contact with Alcatel-Lucent U.K., 3 U.K. and their customers, we have formed the distinct impression that there is a market for the type of product/service that can be derived from our research results and the prototype. A brief analysis, based on Gartner (Gartner, 2011), shows that currently the market for AR applications is \$2m, since the minority of smart phones are AR enabled. However the market is estimated to have rocketed up to \$712m by 2014. By 2012, it is estimated to have reached \$150-200m. The market will rise dramatically in the short term due to large-scale adoption of iPhone devices. AR will be one of the top 10 disruptive technologies between 2008 and 2012. Currently, iPhone has a 15.4% market share while Windows Mobile has 6.8%. Windows Mobile has dropped from 10.2% (2009), but is expected to rise again after the official release of the 7th OS version. Ideally, if our AR application could run on both iPhone and Windows Mobile, it could be targeting at least a quarter of the OS market share. The funding received on June 2010 by the *City University Research & Enterprise Unit* (CREU) has proven invaluable for commencing the porting process of the work, for analysing the market and for developing the commercial strategy of this project.

GPS

It was found that GPS is not always sufficient for localisation in adverse urban environments. Satellite data is often not available, or the quality is low, predominantly because of the height of surrounding buildings, which reflect the incoming signals. This is the reason, which calls for the development of a system that will employ additional absolute positioning techniques. The accuracy of the latest sensors, which have been embedded on new mobile devices, has been enhanced by the introduction of superior chipsets, which enable localisation of a device based not only on the availability of GPS satellites, but also on other technologies such as Cell ID and WLAN. Furthermore, the development of a software 3D filter (e.g. Kalman, Extended Kalman, Particle) can increase the positioning accuracy of measurements so that the application may reach higher-resolution positioning, resulting in the reduction of noise, which is introduced by the GPS. An initial study has been initiated towards the implementation of such a software filter, and it may prove particularly efficient by utilising the digital compass. Since other investigators have conducted similar research and development, this functionality has not been an integral part of this work.

Social Networking Connectivity

Currently, several LBS and social networking tools (e.g. Aka-Aki, Foursquare, Twitter) have been developed and have proved to operate efficiently in real-world conditions. Furthermore, several advancements (e.g. OAuth) have been proposed which enable straightforward connectivity with these solutions. It would be a great advantage of our framework if it would be customised in a way that it could use the services offered by these applications. As a result, further developments on certain aspects (e.g. forming an extensive list of users for networking) of the framework would not be required, and the focus would draw on developing further scenarios for collaboration. This could prove beneficial for all parties, since some companies are still looking to find innovative applications so that they can generate revenue from their consumers.

Server-side

One of the secondary goals of this project was to develop a mobile system by using minimal standardised configurations, available for commercial exploitation. For that reason, the development efforts were focused on the client side (i.e. mobile device). Some of the proposed operational scenarios of *Aura* would be extremely benefited by the introduction of a dedicated server entity. The server entity may offer further advanced functionalities, which could lead to new applications. Certain plans about the design and implementation of the server functionalities have been developed during the course of this project.

Database

A universally accessible server can incorporate a persistent storage engine. The user content that may be stored in the DB contains their preferences and track logs. Additionally, information about selected POIs will be recorded and become available for querying, as well as for being updated by the users of the system. It would be ideal if the database subsystem included spatial functionalities. Spatial Oracle, which is an extension of the famous DBMS, provides access to geospatial information and is considered to be one of the most complete and interoperable packages. Following next, ArcSDE by ESRI offers maximum support for GIS functionalities. The aforementioned DBMSs are expensive commercial solutions, which can significantly increase the development cost. MySQL and PostgreSQL are the preferred options for this stage of development, mainly because they are free of charge and because they support additional plug-ins, which offer spatial functionalities.

Another aspect of the system, which would be enhanced by the introduction of a server entity, is the reconstruction of the 3D models that describe the environment, which is surrounding the user. Chapter 5.4.1 described the process, which has been used to produce the 3D content that is loaded whenever the VR interface of *Aura* is selected. The disadvantage of the current process is that the 3D model must be downloaded manually to the device after it has been created. Automating this process and running it on the server side would allow truly ubiquitous 3D representation of the surrounding environment according to the user's location context. We envisage that a following version of *Aura* will allow the user to send the latest occupied coordinates to the server and in return to receive a 3D model that represents the environment up to a certain radius. This way, the need to manually load a different model to the VR interface, when the user reaches its boundaries, can be avoided. Although this method will not be able to produce textured 3D models, the complexity of geometrical 3D models can be automatically adjusted according to the user needs.

Evaluation

The Extensive Evaluation, which was presented in Chapter 7.4, provided invaluable insights into the effectiveness of the developed system and certain usability issues relevant to its operation. Although it was a very focused survey that provided valuable results for the research, the collected data can sustain further analysis, which may produce additional findings. Due to the fact that the conditions of field-based mobile usability evaluations are difficult to control in natural settings (Kjeldskov et al., 2005), we avoided stressing the data collection process and consequently the participants. Despite that, further analysis of the data may produce more detailed results about specific usability issues. For instance, we can find out which were the features that made *some* participants prefer the AR interface, despite the fact that they performed better when they used the VR alternative. Furthermore, we did not utilise exhaustive statistical techniques on the data, which may have produced further findings, because it was out of the project scope. It would be interesting to correlate between the participants' previous experience and the answers gathered by the subjective part of the survey. This method may prove useful in finding out how to increase the adoption of the system by the end-users. Chapter 1.7 offers a description of the publications, which are currently in preparation and intend to present further work derived from this research.

10 References

The core information sources that have been injected in the conceptual progress and the main body of this document are presented in this chapter. Their style conforms to the *Harvard* referencing system and they are sorted in alphabetical order.

AIKEN, B., STRASSNER, J., CARPENTER, B., FOSTER, I., LYNCH, C., MAMBRETTI, J., MOORE, R. & TEITELBAUM, B. (2000) Network Policy and Services: A Report of a Workshop on Middleware. IN SOCIETY, T. I. (Ed. *Request for Comments (RFC 2768).* Internet Engineering Task Force (IETF).

ANDERSON, H. R. (2003) *Fixed Broadband Wireless System Design*, Wiley & Sons.

APACHE (2011a) Jini/Apache River. Available from: http://www.sun.com/software/jini, http://www.jini.org, Accessed: 29 April 2011

APACHE (2011b) The Apache HTTP Server Project. Available from: http://httpd.apache.org, Accessed: 29 April 2011

APPLE (2011) iPhone. Available from: http://www.apple.com/iphone, http://developer.apple.com/devcenter/ios, Accessed: 5 May 2011

ARBANOWSKI, S., BALLON, P., DAVID, K., DROEGEHORN, O., EERTINK, H., WOLFGANG KELLERER, KRANENBURG, H. V., RAATIKAINEN, K. & POPESCU-ZELETIN, R. (2004) I-centric Communications: Personalization, Ambient Awareness, and Adaptability for future Mobile Services. *IEEE Communications Magazine.*

ARETZ, A. J. & WICKENS, C. D. (1992) The Mental Rotation of Map Displays. *Human Performance,* 5**,** 303-328.

ASHBROOK, D. & STARNER, T. (2003) Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing,* 7**,** pp. 275-286.

AUTODESK (2011) 3ds Max. Available from: http://usa.autodesk.com/3ds-max, Accessed: 29 April 2011

AZUMA, R., BAILLOT, Y., BEHRINGER, R., FEINER, S., JULIER, S. & MACINTYRE, B. (2001) Recent Advances in Augmented Reality. *IEEE Computer Graphics and Applications,* 21**,** 34-47.

AZUMA, R. T. (1995) Course Notes #9: Developing Advanced Virtual Reality Applications. *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95).* Los Angeles, CA, USA, Association for Computing Machinery (ACM).

AZUMA, R. T. (1997) A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments,* 6**,** 48.

BALDAUF, M., DUSTDAR, S. & ROSENBERG, F. (2007) A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing,* 2**,** 263-277.

BALKE, W.-T., KIESSLIN, W. & UNBEHEND, C. (2003) Personalized services for mobile route planning: a demonstration. *Proceedings of the 19th International Conference on Data Engineering (ICDE 2003).* Bangalore, India, IEEE Computer Society.

BANAVAR, G. & BERNSTEIN, A. (2002) Software infrastructure and design challenges for ubiquitous computing applications. *Communications of the ACM,* 45**,** 92-96.

BARBIR, A., PENNO, R., NETWORKS, N., CHEN, R., LABS, A. T., HOFMANN, M., TECHNOLOGIES, B. L. L., ORMAN, H. & DEVELOPMENT, P. S. (2004) An Architecture for Open Pluggable Edge Services (OPES). *Request For Comments (RFC3835).* Internet Engineering Task Force (IETF).

BECKER, C. & DURR, F. (2005) On location models for ubiquitous computing. *Journal of Personal and Ubiquitous Computing,* vol. 9**,** pp. 20-31.

BELL, J. (1999) *Doing Your Research Project: A Guide for First-time Researchers in Education and Social Science,* Maidenhead, UK, Open University Press.

BENFORD, S., CAPRA, M., FLINTHAM, M., CRABTREE, A., DROZD, A., OPPERMANN, L., PANKOKEBABATZ, U., WAERN, A., ERMI, L., JÄPPINEN, A. & MONTOLA, M. (2005) A review of design and evaluation methods for IPerG. IN BENFORD, S. & CAPRA, M. (Eds.) *WP5 Design and Evaluation.*

BENFORD, S., SEAGER, W., FLINTHAM, M., ANASTASI, R., ROWLAND, D., HUMBLE, J., STANTON, D., BOWERS, J., TANDAVANITJ, N., ADAMS, M., FARR, J. R., OLDROYD, A. & SUTTON, J. (2004) The Error of Our Ways: The Experience of Self-Reported Position in a Location-Based Game. *Proceedings of the 6th International Conference on Ubiquitous Computing (UbiComp).* Nottingham, UK, Springer Berlin / Heidelberg.

BERTINI, E., GABRIELLI, S. & KIMANI, S. (2006) Appropriating and assessing heuristics for mobile computing. *Proceedings of the working conference on Advanced Visual Interfaces (AVI 2006).* Venezia, Italy, ACM.

BEVAN, N. (2008) Classifying and selecting UX and usability measures. *Proceedings of Meaningful Measures: Valid Useful User Experience Measurement (VUUM), 5th COST294-MAUSE Open Workshop.* Reykjavik, Iceland.

BHAPKAR, V. P. (1966) A note on the equivalence of two test criteria for hypotheses in categorical data. *Journal of the American Statistical Association,* 61**,** 228-235.

BIMBER, O. & RASKAR, R. (2005) *Spatial Augmented Reality: Merging Real and Virtual Worlds,* Wellesley, Massachusetts, USA, A. K. Peters, Ltd.

BLAS, N. D., GOBBO, E. & PAOLINI, P. (2005) 3D Worlds and Cultural Heritage: Realism vs. Virtual Presence. IN TRANT, J. & BEARMAN, D. (Eds.) *Proceedings of Museums and the Web Conference (MW 2005).* Toronto, Archives & Museum Informatics.

BLOM, J. (2000) Personalization: a taxonomy. *Proceedings of the Conference on Human Factors in Computing Systems (CHI '00); extended abstracts on Human factors in computing systems.* The Hague, The Netherlands, ACM Press. New York, NY, USA.

BOYD DAVIS, S. (2007) A Word about the Weather: Depiction and Visualisation. *From Abstract Data Mapping to 3D Photorealism: understanding emerging intersections in visualisation practices and techniques - AHRC ICT Methods Network Expert Workshop.* Birmingham Institute of Art and Design, UK.

BOYD DAVIS, S. & CARINI, C. (2005) Constructing a Player-Centred Definition of Fun for Video Games Design. IN FINCHER, S., MARKOPOULOS, P., MOORE, D. & RUDDLE, R. (Eds.) *People and Computers XVIII — Design for Life, Proceedings of HCI 2004.* Leeds Metropolitan University, UK, Springer London.

BOYD DAVIS, S., MOAR, M., JACOBS, R., WATKINS, M. & CAPRA, M. (2007) Mapping Inside Out. IN MAGERKURTH, C. & RÖCKER, C. (Eds.) *Pervasive Gaming Applications: A reader for pervasive gaming research.* Aachen, Germany, Shaker Verlag.

BOYD DAVIS, S., MOAR, M., JACOBS, R., WATKINS, M., RIDDOCH, C. & COOKE, K. (2006) 'Ere Be Dragons: Heartfelt Gaming. *Digital Creativity,* 17**,** 157-162.

BROLL, W., OHLENBURG, J., LINDT, I., HERBST, I. & BRAUN, A.-K. (2006) Meeting technology challenges of pervasive augmented reality games. *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games.* Singapore, ACM.

BRUJIC-OKRETIC, V. (2003) A Potential Application of Augmented Reality in Mobile GIS Technologies. *Proceedings of the 11th Annual Conference in GIS Research UK (GISRUK).* London, U.K.

BRUMITT, B., MEYERS, B., ROBBINS, D., KRUMM, J., CZERWINSKI, M. & SHAFER, S. (1998) The New EasyLiving Project. Available from: http://research.microsoft.com/apps/pubs/default.aspx?id=68393, Accessed: 29 April 2011

BRUSILOVSKY, P. (1996) Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction,* 6**,** 87-129.

BURIGAT, S. & CHITTARO, L. (2005) Location-aware visualization of VRML models in GPS-based mobile guides. *Proceedings of the 10th international conference on 3D Web technology.* Bangor, United Kingdom, ACM Press.

BURNETT, G. E., SMITH, D. & MAY, A. J. (2001) Supporting the navigation task: Characteristics of 'good' landmarks. IN HANSON, M. A. (Ed.) *In Proceedings of the Annual Conference of the Ergonomics Society, Contemporary Ergonomics 2001.* Cirencester, UK, Taylor & Fransis, CRC Press.

BURRELL, J., GAY, G. K., KUBO, K. & FARINA, N. (2002) Context-Aware Computing: A Test Case. *Proceedings of the 4th international conference on Ubiquitous Computing.* Göteborg, Sweden, Springer-Verlag.

BUTCHART, B. (2011) Augmented Reality for Smartphones - A guide for developers and content publishers. *TechWatch Report.* 1.1 ed., JISC Observatory.

CAMINEO (2004) WebPark. Available from: http://www.webparkservices.info, Accessed: 2 May 2011

CAPRA, M., RADENKOVIC, M., BENFORD, S., OPPERMANN, L., DROZD, A. & FLINTHAM, M. (2005) The multimedia challenges raised by pervasive games. *Proceedings of the 13th annual ACM international conference on Multimedia.* Hilton, Singapore, ACM.

CARNEGIE MELLON, U. (2002) Project Aura. Available from: http://www.cs.cmu.edu/~aura, Accessed: 29 April 2011

CHANGING MINDS (2009) Types of Data. *Available from: http://changingminds.org/explanations/research/measurement/types_data.htm*.

CHEESMAN, J. & PERKINS, C. (2001) Virtual reality: an exploratory tool for investigating the cognitive mapping and navigational skills of the visually impaired. IN UNWIN, D. (Ed.) *Virtual reality in geography.* London, Taylor & Francis.

CHEN, F., YANG, C., YU, W., LE, X. & YANG, J. (2005) Research on mobile GIS based on LBS. *Proceedings of the Geoscience and Remote Sensing Symposium (IGARSS '05).* Chinese Academy of Science, Beijing, China, IEEE International.

CHEN, G. & KOTZ, D. (2000) A Survey of Context-Aware Mobile Computing Research. *Dartmouth Computer Science Technical Report TR2000-381.* Hanover, NH, USA, Dartmouth College.

CHEOK, A. D., TEH, K. S., NGUYEN, T. H. D., QUI, T. C. T., LEE, S. P., LIU, W., LI, C. C., DIAZ, D. & BOJ, C. (2006) Social and physical interactive paradigms for mixed-reality entertainment. *Computers in Entertainment (CIE),* 4**,** 20.

CHINCHOLLE, D., GOLDSTEIN, M., NYBERG, M. & ERIKSSON, M. (2002) Lost or Found? A Usability Evaluation of a Mobile Navigation and Location-Based Service. *Human Computer Interaction with Mobile Devices.* Springer Berlin / Heidelberg.

COHEN, J. W. (1988) *Statistical power analysis for the behavioral sciences,* Hillsdale, NJ, Lawrence Erlbaum Associates.

COHEN, J. W. (1992) A power primer. *Psychological Bulletin,* 112**,** 155-159.

COLUCCIA, E. & LOUSE, G. (2004) Gender differences in spatial orientation: A review. *Journal of Environmental Psychology,* 24**,** 329-340.

CONROY, R. (2001) Spatial Navigation in Immersive Virtual Environments. *Department of Architecture.* London, University of London.

CONSOLVO, S., SMITH, I. E., MATTHEWS, T., LAMARCA, A., TABERT, J. & POWLEDGE, P. (2005) Location Disclosure to Social Relations: Why, When & What people want to share. *Proceedings of the SIGCHI conference on Human factors in computing systems.* Portland, Oregon, USA, ACM Press, New York, NY, USA.

COOPER, I. & NOTTINGHAM, M. (2002) Web Intermediaries (WebI). Available from: *http://www.ietf.org/wg/concluded/webi, http://tools.ietf.org/wg/webi*, Accessed: 20 April 2011

COULTON, P., BAMFORD, W., CHEVERST, K. & RASHID, O. (2008) 3D Space-time visualization of player behaviour in pervasive location-based games. *International Journal of Computer Games Technology,* 2008**,** 1-5.

DARKEN, R. P. (1995) Wayfinding in Large-Scale Virtual Worlds. *Conference companion on Human Factors in Computing Systems (CHI).* Denver, Colorado, United States, ACM Press New York, NY, USA.

DARKEN, R. P. & PETERSON, B. (2001) Spatial Orientation, Wayfinding, and Representation. IN STANNEY, K. M. (Ed.) *Handbook of Virtual Environment Technology.* Mahway, NJ, USA, Lawrence Erlbaum Associates, Inc,.

DARKEN, R. P. & SIBERT, J. L. (1996a) Navigating large virtual spaces. *International Journal of Human-Computer Interaction,* 8**,** 20.

DARKEN, R. P. & SIBERT, J. L. (1996b) Wayfinding strategies and behaviors in large virtual worlds. *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground.* Vancouver, British Columbia, Canada, ACM Press.

DAY, M., CISCO, CAIN, B., STORIGEN, TOMLINSON, G., GROUP, T., RZEWSKI, P. & MEDIA PUBLISHER, I. (2003) A Model for Content Internetworking (CDI). *Request For Comments (RFC3466).* Internet Engineering Task Force (IETF).

DEPARTMENT OF CONSTITUTIONAL AFFAIRS, U. (1998) Data Protection Act. *Available from: http://www.legislation.gov.uk/ukpga/1998/29.*

DEPARTMENT OF CONSTITUTIONAL AFFAIRS, U. (2000) Freedom of Information Act.

DEY, A. K. (2000) Providing Architectural Support for Building Context-Aware Applications. *Computer Science.* Georgia, USA, Georgia Institute of Technology.

DEY, A. K., ABOWD, G. D., (PETER, J. B., DAVIES, N., SMITH, M. & STEGGLES), P. (1999) Towards a Better Understanding of Context and Context-Awareness. *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing.* Karlsruhe, Germany, Springer-Verlag, London, UK.

DIX, A., FINLAY, J., ABOWD, G. D. & BEALE, R. (2003) *Human Computer Interaction,* Harlow, Prentice Hall.

DOBSON, S. (2005) Leveraging the subtleties of location. *Proceedings of the 2005 Joint Conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies.* Grenoble, France, ACM Press, New York, NY, USA.

DRANSCH, D. (2005) Activity and Context - A Conceptual Framework for Mobile Geoservices. IN MENG, L., ZIPF, A. & REICHENBACHER, T. (Eds.) *Map-based Mobile Services: Theories, Methods and Implementations.* Berlin, Germany, Springer-Verlag.

DÜNSER, A., GRASSET, R. & BILLINGHURST, M. (2008) A survey of evaluation techniques used in augmented reality studies. *ACM SIGGRAPH Asia 2008 courses.* Singapore, ACM.

DURAND, F. (2002) An invitation to discuss computer depiction. *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering.* Annecy, France, ACM.

EMMERICH, W. (2000) *Engineering Distributed Objects,* Chisester, UK, John Wiley & Sons, Ltd.

ESRI (2011) ArcGIS. Available from: http://www.esri.com/software/arcgis, Accessed: 29 April 2011

ETSI, E. T. S. I. (2011) Parlay/Open Service Access (OSA). Available from: http://etsi.org/WebSite/Technologies/OSA.aspx, Accessed: 29 April 2011

EU, P. A. C. (1995) Data Protection Directive. *Available from: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML.*

EYLES, M. & EGLIN, R. (2007) Entering an age of playfulness where persistent, pervasive ambient games create moods and modify behaviour. *Third International Conference on Games Research and Development 2007 (CyberGames 2007).* Manchester Metropolitan University, U.K.

FALLMAN, D. (2003) Design-oriented Human-Computer Interaction. *Proceedings of the SIGCHI conference on Human factors in computing systems.* Ft. Lauderdale, Florida, USA, ACM Press.

FEINER, S. (2003) Taking It to the Streets: How Virtual Reality Can Change Mobile Computing. *Proceedings of the IEEE Virtual Reality 2003.* Los Angeles, CA, USA, IEEE Computer Society.

FEINER, S., MACINTYRE, B., HOLLERER, T. & WEBSTER, A. (1997) A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment *Proceedings of the First International Symposium on Wearable Computers (ISWC '97).* Cambridge, MA, USA, IEEE Computer Society.

FIELD, A. P. (2005) *Discovering statistics using SPSS: and sex and drugs and rock 'n' roll,* London, Sage.

FIPA, F. F. I. P. A. (2007) Agent UML. Available from: http://www.auml.org, Accessed: 29 April 2011

FISHER, P. & UNWIN, D. (Eds.) (2002) *Virtual Reality in Geography,* London, Taylor & Francis.

FISHER, R. A. (1925) Tests of significance of means, difference of means, and reression coefficients. *Statistical Methods for Research Workers.* 3 ed. Edinburgh, Oliver and Boyd.

FRASER, M., GLOVER, T., VAGHI, I., BENFORD, S., GREENHALGH, C., HINDMARSH, J. & HEATH, C. (2000) Revealing the realities of collaborative virtual reality. *Proceedings of the 3rd International Conference on Collaborative Virtual Environments.* San Francisco, California, United States, ACM.

FREKON, E. & NOU, A. A. (1998) Building distributed virtual environments to support collaborative work. *Proceedings of the ACM symposium on Virtual reality software and technology.* Taipei, Taiwan, ACM Press. New York, NY, USA.

FUENTE, C. D. L., RAPER, J., STANLEY, M., NORRIS, P. & DONNELL, M. O. (2005) Location-based services: Understanding the Japanese experience. *Globalwatch Mission Report.* London, Department of Trade and Industry (DTI).

GABBARD, J. L., II, J. E. S., HIX, D., KIM, S.-J. & FITCH, G. (2007) Active Text Drawing Styles for Outdoor Augmented Reality: A User-Based Study and Design Implications. *Proceedings of the Virtual Reality Conference (VR'07).* Charlotte, NC, USA.

GABBARD, J. L., J. EDWARD SWAN, I., HIX, D., LUCAS, J. & GUPTA, D. (2005) An Empirical User-based Study of Text Drawing Styles and Outdoor Background Textures for Augmented Reality. *Proceedings of the IEEE Conference on Virtual Reality (VR'05).* Bonn, Germany, IEEE Computer Society.

GAHEGAN, M. (2003) Is inductive machine learning just another wild goose (or might it lay the golden egg)? *International Journal of Geographical Information Science,* 17**,** pp. 69-92.

GAMMA, E., HELM, R., JOHNSON, R. & VLISSIDES, J. M. (1994) *Design Patterns: Elements of Reusable Object-Oriented Software,* Indianapolis, USA, Addison-Wesley.

GARTNER, I. (2011) Gartner Research. Available from: http://www.gartner.com, Accessed: 29 April 2011

GATZIDIS, C., PAPAKONSTANTINOU, S., BRUJIC-OKRETIC, V. & BAKER, S. (2008) Recent Advances in the User Evaluation Methods and Studies of Non-Photorealistic Visualization and Rendering Techniques. *Proceedings of the 12th International Conference Information Visualisation.* London, UK, IEEE Computer Society.

GEOINFORMATION GROUP, T. (2010) The GeoInformation Group Homepage. Available from: http://www.geoinformationgroup.co.uk, Accessed: 4 May 2011

GEORGIA INSTITUTE OF TECHNOLOGY (2011) The Aware Home. Available from: http://awarehome.imtc.gatech.edu, Accessed: 29 April 2011

GÖKER, A. & MYRHAUG, H. I. (2002) User Context and Personalisation. *In ECCBR Workshop on Case Based Reasoning and Personalisation.* Aberdeen, UK.

GÖKER, A., MYRHAUG, H. I. & BIERIG, R. (2009) Context and Information Retrieval. IN GÖKER, A. & DAVIES, J. (Eds.) *Information Retrieval: Searching in the 21st Century.* 1st ed. London, UK, John Wiley & Sons, Ltd.

GOLLEDGE, R. (1995) Path selection and route preference in human navigation: A progress report. *Lecture Notes in Computer Science (Spatial Information Theory A Theoretical Basis for GIS),* 988**,** 207-222.

GOLLEDGE, R. G. (1999) Human Wayfinding and Cognitive Maps. IN GOLLEDGE, R. G. (Ed.) *Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes.* 1st ed. Baltimore, USA, The John Hopkins University Press.

GOOGLE (2011a) Android. Available from: http://www.android.com, http://developer.android.com, Accessed: 5 May 2011 2011

GOOGLE (2011b) Google Earth. Available from: http://earth.google.com, Accessed: 5 May 2011 2011

GRAUMANN, D., LARA, W., HIGHTOWER, J. & BORRIELLO, G. (2003) Real-world implementation of the location stack: the universal location framework. *Proceedings of the Fifth IEEE Workshop on Mobile Computing Systems and Applications.* Monterey, CA, USA, IEEE Computer Society.

GRIMM, R., DAVIS, J., LEMAR, E., MACBETH, A., SWANSON, S., ANDERSON, T., BERSHAD, B., BORRIELLO, G., GRIBBLE, S. & WETHERALL, D. (2004) System support for pervasive applications. *ACM Transactions on Computer Systems (TOCS),* 22**,** pp. 421-486.

GRISWOLD, W. G., SHANAHAN, P., BROWN, S. W., BOYER, R., RATTO, M., SHAPIRO, R. B. & TRUONG, T. M. (2004) ActiveCampus: Experiments in Community-Oriented Ubiquitous Computing. *IEEE Computer,* 37**,** pp. 73-81.

GWIZDKA, J. (2000) What's in the Context? *Proceedings of the CHI 2000 Workshop on "The What, Who, Where, When, Why and How of Context-Awareness".* The Hague, The Netherlands, Georgia Institute of Technology.

HAINES, E. (1994) Point in polygon strategies. IN HECKBERT, P. (Ed.) *Graphics gems IV.* San Diego, USA, Academic Press Professional, Inc.

HALL, E. L., TIO, J. B. K., MCPHERSON, C. A. & SADJADI, F. A. (1982) Measuring Curved Surfaces for Robot Vision. *Computer.* IEEE Computer Society Press.

HAMMER-LAHAV, E. (2010) The OAuth 1.0 Protocol. *Request For Comments (RFC5849).* Internet Engineering Task Force (IETF).

HART, R. A. & MOORE, G. T. (1973) The development of spatial cognition: A review. IN DOWNS, R. M. & STEA, D. (Eds.) *Image and Environment: Cognitive Mapping and Spatial Behavior.* Chicago, Aldine Publishing.

HART, S. G. (2006) Nasa-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting,* 50**,** 904-908.

HART, S. G., STAVELAND, L. E., PETER, A. H. & NAJMEDIN, M. (1988) Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology.* North-Holland.

HASSENZAHL, M. (2003) The Thing and I: Understanding the Relationship Between User and Product. IN BLYTHE, M., OVERBEEKE, K., MONK, A. & WRIGHT, P. (Eds.) *Funology: From Usability to Enjoy-ment.* Norwell, MA, USA, Kluwer Academic Publishers.

HASSENZAHL, M. & TRACTINSKY, N. (2006) User experience: A research agenda. *Behaviour & Information Technology,* 25**,** 91-97.

HEGARTY, M., RICHARDSON, A. E., MONTELLO, D. R., LOVELACE, K. & SUBBIAH, I. (2002) Development of a self-report measure of environmental spatial ability. *Intelligence,* 30**,** 425-447

HENRICKSEN, K. & INDULSKA, J. (2004) A Software Engineering Framework for Context-Aware Pervasive Computing. *2nd IEEE International Conference on Pervasive Computing and Communications (PerCom'04).* Orlando, Florida.

HENRICKSEN, K., INDULSKA, J. & RAKOTONIRAINY, A. (2001) Infrastructure for Pervasive Computing: Challenges. IN BAUKNECHT, K., BRAUER, W. & MUCK, T. A. (Eds.) *Workshop on Pervasive Computing and Information Logistics at Informatik 2001.* Universitat Wien, Vienna.

HIGHTOWER, J. & BORRIELLO, G. (2001) Location Systems for Ubiquitous Computing. *IEEE Computer.* IEEE Computer Society Press, Los Alamitos, CA, USA.

HIGHTOWER, J., BRUMITT, B. & BORRIELLO, G. (2002) The Location Stack: A Layered Model for Location in Ubiquitous Computing. *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA).* Catskills, New York, USA, IEEE Computer Society, Washington, DC, USA.

HINSKE, S., LAMPE, M., MAGERKURTH, C. & ROECKER, C. (2007) Classifying Pervasive Games: On Pervasive Computing and Mixed Reality. IN MAGERKURTH, C. & ROECKER, C. (Eds.) *Concepts and technologies for Pervasive Games - A Reader for Pervasive Gaming Research.* Aachen, Germany, Shaker Verlag.

HIRTLE, S. C. (2003) Neighbourhoods and Landmarks. IN DUCKHAM, M., GOODCHILD, M. F. & WORBOYS, M. (Eds.) *Foundations of Geographic Information Science.* London, Taylor & Francis.

HITLAB, H. I. T. L. (2007) ARToolKit. WASHINGTON, U. O., Available from: www.hitl.washington.edu/artoolkit, Accessed: 4 May 2011

HIX, D., GABBARD, J. L., II, J. E. S., LIVINGSTON, M. A., TOBIAS H. HÖLLERER, JULIER, S. J., YOHA, B. & BROWN, D. (2004) A Cost-Effective Usability Evaluation Progression for Novel Interactive Systems. *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 9.* Hawaii, IEEE Computer Society.

HÖLL, D., LEPLOW, B., SCHÖNFELD, R. & MEHDORN, M. (2003) Is It Possible to Learn and Transfer Spatial Information from Virtual to Real Worlds? IN FREKSA, C., BRAUER, W., HABEL, C. & WENDER, K. F. (Eds.) *Spatial Cognition III - Routes and Navigation, Human Memory and Learning, Spatial Representation and Spatial Learning.* Berlin, Springer.

HOME OFFICE, U. (2000) Regulation of Investigatory Powers Act.

HONEYWELL (2003) Honeywell Microelectronics & Precision Sensors. Available from: http://www.ssec.honeywell.com, Accessed: 29 April 2011

HOUDE, S. & HILL, C. (1997) What do prototypes prototype? IN HELANDER, M. G., LANDAUER, T. K. & PRABHU, P. V. (Eds.) *In Handbook of Human-Computer Interaction (2nd Ed.).* Amsterdam, Elsevier Science & Technology Books.

HOWARD, J. H. & KERST, S. M. (1981) Memory and Perception of Cartographic Information for Familiar and Unfamiliar Environments. *Human Factors: The Journal of the Human Factors and Ergonomics Society,* 23**,** 495-503.

HP, H.-P. (2001) The Cooltown User Experience. Available from: http://www.hpl.hp.com/techreports/2001/HPL-2001-22.html, Accessed: 29 April 2011

IACHELLO, G., SMITH, I., CONSOLVO, S., CHEN, M. & ABOWD, G. D. (2005) Developing privacy guidelines for social location disclosure applications and services. *Proceedings of the Symposium on Usable Privacy and Security.* Pittsburgh, Pennsylvania, USA, ACM Press, New York, NY, USA.

IBM (2007) WebSphere Software. Available from: http://www-306.ibm.com/software/pervasive, Accessed: 29 April 2011

IBM (2010) SPSS. Available from: http://www.spss.com, Accessed: 29 April 2011

IBM (2011) IBM DB2. Available from: www.ibm.com/software/data/db2, Accessed: 29 April 2011

IEEE, C. S. (2011) The Foundation for Intelligent Physical Agents (FIPA). Available from: http://www.fipa.org, Accessed: 29 April 2011

IKV++ (1999) Grasshopper - A Universal Agent Platform Based on OMG MASIF and FIPA Standards. Available from: http://cordis.europa.eu/infowin/acts/analysys/products/thematic/agents/ch4/ch4.htm, Accessed: 29 April 2011

IMAGINATION TECHNOLOGIES (2011) POWERVR Insider SDK. Available from: http://www.imgtec.com/powervr/insider/powervr-sdk.asp, Accessed: 29 April 2011

INTERNATIONAL SOCIETY FOR PRESENCE RESEARCH (ISPR) (2000) The Concept of Presence: Explication Statement. Available from: http://ispr.info/about-presence-2/about-presence/, Accessed: 15 December 2011

ISO 9241-11 (1998) Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on Usability. *TC 159 Ergonomics, SC4.* International Organisation for Standardisation.

ISO 9241-210 (2010) Ergonomics of human-system interaction - Part 210: Human-centred design process for interactive systems. *TC 159 Ergonomics, SC4.* International Organisation for Standardisation.

ISO/IEC 25010 (2011) Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Systems and software quality models. *JTC 1 Information technology, SC7.* International Organisation for Standardisation.

ITU (2011) International Telecommunication Union. Available from: http://www.itu.int/home/index.html, Accessed: 12 March 2007

JONES, S. & MAIDEN, N. (2005) RESCUE: An Integrated Method for Specifying Requirements for Complex Socio-Technical Systems. IN MATE, J. L. & SILVA, A. (Eds.) *Requirements Engineering for Sociotechnical Systems.* Information Science Publishing, Idea Group Inc.

JONES, S., MAIDEN, N. & MANNING, S., GREENWOOD, J (2004) Human Activity Modelling in the Specification of Operational Requirements: Work in Progress. *Proceedings of the Workshop W1L: Bridging the Gaps II between Software Engineering and Human-Computer Interaction.* Edinburgh, Scotland, 26th International Conference on Software Engineering (ICSE2004).

JØRSTAD, I., THANH, D. V. & DUSTDAR, S. (2004) Personalisation of Future Mobile Services. *9th International Conference on Intelligence in service delivery Networks.* Bordeaux, France.

KAENAMPORNPAN, M. & O'NEILL, E. (2004) An Intergrated Context Model: Bringing Activity to Context. IN INDULSKA, J. & ROURE, D. D. (Eds.) *1st International Workshop on Advanced Context Modelling, Reasoning and Management, The 6th International Conference on Ubiquitous Computing (UbiComp 2004).* Nottingham, UK.

KATO, H. & BILLINGHURST, M. (1999) Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality.* San Francisco, CA, USA, IEEE Computer Society.

KHANNA, P., YU, I., MORTENSEN, J. & SLATER, M. (2006) Presence in response to dynamic visual realism: a preliminary report of an experiment study. *Proceedings of the ACM symposium on Virtual reality software and technology.* Limassol, Cyprus, ACM.

KHRONOS, G. (2011) OpenGL ES - The Standard for Embedded Accelerated 3D Graphics. Available from: http://www.khronos.org/opengles, Accessed: 29 April 2011

KJELDSKOV, J., GRAHAM, C., PEDELL, S., VETERE, F., HOWARD, S., BALBO, S. & DAVIES, J. (2005) Evaluating the usability of a mobile guide: the influence of location, participants and resources. *Behaviour & Information Technology,* 24**,** 15.

KLATZKY, R. L. (1998) Allocentric and egocentric spatial representations: Definitions, distinctions, and interconnections. IN FREKSA, C., HABEL, C. & WENDER, K. F. (Eds.) *Spatial Cognition, An Interdisciplinary Approach to Representing and Processing Spatial Knowledge.* London, UK, Springer-Verlag.

KOBSA, A. (2001) Generic User Modeling Systems. *User Modeling and User-Adapted Interaction,* 11**,** 49-63.

KOBSA, A., KOENEMANN, J. & POHL, W. (2001) Personalised hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review,* 16**,** 111-155.

KODA, T., NAKAZAWA, S. & ISHIDA, T. (2005) Talking Digital Cities: Connecting Heterogeneous Digital Cities Via the Universal Mobile Interface. IN KOIZUMI, P. V. D. B. S. (Ed.) *Digital Cities III - Information Technologies for Social Capital: Cross-cultural Perspectives.* Amsterdam, The Netherlands, Springer Berlin, Heidelberg, New York.

KORKEA-AHO, M. (2000) Context-Aware Applications Survey. *Internetworking Seminar.* Helsinki, Helsinki University of Technology.

KORPIPAA, P., MANTYJARVI, J., KELA, J., KERANEN, H. & MALM, E.-J. (2003) Managing Context Information in Mobile Devices. *IEEE Pervasive Computing,* 2**,** 42-51.

KORTUEM, G., SEGALL, Z. & THOMPSON, T. G. C. (1999) Close Encounters: Supporting Mobile Collaboration through Interchange of User Profiles. IN GOOS, G., HARTMANIS, J. & LEEUWEN, J. V. (Eds.) *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing.* Karlsruhe, Germany, Springer-Verlag, London, UK.

KRAY, C., BAUS, J., ZIMMER, H., SPEISER, H. & KRÜGER, A. (2001) Two Path Preposition: Along and Past. IN MONTELLO, D. R. (Ed.) *Spatial Information Theory (Proceedings of COSIT'01).* Berlin Germany, Springer.

KRAY, C., ELTING, C., LAAKSO, K. & COORS, V. (2003) Presenting route instructions on mobile devices. *Proceedings of the 8th international conference on Intelligent user interfaces.* Miami, Florida, USA, ACM.

KULJU, M. & KAASINEN, E. (2002) Route Guidance Using a 3D City Model on a Mobile Device. *Workshop on Mobile Tourism Support, Mobile HCI 2002 Symposium.* Pisa, Italy, Springer Berlin / Heidelberg.

LAAKSO, K., GJESDAL, O. & SULEBAK, J. R. (2003) Tourist information and navigation support by using 3D maps displayed on mobile devices. *Workshop in HCI in mobile Guides. Mobile HCI 2003 Symposium.* Udine, Italy.

LANKHORST, M. M., VAN KRANENBURG, H., SALDEN, A. & PEDDEMORS, A. J. H. (2002) Enabling technology for personalizing mobile services. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS 02) - Volume 3.* Hawaii, IEEE.

LAW, E. L.-C., ROTO, V., HASSENZAHL, M., VERMEEREN, A. P. O. S. & KORT, J. (2009) Understanding, scoping and defining user experience: a survey approach. *Proceedings of the 27th International Conference on Human Factors in Computing Systems.* Boston, MA, USA, ACM.

LAWTON, C. A. (1994) Gender differences in way-finding strategies: Relationship to spatial ability and spatial anxiety. *Sex Roles,* 30**,** 765-779.

LAWTON, C. A. & KALLAI, J. (2002) Gender Differences in Wayfinding Strategies and Anxiety About Wayfinding: A Cross-Cultural Comparison. *Sex Roles,* 47**,** 389-401.

LAWTON, G. (2006) Making Virtual Reality More Accessible. *Computer.* Los Alamitos, CA, USA, IEEE Computer Society.

LESZCZYNA, R. (2004) Evaluation of agent platforms. IN EUROPEAN COMMISSION, J. R. C. (Ed. *Architecture supporting security of agent systems.* Ispra, Italy, Institute for the Protection and security of the Citizen.

LEVENE, H. (1960) Robust tests for equality of variances. IN INGRAM OLKIN, H. H., ET AL (Ed.) *Contributions to probability and statistics: Essays in honor of Harold Hotelling.* Stanford University Press.

LEYKIN, A. & TUCERYAN, M. (2004) Determining text readability over textured backgrounds in augmented reality systems. *Proceedings of the ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry.* Singapore, ACM.

LIAROKAPIS, F. (2007) An augmented reality interface for visualizing and interacting with virtual content. *Virtual Reality,* 11**,** 23-43.

LIAROKAPIS, F., BRUJIC-OKRETIC, V. & PAPAKONSTANTINOU, S. (2006a) Exploring Urban Environments Using Virtual and Augmented Reality. *Journal of Virtual Reality and Broadcasting (JVRB 2006),* vol. 3**,** p. 1-13.

LIAROKAPIS, F., MOUNTAIN, D., PAPAKONSTANTINOU, S., BRUJIC-OKRETIC, V. & RAPER, J. (2006b) Mixed Reality For Exploring Urban Environments. *Proceedings of the 1st International Conference on Computer Graphics Theory and Applications (GRAPP 2006).* Setubal, Portugal, Insticc Press.

LIAROKAPIS, F., RAPER, J. & BRUJIC-OKRETIC, V. (2006c) Navigating within the urban environment using Location and Orientation-based Services. *European Navigation Conference 2006.* Manchester, UK.

LIKERT, R. (1932) A technique for the measurement of attitudes. *Archives of Psychology,* 22**,** 1-55.

LINDEN, R. (2011) SecondLife. Available from: http://secondlife.com/, Accessed: 29 April 2011

LINDLEY, C. A. (2004) Trans-reality gaming. *Proceedings of the 2nd Annual International Workshop in Computer Game Design and Technology.* John Moores University, Liverpool, UK.

LOMBARD, M. & DITTON, T. (1997) At the Heart of It All: The Concept of Presence. *Journal of Computer-Mediated Communication,* 3.

MACEACHREN, A. M. (1998) Visualization - Cartography for the 21st century. *Proceedings of the Polish Spatial Information Association conference.* Warsaw, Poland.

MACEACHREN, A. M., EDSALL, R., HAUG, D., BAXTER, R., OTTO, G., MASTERS, R., FUHRMANN, S. & QIAN, L. (1999) Virtual Environments for Geographic Visualization: Potential and Challenges. *Proceedings of the ACM Workshop on New Paradigms for Information Visualization and Manipulation in conjunction with the 8th ACM International conference on Information and Knowledge Management.* Kansas City, Missouri, United States, ACM Press -New York, NY, USA.

MACEACHREN, A. M. & KRAAK, M.-J. (2001) Research Challenges in Geovisualization. *Cartography and Geographic Information Systems. Special issue on Geovisualization,* 28**,** 3-12.

MACGREGOR, S. (2003) Customer Service or Invasion of Privacy. *The Journal of the Communication Networks,* 2**,** pp. 40-47.

MAGERKURTH, C., CHEOK, A. D., MANDRYK, R. L. & NILSEN, T. (2005) Pervasive Games: Bringing computer entertainment back to the real world. *ACM Computers in Entertainment (CIE),* 3**,** 19.

MAGERKURTH, C., STENZEL, R., STREITZ, N. & NEUHOLD, E. (2003) A Multimodal Interaction Framework for Pervasive Game Applications. *Workshop at Artificial Intelligence in Mobile System 2003 (AIMS 2003).* Seattle, USA.

MAIDEN, N., JONES, S. & FLYNN, M. (2003) Innovative Requirements Engineering Applied to ATM. *5th USA/Europe Air Traffic Management Research and Development Seminar.* Budapest, Hungary.

MAIDEN, N., NCUBE, C. & ROBERTSON, S. (2007) Can Requirements Be Creative? Experiences with an Enhanced Air Space Management System. *Proceedings of the 29th international conference on Software Engineering.* IEEE Computer Society.

MALINOWSKI, J. C. & GILLESPIE, W. T. (2001) Individual differences in performance on a large-scale, real-world wayfinding task. *Journal of Environmental Psychology,* 21**,** 73-82.

MAXWELL, A. E. (1970) Comparing the classification of subjects by two independent judges. *British Journal of Psychiatry***,** 651-655.

MCCARTY, B. & CASSADY-DORION, L. (1998) *Java Distributed Objects,* Indianapolis, IN, USA, Sams.

MCNEMAR, Q. (1947) Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika,* 12**,** 153–157.

MENG, L. (2005) Ego centres of mobile users and egocentric map design. IN MENG, L., ZIPF, A. & REICHENBACHER, T. (Eds.) *Map-based Mobile Services: Theories, Methods and Implementations.* Berlin, Germany, Springer-Verlag.

MENG, L. & REICHENBACHER, T. (2005) Map-based mobile services. IN MENG, L., ZIPF, A. & REICHENBACHER, T. (Eds.) *Map-based Mobile Services: Theories, Methods and Implementations.* Berlin, Germany, Springer-Verlag.

MICHON, P.-E. & DENIS, M. (2001) When and Why Are Visual Landmarks Used in Giving Directions? IN MONTELLO, D. R. (Ed.) *Spatial Information Theory (Proceedings of COSIT'01).* Berlin, Springer.

MICROSOFT (2010) Windows Mobile 6.5 Video Capture (MSDN), M. D. N., Available from: *http://msdn.microsoft.com/en-us/library/aa924811.aspx*, Accessed: 5 May 2011

MICROSOFT (2011) Windows Live ID. Available from: *https://accountservices.passport.net/ppnetworkhome.srf*, Accessed: 29 April 2011

MILGRAM, P. & KISHINO, F. (1994) A Taxonomy of Mixed Reality Visual Displays. *IEICE TRANSACTIONS on Information and Systems,* E77-D**,** 15.

MILGRAM, P., TAKEMURA, H., UTSUMI, A. & KISHINO, F. (1994) Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum. *Proceedings of SPIE: Telemanipulator and Telepresence Technologies.* Boston, MA, USA, SPIE Vol. 2351.

MILLER, C. & ALLEN, G. (2001) Spatial Frames of Reference Used in Identifying Direction of Movement: An Unexpected Turn. IN MONTELLO, D. R. (Ed.) *Spatial Information Theory (Proceedings of COSIT'01).* Berlin, Springer.

MIT, U. (2004) Project Oxygen - Pervasive Human-Centered Computing. Available from: *http://oxygen.csail.mit.edu/Overview.html*, Accessed: 29 April 2011

MITSEVA, A., IMINE, M. & PRASAD, N. R. (2006) Context-aware privacy protection with profile management. *Proceedings of the 4th international workshop on Wireless mobile applications and services on WLAN hotspots of the International Conference on Mobile Computing and Networking.* Los Angeles, CA, USA, ACM Press New York, NY, USA.

MONTELLO, D. R. (2005) Navigation. IN SHAH, P. & MIYAKE, A. (Eds.) *The Cambridge Handbook of Visuospatial Thinking.* Cambridge University Press.

MONTOLA, M. (2005) Exploring the Edge of the Magic Circle. Defining Pervasive Games. *Proceedings of Digital Experience: Design, Aesthetics, Practice Conference (DAC).* IT University of Copenhagen.

MORRIS, T. (2004) *Computer Vision and Image Processing*, Palgrave MacMillan.

MOSTEFAOUI, G. K., PASQUIER-ROCHA, J. & BREZILLON, P. (2004) Context-Aware Computing: A Guide for the Pervasive Computing Community. *Proceedings of the The IEEE/ACS International Conference on Pervasive Services (ICPS'04) - Volume 00.* American University of Beirut (AUB), Lebanon, IEEE Computer Society.

MOU, W., MCNAMARA, T. P., RUMP, B. R. & XIAO, C. (2006) Roles of Egocentric and Allocentric Spatial Representations in Locomotion and Reorientation. *Journal of Experimental Psychology: Learning, Memory, and Cognition,* 32**,** 1274-1290.

MOUNTAIN, D. & LIAROKAPIS, F. (2007) Mixed reality (MR) interfaces for mobile information systems. *Aslib Proceedings, Special issue: UK library & information schools,* 59**,** 422-436.

MOUNTAIN, D. & MACFARLANE, A. (2007) Geographic information retrieval in a mobile environment: Evaluating the needs of mobile individuals. *Journal of Information Science,* 33**,** 515-530.

MYRHAUG, H., WHITEHEAD, N., GÖKER, A., FAEGRI, T., LECH, T., MARKOPOULOS, P., EGGEN, B., AARTS, E. & CROWLEY, J. (2004) AmbieSense - A System and Reference Architecture for Personalised Context-Sensitive Information Services for Mobile Users. *Ambient Intelligence,* 3295**,** 327-338.

MYRHAUG, H. I. & GÖKER, A. (2003) AmbieSense - Interactive information channels in the surroundings of the mobile user. *2nd International Conference on Universal Access in Human-Computer Interaction (HCI International 2003).* Crete, Greece, Lawrence Erlbaum Associates.

NGUYEN, G., DANG, T. T., HLUCHY, L., LACLAVIK, M., BALOGH, Z. & BUDINSKA, I. (2002) Agent Platform Evaluation and Comparison. IN PELLUCID, E. C. R. D. P. (Ed. Bratislava, Slovakia, Institute of Informatics, Slovak Academy of Sciences.

NIELSEN, J. (1992) Finding usability problems through heuristic evaluation. *Proceedings of the SIGCHI conference on Human factors in computing systems.* Monterey, California, United States, ACM.

NIELSEN, J. (1994a) Enhancing the explanatory power of usability heuristics. *Proceedings of the SIGCHI conference on Human factors in computing systems: Celebrating interdependence.* Boston, Massachusetts, USA, ACM.

NIELSEN, J. (1994b) Guerrilla HCI: Using discount usability engineering to penetrate the intimidation barrier. IN BIAS, R. G. & MAYHEW, D. J. (Eds.) *Cost-Justifying Usability.* Academic Press.

NIELSEN, J. (1994c) *Usability Engineering,* San Francisco, USA, Morgan Kaufmann.

NIEUWDORP, E. (2007) The Pervasive Discourse: An Analysis. *Computers in Entertainment (CIE),* 5**,** 13.

NIVALA, A.-M. & SARJAKOSKI, L. T. (2004) Preventing Interruptions in Mobile Map Reading Process by Personalisation. IN BREWSTER, S. & DUNLOP, M. (Eds.) *Proceedings of the 6th International Symposium on Mobile Human-Computer Interaction (MobileHCI 2004).* University of Strathclyde, Glasgow, Scotland, Springer Berlin / Heidelberg.

NORMAN, D. A. (2002) *The Design of Everyday Things*, Basic Books.

O'LAUGHLIN, E. M. & BRUBAKER, B. S. (1998) Use of landmarks in cognitive mapping: Gender differences in self report versus performance. *Personality and Individual Differences,* 24**,** 595-601.

OHMI, M. (2000) Effects of viewing angle on performance of wayfinding and cognitive-map acquisition. *Proceedings of the 10th International Conference on Artificial Reality and Tele-existence.* National Taiwan University, Taipei, Taiwan.

ONTOPRISE (2011) OntoBroker. Available from: http://www.ontoprise.de/en/products/ontobroker, Accessed: 29 April 2011

ORACLE (2011) Java 2 Platform, Enterprise Edition (J2EE). Available from: http://www.oracle.com/technetwork/java/javaee/overview/index.html, Accessed: 29 April 2011

ORDNANCE SURVEY (2011) Ordnance Survey Homepage: www.ordnancesurvey.co.uk. Available from: www.ordnancesurvey.co.uk, Accessed: 29 April 2011

OULASVIRTA, A., ESTLANDER, S. & NURMINEN, A. (2009) Embodied interaction with a 3D versus 2D mobile map. *Personal Ubiquitous Comput.,* 13**,** 303-320.

PALLANT, J. (2007) *SPSS Survival Manual: A Step by Step Guide to Data Analysis Using SPSS for Windows (Version 15),* Maidenhead, UK, McGraw-Hill/Open University Press.

PAPAGIANNAKIS, G., SINGH, G. & MAGNENAT-THALMANN, N. (2008) A survey of mobile and wireless technologies for augmented reality systems. *Computer Animation and Virtual Worlds,* 19**,** 3-22.

PAPAKONSTANTINOU, S. (2005) Mobile Virtual Reality Interaction with Global Positioning Systems. *Dept. of Information Science.* London, CITY University.

PAPAKONSTANTINOU, S. & BHATIA, M. (2009) Aura - An innovative mobile advertising platform. *Global Idea to Product (i2PG) Summit.* Austin, Texas, USA.

PAPAKONSTANTINOU, S. & BRUJIC-OKRETIC, V. (2006) Design and implementation of an advanced generic location-aware engine. *Proceedings of the 14th Annual Conference in GIS Research UK (GISRUK).* Nottingham, U.K., The University of Nottingham.

PAPAKONSTANTINOU, S. & BRUJIC-OKRETIC, V. (2009a) Framework for Context-Aware Smartphone Applications. *The Visual Computer, International Journal of Computer Graphics,* 25**,** 11.

PAPAKONSTANTINOU, S. & BRUJIC-OKRETIC, V. (2009b) Prototyping a Context-Aware Framework for Pervasive Entertainment Applications. *Proceedings of the 1st International IEEE Conference in Serious Games and Virtual Worlds (VS-Games'09).* Technocentre, Coventry University, Coventry, U.K., IEEE.

PAPAKONSTANTINOU, S. & LIAROKAPIS, F. (2007) LOCUS - Augmented Reality Interface. *KTN Flagship Projects Day.* LOCUS ed. National Physical Laboratory (NPL), Teddington, U.K.

PARALLELGRAPHICS (2003) ParallelGraphics Cortona SDK 4.1 Documentation. Dublin, Ireland, ParallelGraphics.

PASCOE, J. (1998) Adding Generic Contextual Capabilities to Wearable Computers. *Proceedings of the 2nd IEEE International Symposium on Wearable Computers.* IEEE Computer Society.

PENNINGTON, S. (2001) Corporate Mobility: The Next Big Wave in Business. *Available from:* http://www.mobilecommerceworld.com/Tmpl/article.asp?CID=1&AID=5169&TCode=FT&T1=21/7/2002.

PIEKARSKI, W. & THOMAS, B. H. (2003) An Object-Oriented Software Architecture for 3D Mixed Reality Applications. IN TAMURA, H. & YOKOYA, N. (Eds.) *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 2003).* Tokyo, Japan, IEEE Computer Society.

PINGEL, T. J. & CLARKE, K. C. (2005) Assessing the usability of a wearable computer system for outdoor pedestrian navigation. *Proceedings of AutoCarto 2005.* Las Vegas, USA, Ingenta.

POLANYI, M. (1966) *Tacit Dimension*, DoubleDay & Co.

PORTER, M. E. (1980) *Competitive Strategy: Techniques for Analyzing Industries and Competitors*, The Free Press.

PREECE, J., ROGERS, Y. & SHARP, H. (2002) *Interaction Design: Beyond Human-Computer Interaction*, John Wiley & Sons.

PULLI, K., AARNIO, T., ROIMELA, K. & VAARALA, J. (2005) Designing Graphics Programming Interfaces for Mobile Devices. *IEEE Computer Graphics and Applications,* 25**,** 66-75.

RAATIKAINEN, K., CHRISTENSEN, H. B. & NAKAJIMA, T. (2002) Application requirements for middleware for mobile and pervasive systems. *ACM SIGMOBILE Mobile Computing and Communications Review,* 6**,** pp. 16-24.

RAKKOLAINEN, I., TIMMERHEID, J. & VAINIO, T. (2001) A 3D City Info for mobile users. *Computers and Graphics,* 25**,** pp. 619-625(7).

RAPER, J. (2000) *Multidimensional Geographic Information Science* London, Taylor and Francis.

RAPER, J. (2002) Locational Privacy: a new challenge for geographic information science. *5th AGILE Conference on Geographic Information Science.* Universitat de les Illes Balears, LSIG, Palma, Mallorca.

RAPER, J., GARTNER, G., KARIMI, H. & RIZOS, C. (2007) A critical evaluation of location based services and their potential. *Journal of Location Based Services,* 1**,** 5-45.

RASHID, O., MULLINS, I., COULTON, P. & EDWARDS, R. (2006) Extending Cyberspace: Location Based Games using Cellular Phones. *Computers in Entertainment (CIE),* Vol. 4**,** 4.

REICHENBACHER, T. (2005) Adaptive egocentric maps for mobile users. IN MENG, L., ZIPF, A. & REICHENBACHER, T. (Eds.) *Map-based Mobile Services: Theories, Methods and Implementations.* Berlin, Germany, Springer-Verlag.

REICHENBACHER, T. (2007) Adaptation in mobile and ubiquitous cartography. IN CARTWRIGHT, W., PETERSON, M. P. & GARTNER, G. (Eds.) *Multimedia Cartography.* 2nd ed. Berlin, Springer Verlag.

REID, J., GEELHOED, E., HULL, R., CATER, K. & CLAYTON, B. (2005) Parallel worlds: immersion in location-based experiences. *Extended abstracts on Human Factors in Computing Systems (CHI 2005).* Portland, OR, USA, ACM.

REITMAYR, G. & DRUMMOND, T. (2006) Going out: robust model-based tracking for outdoor augmented reality. *Fifth IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR '06).* University of California, Santa Barbara, USA

REITMAYR, G. & SCHMALSTIEG, D. (2004) Collaborative Augmented Reality for Outdoor Navigation and Information Browsing. *Proceedings of the Symposium of Location Based Services and TeleCartography.* Vienna, Austria.

RICH, E. (1979) User modeling via stereotypes. *Cognitive Science,* 3**,** 329-354.

RICHARDSON, A., MONTELLO, D. & HEGARTY, M. (1999) Spatial knowledge acquisition from maps and from navigation in real and virtual environments. *Memory & Cognition,* 27**,** 741-750.

ROBERTSON, J. & ROBERTSON, S. (2010) Volere Requirements Specification Template. Available from: http://www.volere.co.uk/template.htm, Accessed: 29 April 2011

ROECKER, C., MAGERKURTH, C., HINSKE, S. & LAMPE, M. (2007) Designing User-Friendly Interfaces for Pervasive Gaming Applications. IN MAGERKURTH, C. & ROECKER, C. (Eds.) *Concepts and technologies for Pervasive Games - A Reader for Pervasive Gaming Research.* Achen, Germany, Shaker Verlag.

ROGERS, Y. & SCAIFE, M. (1997) Activity Theory. Available from: http://www.irit.fr/ACTIVITES/GRIC/cotcos/pjs/TheoreticalApproaches/Actvity/ActivitypaperRogers.htm, Accessed: 09 December 2011

ROMÃO, T., CORREIA, N., DIAS, E., DANADO, J., TRABUCO, A., SANTOS, C., SANTOS, R., CÂMARA, A. & NOBRE, E. (2004) ANTS--Augmented Environments. *Computers & Graphics,* 28**,** 625-633.

ROSEENTHAL, R. (1991) *Meta-analytic procedures for social research,* Newbury Park, CA, Sage Publications Inc.

RUBIO, S., DÍAZ, E., MARTÍN, J. & PUENTE, J. M. (2004) Evaluation of Subjective Mental Workload: A Comparison of SWAT, NASA-TLX, and Workload Profile Methods. *Applied Psychology,* 53**,** 61-86.

RUDDLE, R. A., PAYNE, S. J. & JONES, D. M. (1997) Navigating buildings in desktop virtual environments: Experimental investigations using extended navigational experience. *Journal of Experimental Psychology Applied,* 3**,** 143–159.

SAHA, D. & MUKHERJEE, A. (2003) Pervasive Computing: A Paradigm for the 21st Century. *IEEE Computer,* 36**,** pp. 25-31.

SALBER, D., DEY, A. K. & ABOWD, G. D. (1999) The context toolkit: aiding the development of context-enabled applications. *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit.* Pittsburgh, Pennsylvania, United States, ACM.

SALVI, J., ARMANGUÉ, X. & BATLLE, J. (2002) A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition,* 35**,** 1617-1635.

SALVI, J., BATLLE, J. & MOUADDIB, E. (1998) A robust-coded pattern projection for dynamic 3D scene measurement. *Pattern Recognition Letters,* 19**,** 1055-1065.

SANCHEZ-VIVES, M. V. & SLATER, M. (2005) From presence to consciousness through virtual reality. *Nature Reviews Neuroscience,* 6**,** 332-339.

SATYANARAYANAN, M. (2001) Pervasive Computing: Vision and Challenges. *IEEE Personal Communications,* 8**,** pp. 10--17.

SCAIFE, M. & ROGERS, Y. (1996) External cognition: how do graphical representations work? *International Journal of Human-Computer Studies,* 45**,** 185-213.

SCHALL, G., WAGNER, D., REITMAYR, G., TAICHMANN, E., WIESER, M., SCHMALSTIEG, D. & HOFMANN-WELLENHOF, B. (2009) Global pose estimation using multi-sensor fusion for outdoor Augmented Reality. *Proceedings of the 8th IEEE International Symposium on Mixed and Augmented Reality.* Washington, DC, USA, IEEE Computer Society.

SCHILIT, B. N., ADAMS, N. & WANT, R. (1994) Context-aware Computing Applications. *Proceedings of the 2nd Workshop on Mobile Computing Systems and Applications.* Santa Cruz, CA, USA, IEEE.

SCHMALSTIEG, D. & TU GRAZ, G. U. O. T. (2010) Studierstube Tracker. Available from: http://studierstube.icg.tu-graz.ac.at/handheld_ar/stbtracker.php, Accessed: 5 May 2011

SCHMIDT, A., AIDOO, K. A., TAKALUOMA, A., TUOMELA, U., LAERHOVEN, K. V. & VELDE, W. V. D. (1999) Advanced Interaction in Context. *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing.* Karlsruhe, Germany, Springer-Verlag.

SCHMIDT-BELZ, B., NICK, A., POSLAD, S. & ZIPF, A. (2002) Personalized and Location-based Mobile Tourism Services. *Workshop on "Mobile Tourism Support Systems", in the 4th International Symposium on Human Computer Interaction with Mobil Devices (Mobile HCI '02).* Pisa, Italy, Springer Berlin / Heidelberg.

SCHNEIDERMAN, B. (1998) *Designing the user interface: strategies for effective human-computer interaction*, Addison-Wesley Publishing.

SCHUBERT, T., FRIEDMANN, F. & REGENBRECHT, H. (2001) The Experience of Presence: Factor Analytic Insights. *Presence: Teleoper. Virtual Environ.,* 10**,** 266-281.

SEARBY, S. (2004) An Introduction to Personalisation and Identity. *The Journal of the Communications Network,* 3**,** pp.36-41.

SHAPIRO, S. S. & WILK, M. B. (1965) An Analysis of Variance Test for Normality (Complete Samples). *Biometrika,* 5**,** 21.

SHOLL, J. M. (2001) The Role of a Self-Reference System in Spatial Navigation. IN MONTELLO, D. R. (Ed.) *Spatial Information Theory (Proceedings of COSIT'01).* Berlin, Springer.

SIEGEL, S. (1956) Nonparametric statistics for the behavioral sciences / Sidney Siegel. New York, McGraw-Hill.

SLATER, M., KHANNA, P., MORTENSEN, J. & YU, I. (2009a) Visual Realism Enhances Realistic Response in an Immersive Virtual Environment. *IEEE Computer Graphics and Applications,* 29**,** 76-84.

SLATER, M., LINAKIS, V., USOH, M. & KOOPER, R. (1996) Immersion, Presence, and Performance in Virtual Environments: An Experiment with Tri-Dimensional Chess. *Virtual Reality Software and Technology (VRST 1996).* Hong Kong, China, ACM.

SLATER, M., LOTTO, B., ARNOLD, M. M. & SANCHEZ-VIVES, M. V. (2009b) How we experience immersive virtual environments : the concept of presence and its measurement. *Anuario de Psicología,* 40**,** 193-210.

SMAILAGIC, A. & KOGAN, D. (2002) Location Sensing and Privacy in a Context-Aware Computing Environment. *IEEE Wireless Communications.*

SOUSA, J. P. & GARLAN, D. (2002) Aura: an Architectural Framework for User Mobility in Ubiquitous Computing Environments. *Proceedings of the IFIP 17th World Computer Congress - TC2 Stream / 3rd IEEE/IFIP Conference on Software Architecture: System Design, Development and Maintenance.* Kluwer, B.V. Deventer, The Netherlands.

SPREITZER, M. & THEIMER, M. (1993) Providing location information in a ubiquitous computing environment. *In Proceedings of the 14th ACM Symposium on Operating System Principles.* Asheville, NC, USA.

STEIN, B. (1997) A Point About Polygons. *Linux Journal,* March 97.

STRANG, T. & LINNHOFF-POPIEN, C. (2004) A Context Modeling Survey. IN INDULSKA, J. & ROURE, D. D. (Eds.) *1st International Workshop on Advanced Context Modelling, Reasoning and Management, The 6th International Conference on Ubiquitous Computing (UbiComp 2004).* Nottingham, UK.

STUART, A. (1955) A Test for Homogeneity of the Marginal Distributions in a Two-Way Classification. *Biometrika,* 42**,** 412-416.

SUDEIKAT, J., BRAUBACH, L., POKAHR, A. & LAMERSDORF, W. (2005) Evaluation of Agent–Oriented Software Methodologies: Examination of the Gap Between Modelling and Platform. *Proceedings of the 5th International Workshop on Agent-Oriented Software Engineering (AOSE).* New York, NY, USA, Lecture Notes in Computer Science, Springer Berlin / Heidelberg.

SWAN, J. E. & GABBARD, J. L. (2005) Survey of User-Based Experimentation in Augmented Reality. *Proceedings 1st International Conference on Virtual Reality.* Las Vegas, Nevada, USA.

TERZIYAN, V. (2001) Architecture for Mobile P-Commerce: Multilevel Profiling Framework. *Proceedings of the Workshop on E-Business & the Intelligent Web, 17th International Joint Conference on Artificial Intelligence (IJCAI 01).* Seattle, Washington, USA, Kaufmann Publishers, San Francisco, USA (Elsevier).

TERZIYAN, V. Y. & PUURONEN, S. (2000) Reasoning with Multilevel Contexts in Semantic Metanetworks. IN BONZON, P., NOSSUM, R. T. & CAVALCANTI, M. (Eds.) *Formal Aspects of Context.* Dordrecht, The Netherlands, Kluwer Academic Publishers, Norwell, MA, USA.

THE LIBERTY ALLIANCE (2010) Liberty Alliance Project. Available from: *http://www.projectliberty.org*, Accessed: 20 April 2011

THOMAS, B., PIEKARSKI, W., HEPWORTH, D., GUNTHER, B. & DEMCZUK, V. (1998) A Wearable Computer System with Augmented Reality to Support Terrestrial Navigation. *Proceedings of the Second International Symposium on Wearable Computers (ISWC '98).* Pittsburgh, Pennsylvania, USA, IEEE Computer Society.

THOMAS, K. (2006) Mobile telephones offer a new sense of direction. *Financial Times.* London.

TOPOGRAFIX (2011) GPX 1.1 Schema Documentation. Available from: *http://www.topografix.com/gpx/1/1*, Accessed: 29 April 2011

TRYLLIAN (2007) Tryllian Agent Development Kit. Available from: *http://www.tryllian.org*, Accessed: 12 March 2007

TSAI, R. Y. (1987) A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *Journal of Robotics and Automation,* 3**,** 323- 344.

TVERSKY, B. (1981) Distortions in memory for maps. *Cognitive Psychology,* 13**,** 407-433.

TVERSKY, B. (1996) Spatial Perspective in Descriptions. IN BLOOM, P., PETERSON, M. A., NADEL, L. & GARRETT, M. F. (Eds.) *Language and Space.* Boston, USA, MIT Press.

VACCHETTI, L., LEPETIT, V. & FUA, P. (2004) Combining Edge and Texture Information for Real-Time Accurate 3D Camera Tracking. *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality.* Arlington, VA, USA, IEEE Computer Society.

VAINIO, T. & KOTALA, O. (2002) Developing 3D information systems for mobile users: some usability issues. *Proceedings of the second Nordic conference on Human-computer interaction.* Aarhus, Denmark, ACM Press.

VERBREE, E., MAREN, G. V., GERMS, R., JANSEN, F. & KRAAK, M.-J. (1999) Interaction in virtual world views-linking 3D GIS with VR. *International Journal of Geographical Information Science,* 13**,** 12.

VEZHNEVETS, V. & VELIZHEV, A. (2005) GML C++ Camera Calibration Toolbox. http://research.graphicon.ru/calibration/gml-c++-camera-calibration-toolbox.html.

VIEGA, J., KOHNO, T. & POTTER, B. (2001) Trust (and mistrust) in secure applications. *Communications of the ACM,* 44**,** pp. 31-36.

VINAYAGAMOORTHY, V., BROGNI, A., GILLIES, M., SLATER, M. & STEED, A. (2004) An Investigation of Presence Response across Variations in Visual Realism. *Proceedings of the 7th International Conference on Presence.* Valencia, Spain.

VINCENT PERVASIVE MEDIA TECHNOLOGIES (2008) Vincent 3D Rendering Library. Available from: *http://www.vincent3d.com*, Accessed: 29 April 2011

VINSON, N. G. (1999) Design guidelines for landmarks to support navigation in virtual environments. *Proceedings of the SIGCHI conference on Human factors in computing systems: The CHI is the limit.* Pittsburgh, Pennsylvania, United States, ACM Press.

W3C, W. W. W. C. (2007) Composite Capability/Preference Profiles (CC/PP). KISS, C., Available from: *http://www.w3.org/TR/2007/WD-CCPP-struct-vocab2-20070430*, Accessed: 20 April 2011

W3C, W. W. W. C. (2011) Semantic Web. Available from: *http://www.w3.org/2001/sw*, Accessed: 29 April 2011

WACHOWICZ, M., BULENS, J., RIP, F., KRAMER, H., RON VAN LAMMEREN & LIGTENBERG, A. (2002) GeoVR construction and use: The seven factors. *5th AGILE Conference on Geographic Information Science.* Palma, Balearic Islands, Spain.

WAG, W. A. P. F. (2001) UAProf. Available from: *http://www.openmobilealliance.org/tech/affiliates/wap/wap-248-uaprof-20011020-a.pdf*, Accessed: 20 April 2011

WAGNER, D. (2009) History of Mobile AR. REALITY, C. D. L. F. H. A., Available from: *https://www.icg.tugraz.at/~daniel/HistoryOfMobileAR/*, Accessed: 19 December 2011

WAGNER, D. & SCHMALSTIEG, D. (2007) ARToolKitPlus for Pose Tracking on Mobile Devices. IN GRABNER, M. & GRABNER, H. (Eds.) *Proceedings of 12th Computer Vision Winter Workshop (CVWW'07).* Graz Technical University, St. Lambrecht, Austria.

WAGNER, M., BALKE, W.-T., HIRSCHFELD, R. & KELLERER, W. (2002) A Roadmap to Advanced Personalization of Mobile Services. *Proceedings of the International Federated Conferences DOA, ODBASE, CoopIS 2002.* University of California, Irvine, CA, USA, Lecture Notes in Computer Science, Springer Verlag, Berlin / Heidelberg.

WALTHER, B. K. (2005) Reflections on the methodology of pervasive gaming. *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology.* Valencia, Spain, ACM.

WANG, J., ZHAI, S. & CANNY, J. (2006) Camera phone based motion sensing: interaction techniques, applications and performance study. *Proceedings of the 19th annual ACM symposium on User interface software and technology.* Montreux, Switzerland, ACM.

WANT, R. & PERING, T. (2005) System challenges for ubiquitous & pervasive computing. *Proceedings of the 27th international conference on Software engineering.* St. Louis, Missouri, USA, ACM Press. New York, NY, USA.

WARD, A., STEGGLES, P., CURWEN, R. & WEBSTER, P. (1998) Sentient Computing Project. GROUP, T. D., Available from: http://www.cl.cam.ac.uk/research/dtg/attarchive/spirit/, Accessed: 09 November 2011

WASINGER, R. & KRÜGER, A. (2004) Multi-modal Interaction with Mobile Navigation Systems. *Information Technology (it) Special Issue: "Conversational User Interfaces",* Volume: 46**,** 322-331.

WASINGER, R., STAHL, C. & KRÜGER, A. (2003) Mobile Multi-Modal Pedestrian Navigation. *In 2nd International Workshop on Interactive Graphical Communication (IGC2003).* London, UK.

WAYNE, P. & BRUCE, T. (2002) ARQuake: the outdoor augmented reality gaming system. *Communications of the ACM,* 45**,** 36-38.

WEB3D CONSORTIUM (2011) What is X3D? Available from: http://www.web3d.org/realtime-3d/x3d/what-x3d, Accessed: 13 December 2011

WEIDENHAUPT, K., POHL, K., JARKE, M. & HAUMER, P. (1998) Scenarios in System Development: Current Practice. *IEEE Software,* 15**,** 34-45.

WEISER, M. (1993) Some computer science issues in ubiquitous computing. *Communications of the ACM,* 36**,** pp.75-84.

WEISER, M. (2002) The computer for the 21 century. *IEEE Pervasive Computing,* 1**,** pp. 19-25.

WEIBENBERG, N., VOISARD, A. & GARTMANN, R. (2004) Using ontologies in Personalized Mobile Applications. *Proceedings of the 12th annual ACM international workshop on Geographic Information Systems.* Washington DC, USA, ACM Press, New York, NY, USA.

WILCOXON, F. (1945) Individual Comparisons by Ranking Methods. *Biometrics Bulletin,* 1**,** 80-83.

WITMER, B. G. & SINGER, M. J. (1998) Measuring Presence in Virtual Environments: A Presence Questionnaire. *Presence: Teleoperators and Virtual Environments,* 7**,** 225-240.

WOOD, J. (2005) Multim im parvo - Many Things in a Small Place. IN DYKES, J., MACEACHREN, A. M. & KRAAK, M.-J. (Eds.) *Exploring Geovisualization.* 1st ed. Oxford, U.K., Elsevier.

WOOD, M., PEARSON, D. G., CALDER, C. & MILLER, D. (2005) Comparing the effects of 2D and 3D representations on human wayfinding. IN GARTNER, G. F. (Ed. *Proceedings of the Symposium 2005 on Location Based Services and Telecartography.* Vienna, Austria: Technische Universita¨t Wien.

YU, E. (1997) Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. *Proceedings of the 3rd IEEE Inernational Symposium on Requirements Engineering (RE'97).* Washington D.C., USA.

YU, E. S. K. & MYLOPOULOS, J. (1994) Understanding "why" in software process modelling, analysis and design. *Proceedings of the 16th international conference on Software engineering.* Sorrento, Italy, IEEE Computer Society Press.

YUAN, M., BUTTENFIELD, B., GAHEGAN, M. & MILLER, H. (2004) Geospatial Data Mining and Knowledge Discovery. IN MCMASTER, R. B. & USERY, E. L. (Eds.) *A Research Agenda for Geographic Information Science.* Boca Raton, Florida, CRC Press.

ZAHORIK, P. & JENISON, R. L. (1998) Presence as Being-in-the-World. *Presence: Teleoper. Virtual Environ.,* 7**,** 78-89.

ZIPF, A. (2002) User-Adaptive Maps for Location-Based Services (LBS) for Tourism. *9th International Conference for Information and Communication Technologies in Tourism (ENTER 2002).* Innsbruck, Austria, Springer Verlag.

11 Bibliography

This chapter presents information about the sources of documentation, mostly found online, that were considered relevant and useful during the course of the research. They are presented in an *Annotated* style and they are sorted in alphabetical order.

Aka-Aki, N. (2011). "Aka-Aki Homepage." Retrieved 29 April, 2011, from: http://www.aka-aki.com.

Franson (2011) Franson GpsTools.." Retrieved 29 April, 2011, from: http://franson.com/gpstools.

Foursquare. (2011). "Foursquare." Retrieved 29 April, 2011, from: http://foursquare.com.

Google. (2011). "Google Advertising Programs." Retrieved 29 April, 2011, from: http://www.google.com/ads.

Google. (2011). "Google Maps." Retrieved 5 May 2011, from: http://maps.google.com.

Google. (2011). "Google Latitude." Retrieved 5 May 2011, from: http://www.google.co.uk/latitude.

Hutchison, G. (2011). "3 Mobile." Retrieved 29 April, 2011, from: http://www.three.co.uk.

IBM, R. (2000). "The PIMA Project: Platform-Independent Model for Applications." Retrieved 29 April, 2011, from: http://www.research.ibm.com/PIMA/index.html.

iPhoneNess. (2010). "40 Best Augmented Reality iPhone Apps." Retrieved 21 April, 2011, from: http://www.iphoneness.com/iphone-apps/best-augmented-reality-iphone-applications.

ITU. (2007). "International Telecommunication Union." Retrieved 12 March, 2007, from: http://www.itu.int/home/index.html.

Layar. (2011). "Layar Reality Browser." Retrieved 5 May, 2011, from: http://www.layar.com.

Mobilizy. (2011). "Wikitude." Retrieved 5 May, 2011, from: http://www.wikitude.org.

Moore, D. S. (1995). The Basic Practice of Statistics. New York, NY, W.H.Freeman & Co.

Moore, D. S. and G. P. McCabe (1998). The Practice of Statistics. New York, NY, W. H. Freeman and Co.

MySQL. (2011). " MySQL 5.6 Spatial Extensions." Retrieved 29 April, 2011, from: http://dev.mysql.com/doc/refman/5.6/en/spatial-extensions.html.

OGC, O. G. C. (2011). "Open Geospatial Consortium." Retrieved 29 April, 2011, from: http://www.opengeospatial.org.

OMA, O. M. A. (2010). "Open Mobile Alliance." Retrieved 29 April, 2011, from: http://www.openmobilealliance.org.

Oracle. (2011). "Oracle Spatial & Oracle Locator." Retrieved 29 April, 2011, from: http://www.oracle.com/technetwork/database/options/spatial/index.html.

Tukey, J. W. (1977). Exploratory Data Analysis. Reading, MA, Addison Wesley.

Twitter, I. (2011). "Twitter." Retrieved 7 May, 2011, from: http://twitter.com.

UMTS World. (February 2007). "3G and UMTS Technology." Retrieved 12 March, 2007, from: http://www.umtsworld.com.

W3C, W. W. W. C. (2010). "Scalable Vector Graphics (SVG)." Retrieved 29 April, 2011, from: http://www.w3.org/Graphics/SVG.

W3C, W. W. W. C. (2011). "The World Wide Web Consortium" Retrieved 29 April, 2011, from: http://www.w3.org.

Yahoo! (2011). "Flickr." Retrieved 29 April, 2011, from: http://www.flickr.com.

12 Appendices

This chapter contains relevant and explanatory information about several aspects applicable to this project.

12.1 Appendix I – Requirements Acquisition Questionnaire

This Appendix presents the *Questionnaire* that was provided to the participants of the *Requirements Acquisition Survey*.

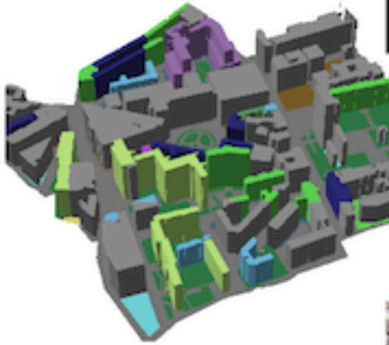
The *Questionnaire* can also be found in digital format on the CD that accompanies the Thesis. It was created with Adobe Acrobat v7.0 (.pdf), but is accessible by v6.0 onwards.

School of Informatics
CITY UNIVERSITY LONDON

Research Questionnaire

Dept. of Information Science


Augmented Reality interface with stylized aids


Virtual Reality interface without texturing (oblique view)


Virtual Reality interface with texturing (1st person view)

Aim of this questionnaire

This questionnaire was designed to acquire user-related requirements, which are going to form a user model, based on the opinions of the participants. The user model will feed the development methodology of a mobile system aiming to deliver contextualised services to the user through real-time information (e.g. location & preferences) processing and interaction with remote parties. Visualisation and interaction of relevant information will be accomplished through the use of advanced user interfaces (e.g. VR & AR), which can complement the usability of current mobile devices (e.g. phones & PDAs)

Author(s): Stelios Papakonstantinou	**Version**: 6.0
Contact & Submission e-mail: stelios@soi.city.ac.uk	**Release Date**: 19 March 2009

Preface

Familiarity with Mobile Devices

1. How many years have you been using a mobile phone or PDA? ______

2. Which is the latest mobile phone or PDA that you own?	
Manufacturer:	
Model:	

3. Which are your favourite functionalities of your mobile device? *(tick the appropriate box(es))*

☐ Web Browsing	☐ Navigation	☐ Games	☐ Contacts & Time Management
☐ Data Transfer	☐ Multimedia	☐ Messaging	☐ Other *(specify)*:

4. My mobile device assists me in many daily activities. *(tick the most appropriate)*

Strongly Disagree ○	Disagree ○	Undecided ○	Agree ○	Strongly Agree ○

5. How often do you connect your mobile phone to the Internet? *(tick the most appropriate)*

Never ○	Monthly ○	Weekly ○	Daily ○	Always Connected ○

Familiarity with Information Management

6. I often search for the location of an item, person or place with the help of a computing device. *(tick the most appropriate)*

Strongly Disagree ○	Disagree ○	Undecided ○	Agree ○	Strongly Agree ○

7. I often navigate towards a location with the help of a computing device? *(tick the most appropriate)*

Strongly Disagree ○	Agree ○	Undecided ○	Agree ○	Strongly Agree ○

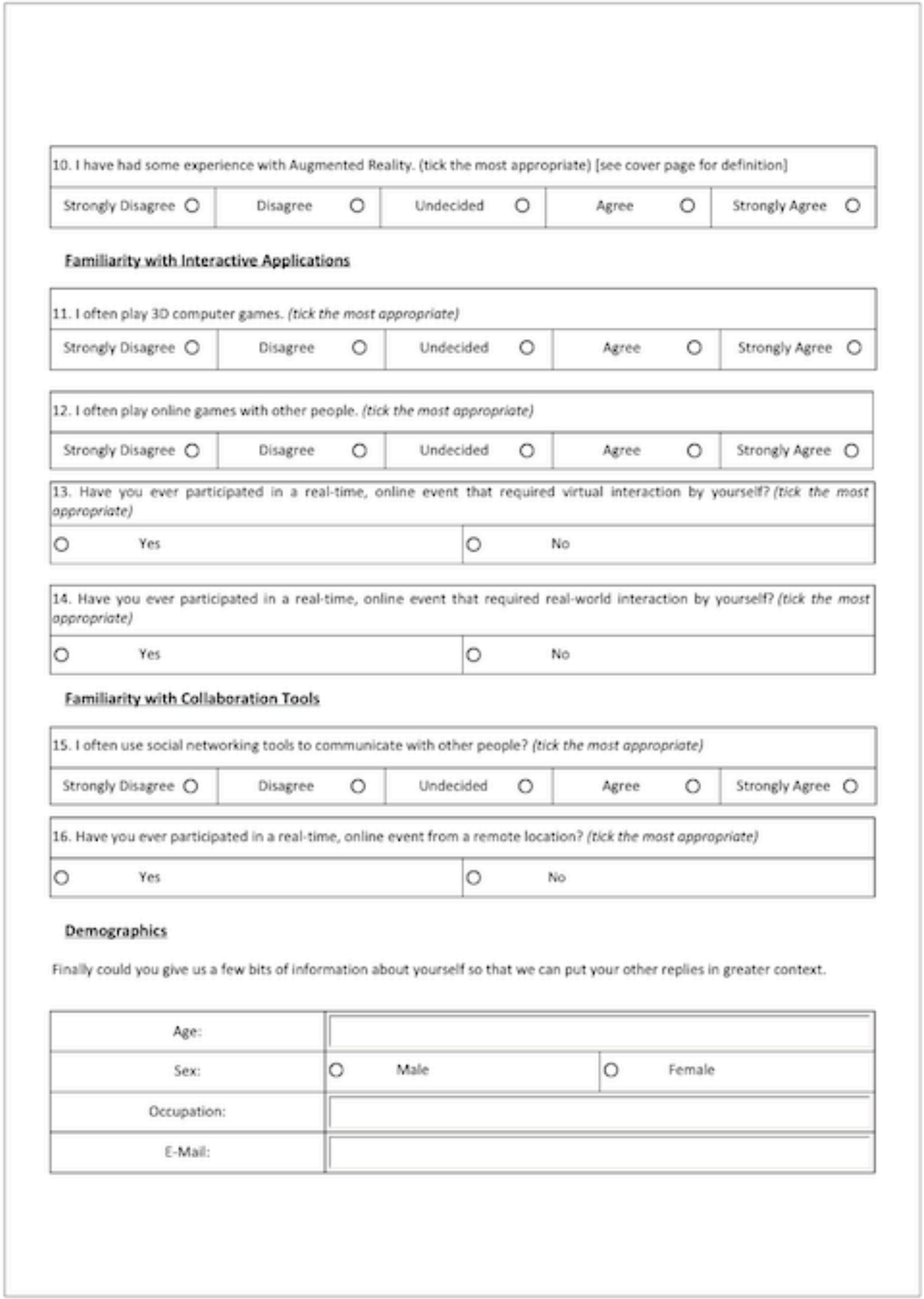
8. How many public profiles have you created in exchange for personalized services? *(tick the most appropriate)*

None ○	1-3 ○	4-10 ○	11-20 ○	More than 20 ○

Familiarity with User Interfaces

9. I have had some experience with Virtual Reality. (tick the most appropriate) [see cover page for definition]

Strongly Disagree ○	Disagree ○	Undecided ○	Agree ○	Strongly Agree ○



10. I have had some experience with Augmented Reality. (tick the most appropriate) [see cover page for definition]				
Strongly Disagree ○	Disagree ○	Undecided ○	Agree ○	Strongly Agree ○

Familiarity with Interactive Applications

11. I often play 3D computer games. *(tick the most appropriate)*				
Strongly Disagree ○	Disagree ○	Undecided ○	Agree ○	Strongly Agree ○

12. I often play online games with other people. *(tick the most appropriate)*				
Strongly Disagree ○	Disagree ○	Undecided ○	Agree ○	Strongly Agree ○

13. Have you ever participated in a real-time, online event that required virtual interaction by yourself? *(tick the most appropriate)*	
○ Yes	○ No

14. Have you ever participated in a real-time, online event that required real-world interaction by yourself? *(tick the most appropriate)*	
○ Yes	○ No

Familiarity with Collaboration Tools

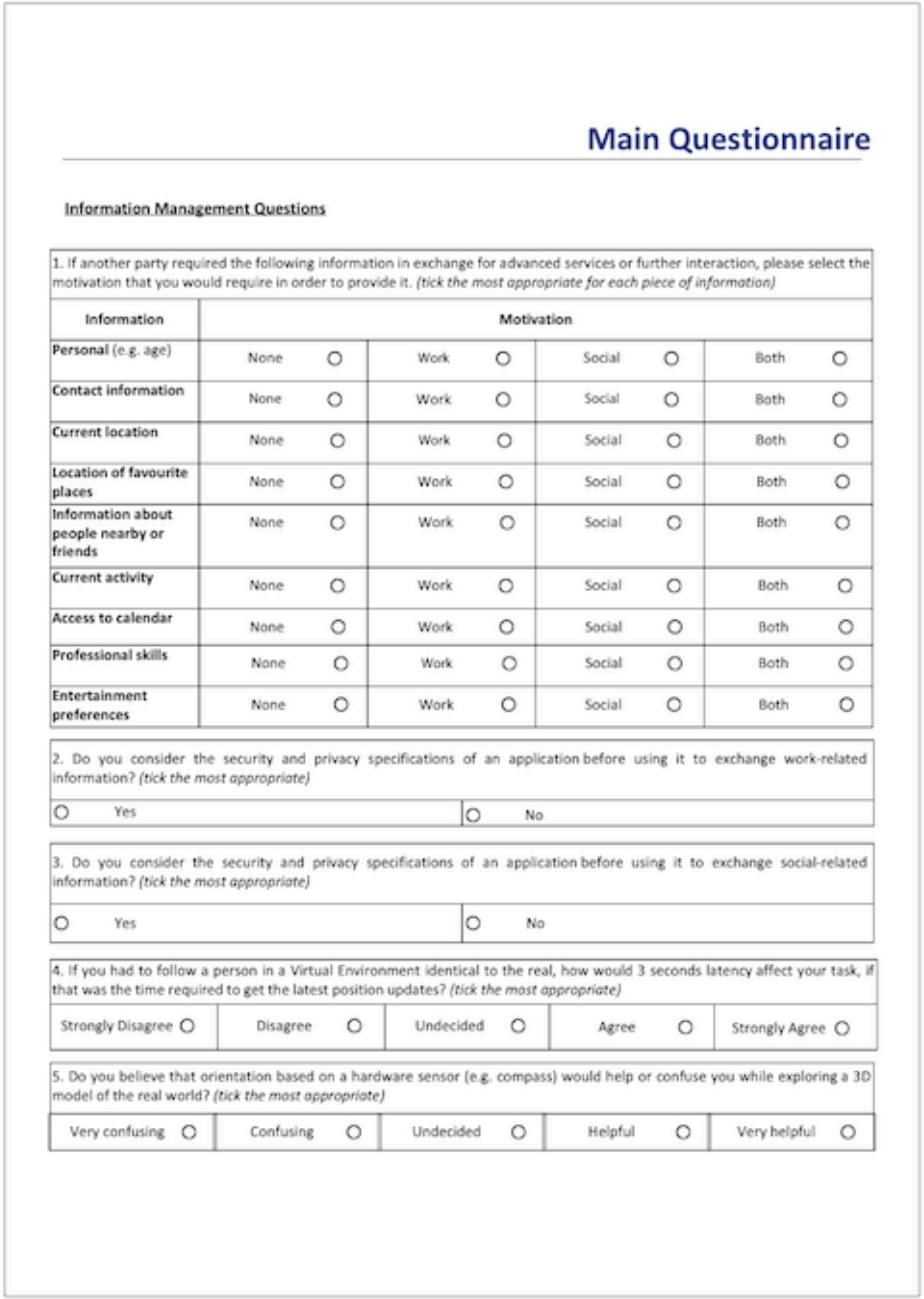
15. I often use social networking tools to communicate with other people? *(tick the most appropriate)*				
Strongly Disagree ○	Disagree ○	Undecided ○	Agree ○	Strongly Agree ○

16. Have you ever participated in a real-time, online event from a remote location? *(tick the most appropriate)*	
○ Yes	○ No

Demographics

Finally could you give us a few bits of information about yourself so that we can put your other replies in greater context.

Age:		
Sex:	○ Male	○ Female
Occupation:		
E-Mail:		



Main Questionnaire

Information Management Questions

1. If another party required the following information in exchange for advanced services or further interaction, please select the motivation that you would require in order to provide it. *(tick the most appropriate for each piece of information)*

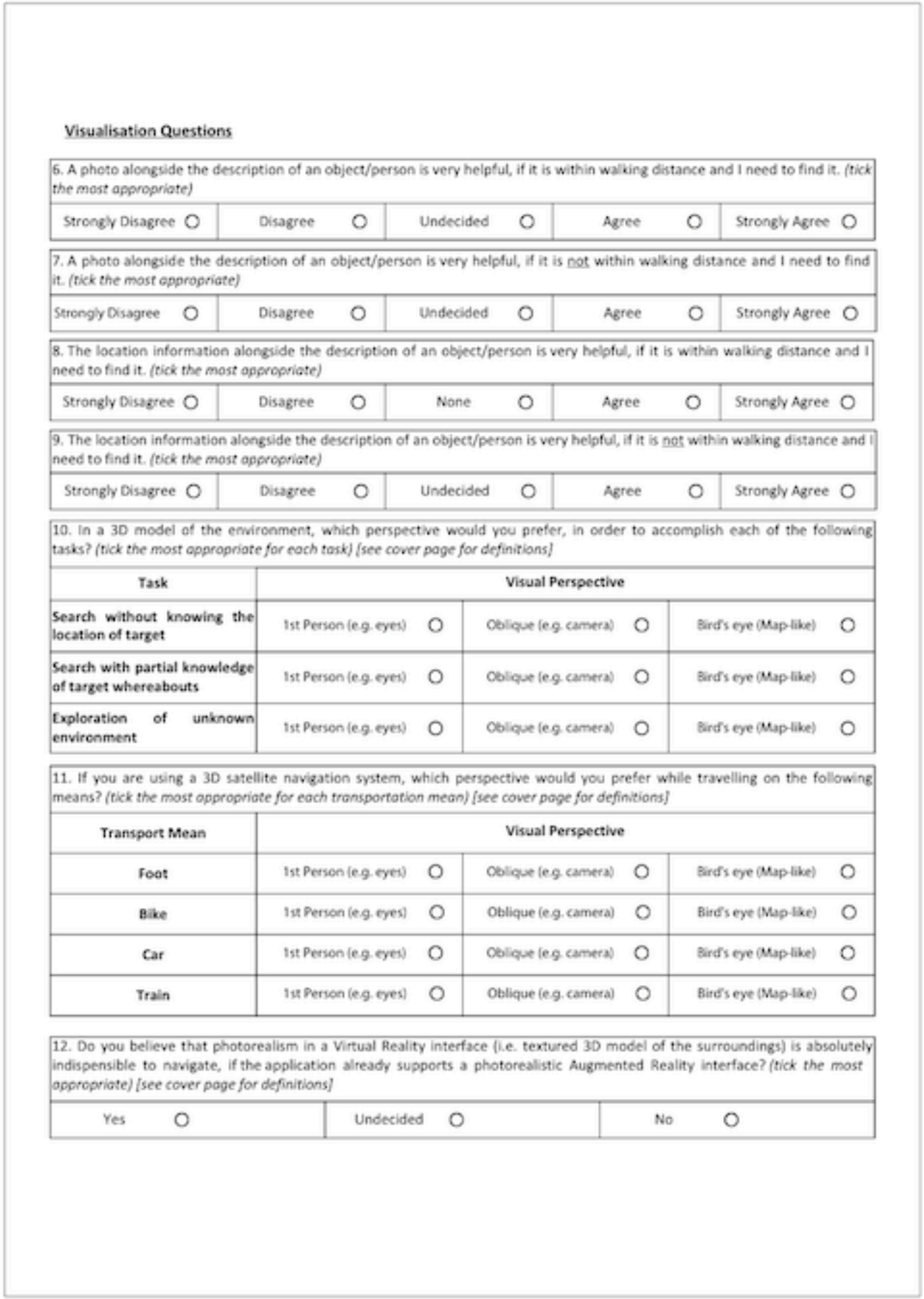
Information	Motivation			
Personal (e.g. age)	None ○	Work ○	Social ○	Both ○
Contact information	None ○	Work ○	Social ○	Both ○
Current location	None ○	Work ○	Social ○	Both ○
Location of favourite places	None ○	Work ○	Social ○	Both ○
Information about people nearby or friends	None ○	Work ○	Social ○	Both ○
Current activity	None ○	Work ○	Social ○	Both ○
Access to calendar	None ○	Work ○	Social ○	Both ○
Professional skills	None ○	Work ○	Social ○	Both ○
Entertainment preferences	None ○	Work ○	Social ○	Both ○

2. Do you consider the security and privacy specifications of an application before using it to exchange work-related information? *(tick the most appropriate)*	
○ Yes	○ No

3. Do you consider the security and privacy specifications of an application before using it to exchange social-related information? *(tick the most appropriate)*	
○ Yes	○ No

4. If you had to follow a person in a Virtual Environment identical to the real, how would 3 seconds latency affect your task, if that was the time required to get the latest position updates? *(tick the most appropriate)*				
Strongly Disagree ○	Disagree ○	Undecided ○	Agree ○	Strongly Agree ○

5. Do you believe that orientation based on a hardware sensor (e.g. compass) would help or confuse you while exploring a 3D model of the real world? *(tick the most appropriate)*				
Very confusing ○	Confusing ○	Undecided ○	Helpful ○	Very helpful ○



Visualisation Questions

6. A photo alongside the description of an object/person is very helpful, if it is within walking distance and I need to find it. *(tick the most appropriate)*				
Strongly Disagree ○	Disagree ○	Undecided ○	Agree ○	Strongly Agree ○

7. A photo alongside the description of an object/person is very helpful, if it is not within walking distance and I need to find it. *(tick the most appropriate)*				
Strongly Disagree ○	Disagree ○	Undecided ○	Agree ○	Strongly Agree ○

8. The location information alongside the description of an object/person is very helpful, if it is within walking distance and I need to find it. *(tick the most appropriate)*				
Strongly Disagree ○	Disagree ○	None ○	Agree ○	Strongly Agree ○

9. The location information alongside the description of an object/person is very helpful, if it is not within walking distance and I need to find it. *(tick the most appropriate)*				
Strongly Disagree ○	Disagree ○	Undecided ○	Agree ○	Strongly Agree ○

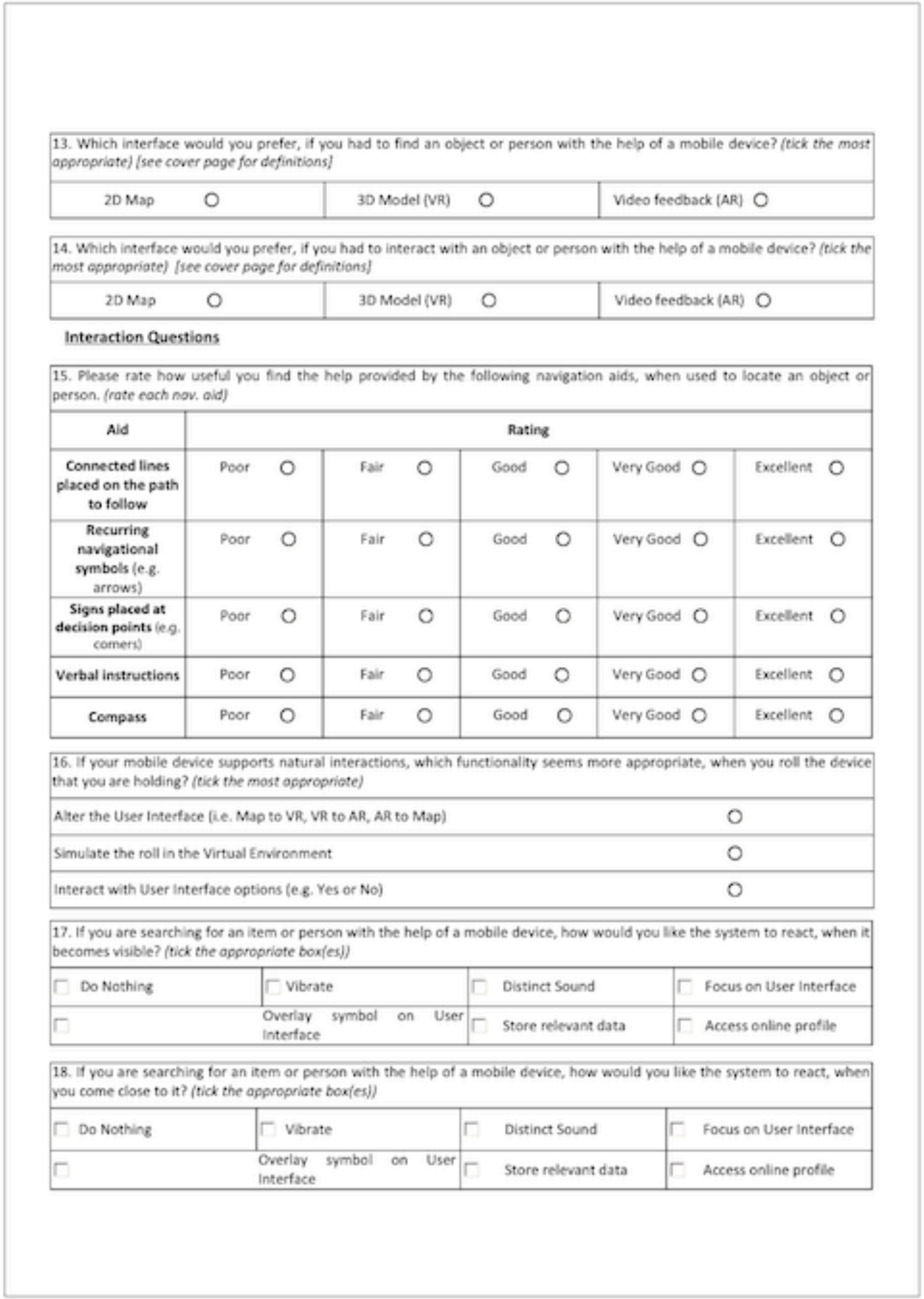
10. In a 3D model of the environment, which perspective would you prefer, in order to accomplish each of the following tasks? *(tick the most appropriate for each task) [see cover page for definitions]*

Task	Visual Perspective		
Search without knowing the location of target	1st Person (e.g. eyes) ○	Oblique (e.g. camera) ○	Bird's eye (Map-like) ○
Search with partial knowledge of target whereabouts	1st Person (e.g. eyes) ○	Oblique (e.g. camera) ○	Bird's eye (Map-like) ○
Exploration of unknown environment	1st Person (e.g. eyes) ○	Oblique (e.g. camera) ○	Bird's eye (Map-like) ○

11. If you are using a 3D satellite navigation system, which perspective would you prefer while travelling on the following means? *(tick the most appropriate for each transportation mean) [see cover page for definitions]*

Transport Mean	Visual Perspective		
Foot	1st Person (e.g. eyes) ○	Oblique (e.g. camera) ○	Bird's eye (Map-like) ○
Bike	1st Person (e.g. eyes) ○	Oblique (e.g. camera) ○	Bird's eye (Map-like) ○
Car	1st Person (e.g. eyes) ○	Oblique (e.g. camera) ○	Bird's eye (Map-like) ○
Train	1st Person (e.g. eyes) ○	Oblique (e.g. camera) ○	Bird's eye (Map-like) ○

12. Do you believe that photorealism in a Virtual Reality interface (i.e. textured 3D model of the surroundings) is absolutely indispensible to navigate, if the application already supports a photorealistic Augmented Reality interface? *(tick the most appropriate) [see cover page for definitions]*		
Yes ○	Undecided ○	No ○



13. Which interface would you prefer, if you had to find an object or person with the help of a mobile device? *(tick the most appropriate) [see cover page for definitions]*		
2D Map ○	3D Model (VR) ○	Video feedback (AR) ○

14. Which interface would you prefer, if you had to interact with an object or person with the help of a mobile device? *(tick the most appropriate) [see cover page for definitions]*		
2D Map ○	3D Model (VR) ○	Video feedback (AR) ○

Interaction Questions

15. Please rate how useful you find the help provided by the following navigation aids, when used to locate an object or person. *(rate each nav. aid)*

Aid	Rating				
Connected lines placed on the path to follow	Poor ○	Fair ○	Good ○	Very Good ○	Excellent ○
Recurring navigational symbols (e.g. arrows)	Poor ○	Fair ○	Good ○	Very Good ○	Excellent ○
Signs placed at decision points (e.g. corners)	Poor ○	Fair ○	Good ○	Very Good ○	Excellent ○
Verbal instructions	Poor ○	Fair ○	Good ○	Very Good ○	Excellent ○
Compass	Poor ○	Fair ○	Good ○	Very Good ○	Excellent ○

16. If your mobile device supports natural interactions, which functionality seems more appropriate, when you roll the device that you are holding? *(tick the most appropriate)*	
Alter the User Interface (i.e. Map to VR, VR to AR, AR to Map)	○
Simulate the roll in the Virtual Environment	○
Interact with User Interface options (e.g. Yes or No)	○

17. If you are searching for an item or person with the help of a mobile device, how would you like the system to react, when it becomes visible? *(tick the appropriate box(es))*			
☐ Do Nothing	☐ Vibrate	☐ Distinct Sound	☐ Focus on User Interface
☐	Overlay symbol on User Interface	☐ Store relevant data	☐ Access online profile

18. If you are searching for an item or person with the help of a mobile device, how would you like the system to react, when you come close to it? *(tick the appropriate box(es))*			
☐ Do Nothing	☐ Vibrate	☐ Distinct Sound	☐ Focus on User Interface
☐	Overlay symbol on User Interface	☐ Store relevant data	☐ Access online profile

19. If you would like to explore a 3D world on your mobile device screen, which input method would you prefer? *(tick the most appropriate)*

Touch-screen/Stylus ○	Navigation button ○	Rotate mobile device ○

20. Which way do you prefer to pass your preferences to a mobile application? *(tick the most appropriate)*

Create, once, a complete user profile that is available every time the application runs	○
Input core information every time the application is executed	○
Do not create a user profile. Provide explicit answer to all questions, when user interaction is required	○

21. How often are you willing to update a mobile application with information about your current activity or task in progress? *(tick the most appropriate)*

Once, when the application is executed ○	Every time I change activity ○
At random intervals ○	After reminder ○

Collaboration Questions

22. In a mobile 3D environment, if you needed to communicate with another person, which feedback type would you prefer, if that person was ...

... visible from your point of view *(tick the most appropriate)*

Text Description ○	Static image ○	3D Avatar ○	Video-conference style ○

... not visible from your point of view *(tick the most appropriate)*

Text Description ○	Static image ○	3D Avatar ○	Video-conference style ○

23. How much actual time are you willing to consume while participating in a real-time entertainment event that requires virtual, as well as real-world interaction and lasts for ... *(tick the most appropriate for each event) [continues in next page]*

Event Time	Percentage (Actual Time)				
1 hour (e.g. Hide & Seek)	5% (3min) ○	10% (6min) ○	25% (15min) ○	50% (30min) ○	100% ○
3 hours (e.g. Race)	5% (9min) ○	10% (18min) ○	25% (45min) ○	50% (1.5hrs) ○	100% ○
6 hour (e.g. Action / Shooter)	5% (18min) ○	10% (36min.) ○	25% (1.5hrs) ○	50% (3hrs) ○	100% ○
1 day (e.g. Adventure)	5% (72min) ○	10% (2.4hrs) ○	25% (6hrs) ○	50% (12hrs) ○	100% ○
1 week (e.g. Social)	5% (8.4hrs) ○	10% (16.8hrs) ○	25% (42hrs) ○	50% (3.5days) ○	100% ○

24. How far are you prepared to travel to reach your final goal, while participating in a real-time entertainment event that requires virtual, as well as real-world interaction and lasts for ... *(complete distance for each event) [in meters]*

Event Time	Distance
1 hour (e.g. Hide & Seek)	Meters:
3 hours (e.g. Race)	Meters:
6 hour (e.g. Action /Shooter)	Meters:
1 day (e.g. Adventure)	Meters:
1 week (e.g. Social)	Meters:

25. Consider yourself standing outside a train station, trying to find the restaurant, where your friends have lunch. Would you follow the route sent to you by one of your friends or try to find your own way to the restaurant? *(tick the most appropriate)*

○ Friend's suggestion	○ My way

26. If, accidentally, you were found at the same train station with a person that you have met online and share the same interests, would you initiate conversation through a mobile application, in order to meet him in-person, while waiting for the train? *(tick the most appropriate)*

○ Yes	○ No

27. If you need to be guided by a person who is at a remote location, which feedback combination would you find more valuable? *(tick the appropriate box(es))*

☐ Written Instructions	☐ Minimal verbal instructions	☐ Extensive verbal instructions
☐ 2D Map with nav. aids	☐ 3D Map with nav. aids	☐ Video feedback with nav. aids

28. Would you contact a person who is at an event (e.g. concert) that you would like as well to be? *(tick the most appropriate)*

○ Yes	○ No

29. If you find out that a friend is at an event that you would like to attend as well, would it increase the possibility to contact him and experience the virtual version of that event? *(tick the most appropriate)*

○ Yes	○ No

Submit by Email

12.2 Appendix II – Requirements Acquisition Data

The raw *Data* that has been collected from the participant replies during the *Requirements Acquisition Survey* is presented in this Appendix.

The *Data* file includes the subjective responses retrieved from 30 individuals that participated in the *Survey*. Any features that can identify the individuals have been intentionally removed due to privacy protection issues.

The *Data* can be found in digital format, accessible by IBM SPSS v17.0.0 (.sav) or later, on the CD that accompanies the Thesis.

12.3 Appendix III – Requirements Acquisition Analysis

The *Statistical Exploratory Analysis* generated by the data collected from the participant replies during the *Requirements Acquisition Survey* is presented in this Appendix.

The type of most variables is categorical, either ordinal or nominal. Furthermore, the distribution of our sample was not normal in most cases. Thus, most of the results found in this section do not validate *Parametric Test* assumptions and have been taken under consideration as preliminary user reactions that directed further developments of the system. A selected set of the *Analysis* was included in the main body of the Thesis. The full set is provided in this Appendix.

The *Statistical Exploratory Analysis* can be found in digital format, accessible either by IBM SPSS v17.0.0 (.sav) or Adobe Acrobat v6.0 (.pdf), on the CD that accompanies the Thesis.

The folder contains two sets of files – those that examine the *Frequencies* of string variables and those that *Explore* numerical variables.

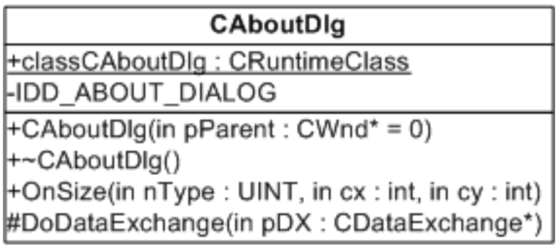
12.4 Appendix IV – Windows Mobile Device Specifications

A sample of the smartphone devices that were available in the market while working on this project is provided in this Appendix. Every device supports the Windows Mobile OS. The available specifications for each device are listed according to the manufacturer name and model.

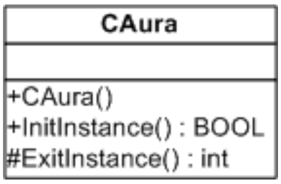
The *Device List* can be found in digital format, accessible by Adobe Acrobat v6.0 (.pdf) or later, on the CD that accompanies the Thesis.

12.5 Appendix V – Aura’s Class Diagrams

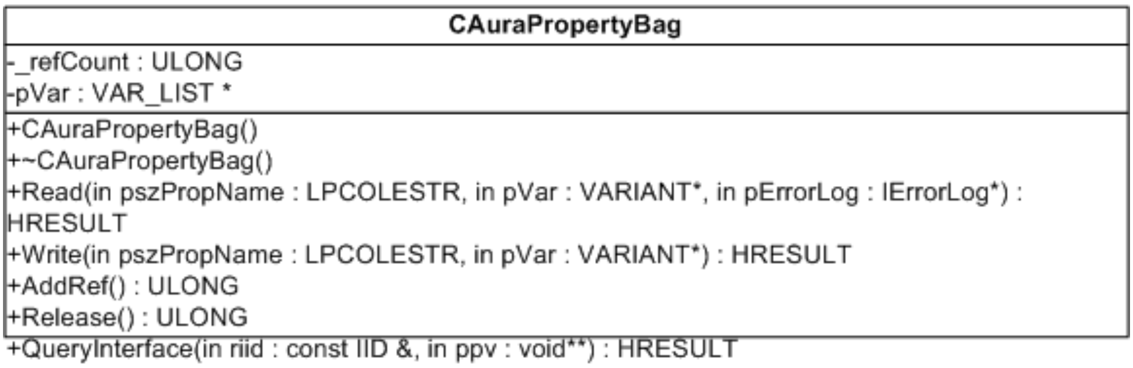
In this Appendix, the reader can inspect the analytical structure of 24 high-level classes that formulate the advanced functionality of the system.



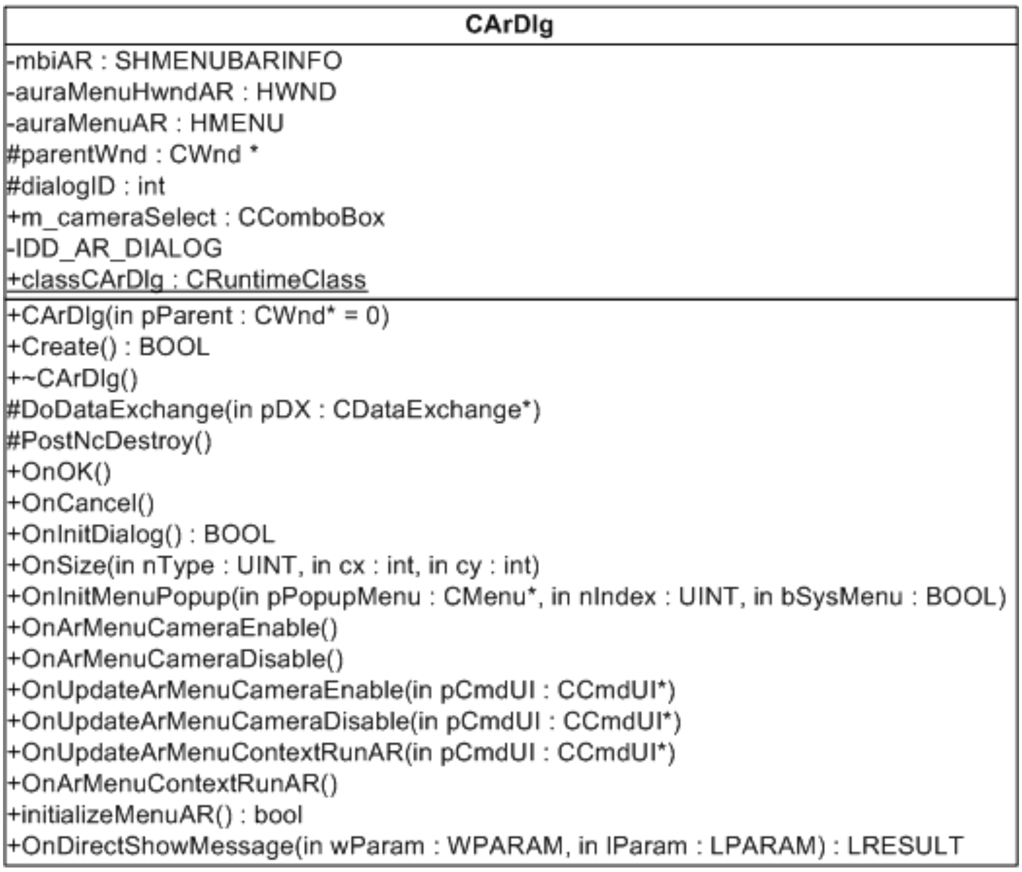
```
CAboutDlg
+classCAboutDlg : CRuntimeClass
-IDD_ABOUT_DIALOG
+CAboutDlg(in pParent : CWnd* = 0)
+~CAboutDlg()
+OnSize(in nType : UINT, in cx : int, in cy : int)
#DoDataExchange(in pDX : CDataExchange*)
```

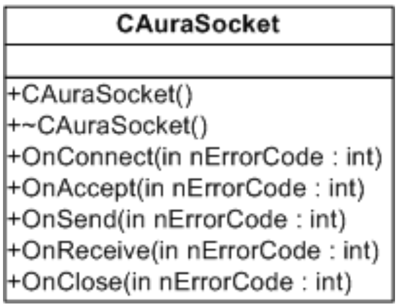


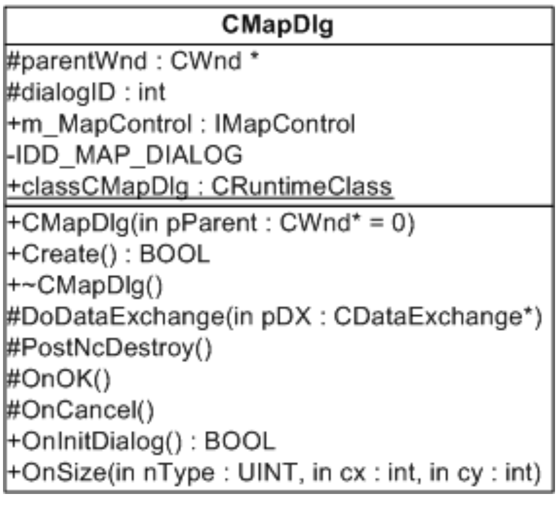
```
CAura
+CAura()
+InitInstance() : BOOL
#ExitInstance() : int
```

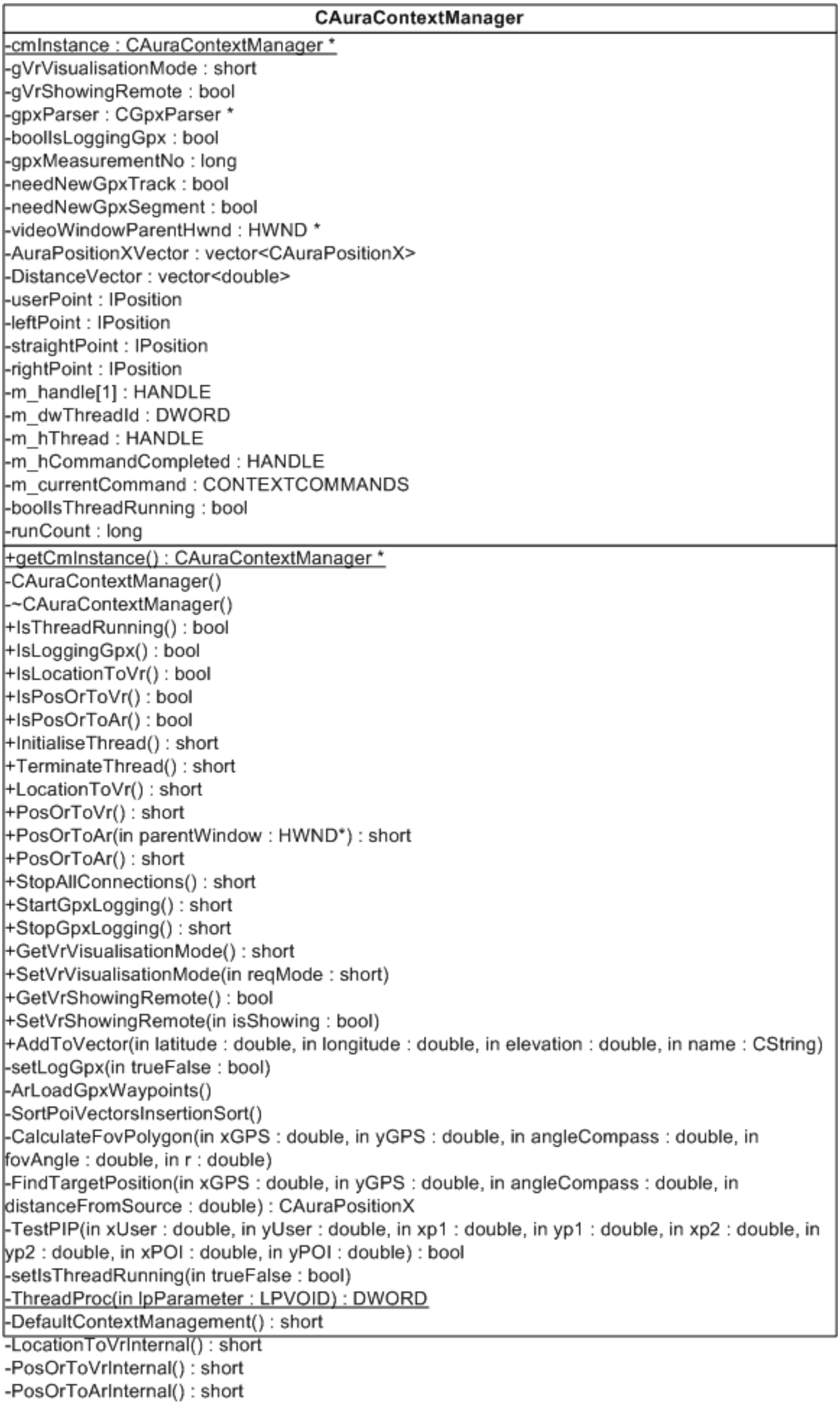


```
CAuraPropertyBag
-_refCount : ULONG
-pVar : VAR_LIST *
+CAuraPropertyBag()
+~CAuraPropertyBag()
+Read(in pszPropName : LPCOLESTR, in pVar : VARIANT*, in pErrorLog : IErrorLog*) :
HRESULT
+Write(in pszPropName : LPCOLESTR, in pVar : VARIANT*) : HRESULT
+AddRef() : ULONG
+Release() : ULONG
+QueryInterface(in riid : const IID &, in ppv : void**) : HRESULT
```


CArDlg
-mbiAR : SHMENUBARINFO
-auraMenuHwndAR : HWND
-auraMenuAR : HMENU
#parentWnd : CWnd *
#dialogID : int
+m_cameraSelect : CComboBox
-IDD_AR_DIALOG
+classCArDlg : CRuntimeClass
+CArDlg(in pParent : CWnd* = 0)
+Create() : BOOL
+~CArDlg()
#DoDataExchange(in pDX : CDataExchange*)
#PostNcDestroy()
+OnOK()
+OnCancel()
+OnInitDialog() : BOOL
+OnSize(in nType : UINT, in cx : int, in cy : int)
+OnInitMenuPopup(in pPopupMenu : CMenu*, in nIndex : UINT, in bSysMenu : BOOL)
+OnArMenuCameraEnable()
+OnArMenuCameraDisable()
+OnUpdateArMenuCameraEnable(in pCmdUI : CCmdUI*)
+OnUpdateArMenuCameraDisable(in pCmdUI : CCmdUI*)
+OnUpdateArMenuContextRunAR(in pCmdUI : CCmdUI*)
+OnArMenuContextRunAR()
+initializeMenuAR() : bool
+OnDirectShowMessage(in wParam : WPARAM, in lParam : LPARAM) : LRESULT


CAuraSocket
+CAuraSocket()
+~CAuraSocket()
+OnConnect(in nErrorCode : int)
+OnAccept(in nErrorCode : int)
+OnSend(in nErrorCode : int)
+OnReceive(in nErrorCode : int)
+OnClose(in nErrorCode : int)

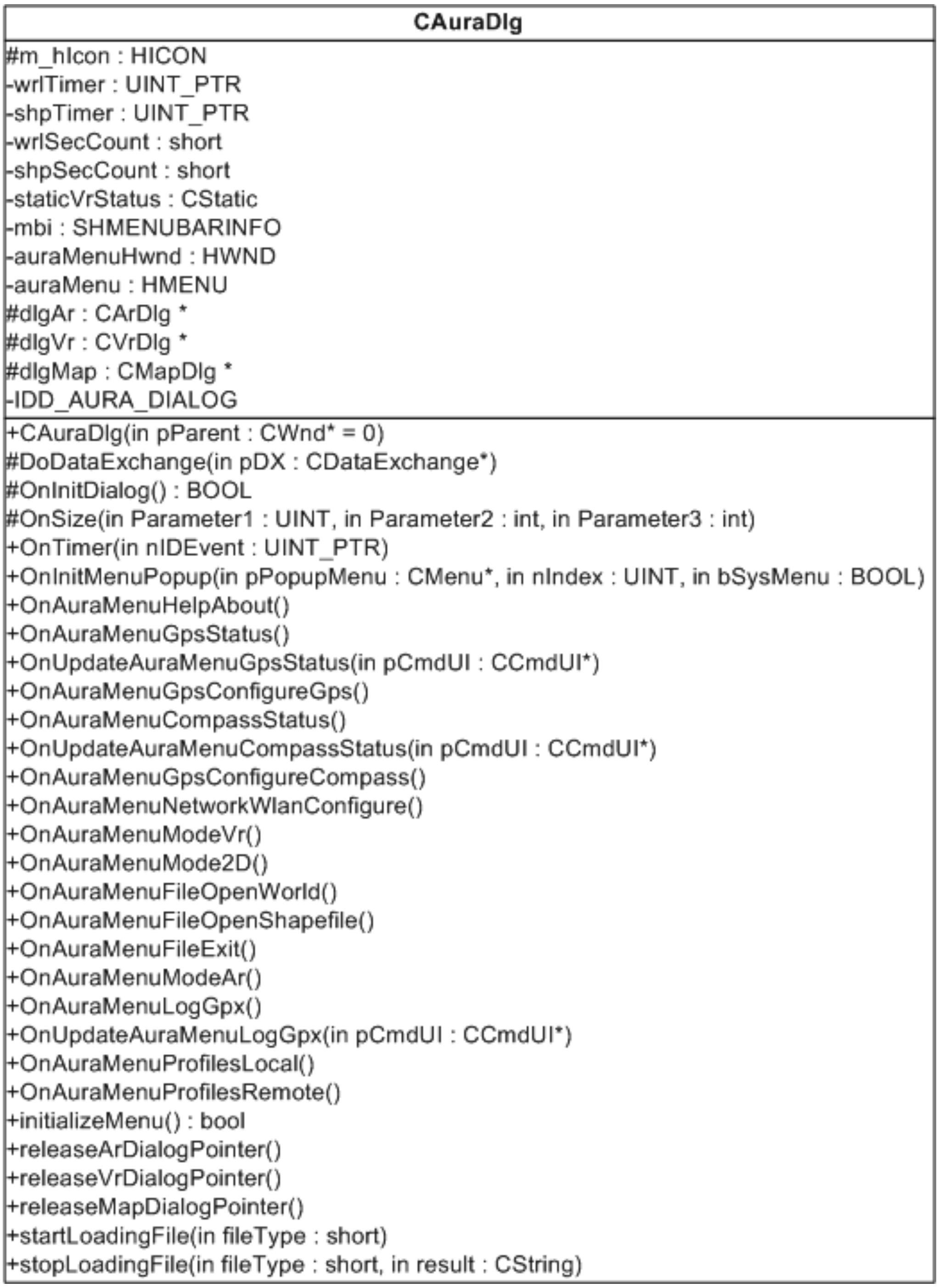

CMapDlg
#parentWnd : CWnd *
#dialogID : int
+m_MapControl : IMapControl
-IDD_MAP_DIALOG
+classCMapDlg : CRuntimeClass
+CMapDlg(in pParent : CWnd* = 0)
+Create() : BOOL
+~CMapDlg()
#DoDataExchange(in pDX : CDataExchange*)
#PostNcDestroy()
#OnOK()
#OnCancel()
+OnInitDialog() : BOOL
+OnSize(in nType : UINT, in cx : int, in cy : int)



CAuraContextManager

-cmInstance : CAuraContextManager *
-gVrVisualisationMode : short
-gVrShowingRemote : bool
-gpxParser : CGpxParser *
-boolIsLoggingGpx : bool
-gpxMeasurementNo : long
-needNewGpxTrack : bool
-needNewGpxSegment : bool
-videoWindowParentHwnd : HWND *
-AuraPositionXVector : vector<CAuraPositionX>
-DistanceVector : vector<double>
-userPoint : IPosition
-leftPoint : IPosition
-straightPoint : IPosition
-rightPoint : IPosition
-m_handle[1] : HANDLE
-m_dwThreadId : DWORD
-m_hThread : HANDLE
-m_hCommandCompleted : HANDLE
-m_currentCommand : CONTEXTCOMMANDS
-boolIsThreadRunning : bool
-runCount : long

+getCmInstance() : CAuraContextManager *
-CAuraContextManager()
-~CAuraContextManager()
+IsThreadRunning() : bool
+IsLoggingGpx() : bool
+IsLocationToVr() : bool
+IsPosOrToVr() : bool
+IsPosOrToAr() : bool
+InitialiseThread() : short
+TerminateThread() : short
+LocationToVr() : short
+PosOrToVr() : short
+PosOrToAr(in parentWindow : HWND*) : short
+PosOrToAr() : short
+StopAllConnections() : short
+StartGpxLogging() : short
+StopGpxLogging() : short
+GetVrVisualisationMode() : short
+SetVrVisualisationMode(in reqMode : short)
+GetVrShowingRemote() : bool
+SetVrShowingRemote(in isShowing : bool)
+AddToVector(in latitude : double, in longitude : double, in elevation : double, in name : CString)
-setLogGpx(in trueFalse : bool)
-ArLoadGpxWaypoints()
-SortPoiVectorsInsertionSort()
-CalculateFovPolygon(in xGPS : double, in yGPS : double, in angleCompass : double, in fovAngle : double, in r : double)
-FindTargetPosition(in xGPS : double, in yGPS : double, in angleCompass : double, in distanceFromSource : double) : CAuraPositionX
-TestPIP(in xUser : double, in yUser : double, in xp1 : double, in yp1 : double, in xp2 : double, in yp2 : double, in xPOI : double, in yPOI : double) : bool
-setIsThreadRunning(in trueFalse : bool)
-ThreadProc(in lpParameter : LPVOID) : DWORD
-DefaultContextManagement() : short
-LocationToVrInternal() : short
-PosOrToVrInternal() : short
-PosOrToArInternal() : short



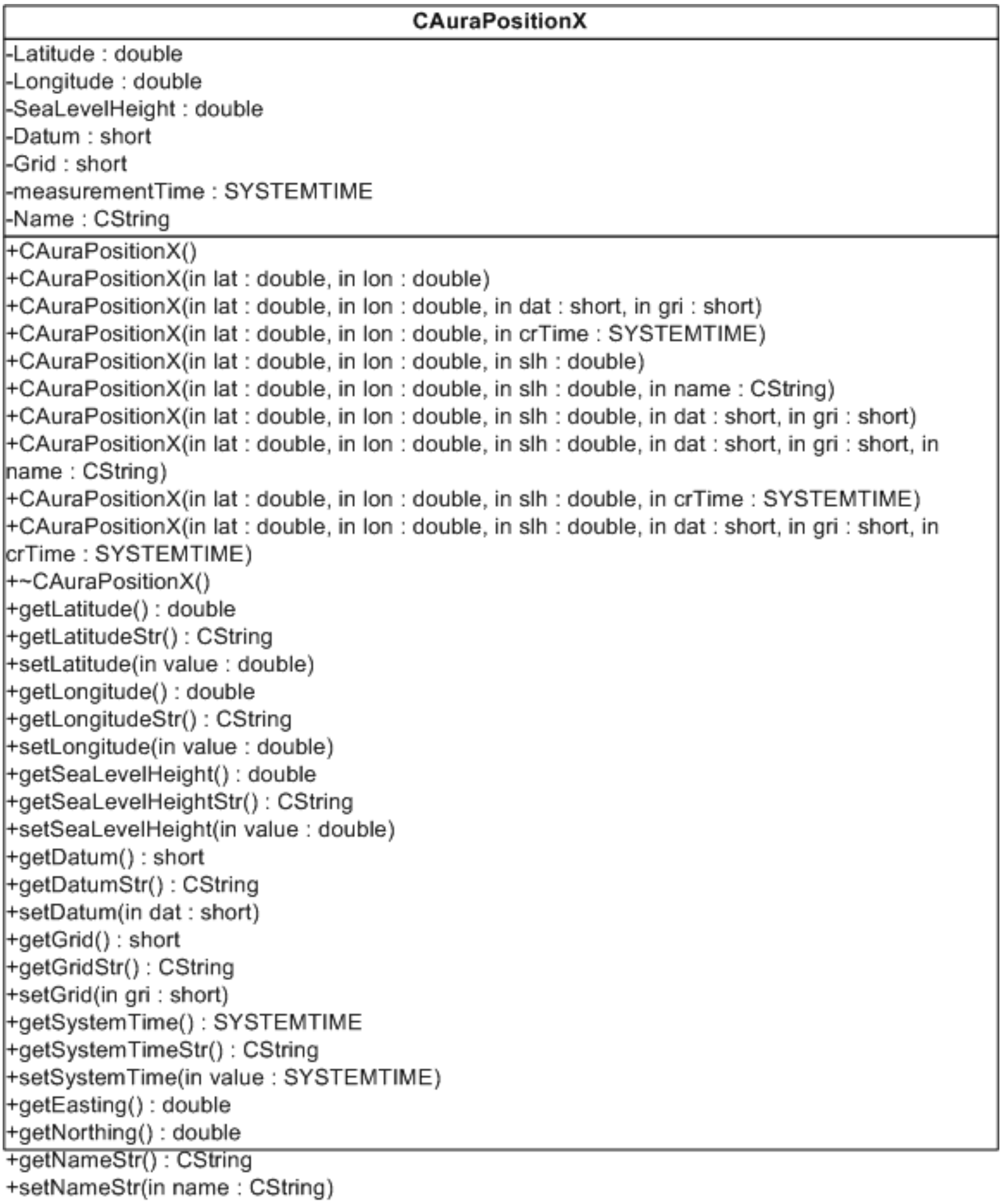
CAuraDlg

#m_hIcon : HICON
-wrlTimer : UINT_PTR
-shpTimer : UINT_PTR
-wrlSecCount : short
-shpSecCount : short
-staticVrStatus : CStatic
-mbi : SHMENUBARINFO
-auraMenuHwnd : HWND
-auraMenu : HMENU
#dlgAr : CArDlg *
#dlgVr : CVrDlg *
#dlgMap : CMapDlg *
-IDD_AURA_DIALOG

+CAuraDlg(in pParent : CWnd* = 0)
#DoDataExchange(in pDX : CDataExchange*)
#OnInitDialog() : BOOL
#OnSize(in Parameter1 : UINT, in Parameter2 : int, in Parameter3 : int)
+OnTimer(in nIDEvent : UINT_PTR)
+OnInitMenuPopup(in pPopupMenu : CMenu*, in nIndex : UINT, in bSysMenu : BOOL)
+OnAuraMenuHelpAbout()
+OnAuraMenuGpsStatus()
+OnUpdateAuraMenuGpsStatus(in pCmdUI : CCmdUI*)
+OnAuraMenuGpsConfigureGps()
+OnAuraMenuCompassStatus()
+OnUpdateAuraMenuCompassStatus(in pCmdUI : CCmdUI*)
+OnAuraMenuGpsConfigureCompass()
+OnAuraMenuNetworkWlanConfigure()
+OnAuraMenuModeVr()
+OnAuraMenuMode2D()
+OnAuraMenuFileOpenWorld()
+OnAuraMenuFileOpenShapefile()
+OnAuraMenuFileExit()
+OnAuraMenuModeAr()
+OnAuraMenuLogGpx()
+OnUpdateAuraMenuLogGpx(in pCmdUI : CCmdUI*)
+OnAuraMenuProfilesLocal()
+OnAuraMenuProfilesRemote()
+initializeMenu() : bool
+releaseArDialogPointer()
+releaseVrDialogPointer()
+releaseMapDialogPointer()
+startLoadingFile(in fileType : short)
+stopLoadingFile(in fileType : short, in result : CString)

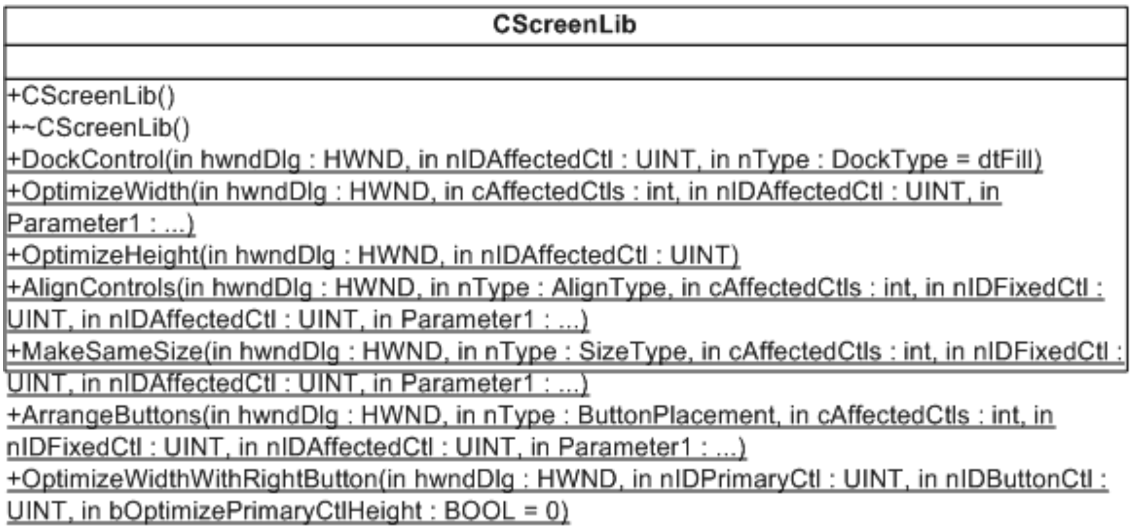
CAuraGraphManager
-gmInstance : CAuraGraphManager *
-boolIsThreadRunning : bool
-boolIsGraphBuilt : bool
-boolIsGraphCapturing : bool
-sourceFilterType : short
-rendererFilterType : short
-parentHwnd : HWND *
-m_handle[2] : HANDLE
-m_dwThreadId : DWORD
-m_hThread : HANDLE
-m_hCommandCompleted : HANDLE
-m_currentCommand : GRAPHCOMMANDS
-gCaptureGraphBuilder : CComPtr<ICaptureGraphBuilder2>
-gVideoCaptureBaseFilter : CComPtr<IBaseFilter>
-gVideoWindow : CComPtr<IVideoWindow>
-gVideoWindowHWnd : HWND
-m_pSampleLock : CCritSec
-centreMainStr : CString
-topCentreStrX : CString
-topCentreStrY : CString
-topLeftStrX : CString
-topLeftStrY : CString
-topRightStrX : CString
-topRightStrY : CString
-centreBottomStrX : CString
-centreBottomStrY : CString
+getGmInstance() : CAuraGraphManager *
+destroyGmInstance()
-CAuraGraphManager()
-~CAuraGraphManager()
+StartAuraGM(in hwnd : HWND*) : short
+StopAuraGM() : short
+IsThreadRunning() : bool
+IsGraphBuilt() : bool
+IsGraphCapturing() : bool
+InitialiseThread() : short
+TerminateThread() : short
+BuildCaptureGraph() : short
+RunCaptureGraph() : short
+CleanCaptureGraph() : short
+StartPreviewVideo() : short
+StopPreviewVideo() : short
+GetVideoWindowHandle() : HWND *
+SetStrings(in centreMainStr : CString, in topCentreTextX : CString, in topCentreTextY : CString, in topLeftTextX : CString, in topLeftTextY : CString, in topRightTextX : CString, in topRightTextY : CString, in centreBottomTextX : CString, in centreBottomTextY : CString)
-setIsThreadRunning(in trueFalse : bool)
-setIsGraphBuilt(in trueFalse : bool)
-setIsGraphCapturing(in trueFalse : bool)
-ThreadProc(in lpParameter : LPVOID) : DWORD
-ProcessInterfaceCommand() : HRESULT
-ProcessDirectShowEvent() : HRESULT
-NotifyMessage(in message : DSHOW_MESSAGE, in wzText : WCHAR*) : HRESULT
-CreateCaptureGraphInternal() : HRESULT
-RunCaptureGraphInternal() : HRESULT
-CleanCaptureGraphInternal() : HRESULT
-StartPreviewVideoInternal() : HRESULT
-StopPreviewVideoInternal() : HRESULT
-LoadCameraDriverToSourceFilter() : HRESULT
-ManageInputStreamForDirectShow() : HRESULT
-ManageInputStreamForHTC() : HRESULT
-ManageInputStreamForHtcTouchDiamond() : HRESULT
-ManageOutputStreamForDirectShow() : HRESULT
-ManageOutputStreamForHTC() : HRESULT
-ManageOutputStreamForHtcTouchDiamond() : HRESULT
-OnSampleProcessed(in pMediaSample : IMediaSample*, in pMediaType : AM_MEDIA_TYPE*)
-CopyFrame(in pSource : IMediaSample*, in pDest : IMediaSample*) : HRESULT
-ChangePixelColor(in imageData : BYTE*, in pixelPosX : int, in pixelPosY : int, in imageWidth : int, in imageHeight : int, in bytesPerPixel : int, in red : int, in green : int, in blue : int)
-ChangePixelsColor(in imageData : BYTE*, in pixelPosX : int, in pixelPosY : int, in imageWidth : int, in imageHeight : int, in bytesPerPixel : int, in red : int, in green : int, in blue : int)
-FlipImageVertical(in pImage : BYTE*, in width : int, in height : int, in bytesPerPixel : int)
-FlipImageHorizontal(in pImage : BYTE*, in width : int, in height : int, in bytesPerPixel : int)
-FlipImageRGB32(in pBuf : BYTE*, in width : int, in height : int, in flipImageH : bool, in flipImageV : bool)

-DrawGDICircle(in hdc : HDC, in radius : LONG, in ptCenter : POINT)

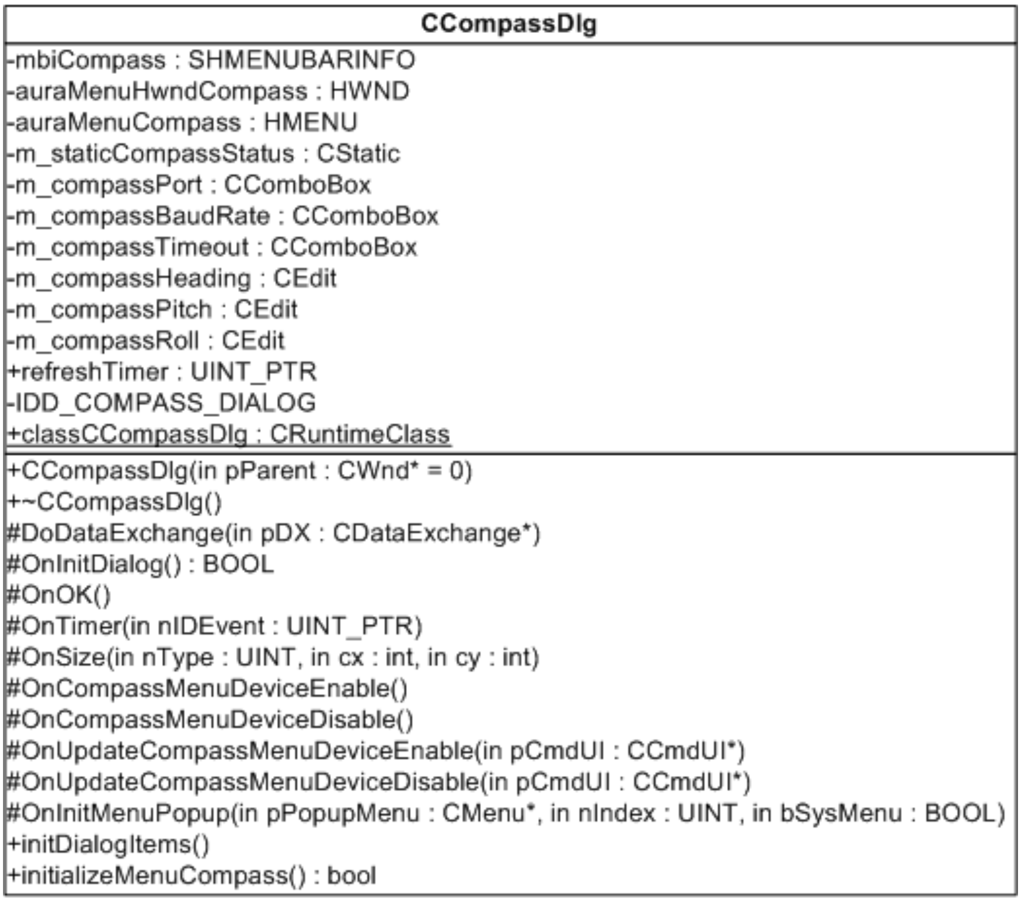


```
CAuraPositionX
-Latitude : double
-Longitude : double
-SeaLevelHeight : double
-Datum : short
-Grid : short
-measurementTime : SYSTEMTIME
-Name : CString
+CAuraPositionX()
+CAuraPositionX(in lat : double, in lon : double)
+CAuraPositionX(in lat : double, in lon : double, in dat : short, in gri : short)
+CAuraPositionX(in lat : double, in lon : double, in crTime : SYSTEMTIME)
+CAuraPositionX(in lat : double, in lon : double, in slh : double)
+CAuraPositionX(in lat : double, in lon : double, in slh : double, in name : CString)
+CAuraPositionX(in lat : double, in lon : double, in slh : double, in dat : short, in gri : short)
+CAuraPositionX(in lat : double, in lon : double, in slh : double, in dat : short, in gri : short, in name : CString)
+CAuraPositionX(in lat : double, in lon : double, in slh : double, in crTime : SYSTEMTIME)
+CAuraPositionX(in lat : double, in lon : double, in slh : double, in dat : short, in gri : short, in crTime : SYSTEMTIME)
+~CAuraPositionX()
+getLatitude() : double
+getLatitudeStr() : CString
+setLatitude(in value : double)
+getLongitude() : double
+getLongitudeStr() : CString
+setLongitude(in value : double)
+getSeaLevelHeight() : double
+getSeaLevelHeightStr() : CString
+setSeaLevelHeight(in value : double)
+getDatum() : short
+getDatumStr() : CString
+setDatum(in dat : short)
+getGrid() : short
+getGridStr() : CString
+setGrid(in gri : short)
+getSystemTime() : SYSTEMTIME
+getSystemTimeStr() : CString
+setSystemTime(in value : SYSTEMTIME)
+getEasting() : double
+getNorthing() : double
+getNameStr() : CString
+setNameStr(in name : CString)
```

```
CAuraRemoteProfile
-rpInstance : CAuraRemoteProfile *
-Name : CString
-Surname : CString
-Age : short
-AllowMainProfileProvision : bool
-AllowLocationProvision : bool
+PositionVector : vector<CAuraPositionX>
+getRpInstance() : CAuraRemoteProfile *
-CAuraRemoteProfile()
-~CAuraRemoteProfile()
+initProfile()
+getName() : CString
+setName(in value : CString)
+getSurname() : CString
+setSurname(in value : CString)
+getAge() : short
+getAgeString() : CString
+setAge(in value : short)
+getAllowMainProfileProvision() : bool
+setAllowMainProfileProvision(in value : bool)
+getAllowLocationProvision() : bool
+setAllowLocationProvision(in value : bool)
+getLatestLatitude() : double
+getLatestLatitudeStr() : CString
+getLatestLongitude() : double
+getLatestLongitudeStr() : CString
+getLatestSeaHeight() : double
+getLatestSeaHeightStr() : CString
+getLatestPositionUpdateTime() : SYSTEMTIME
+getLatestPositionUpdateTimeStr() : CString
+getLatestDatum() : short
+getLatestDatumStr() : CString
+getLatestGrid() : short
+getLatestGridStr() : CString
```



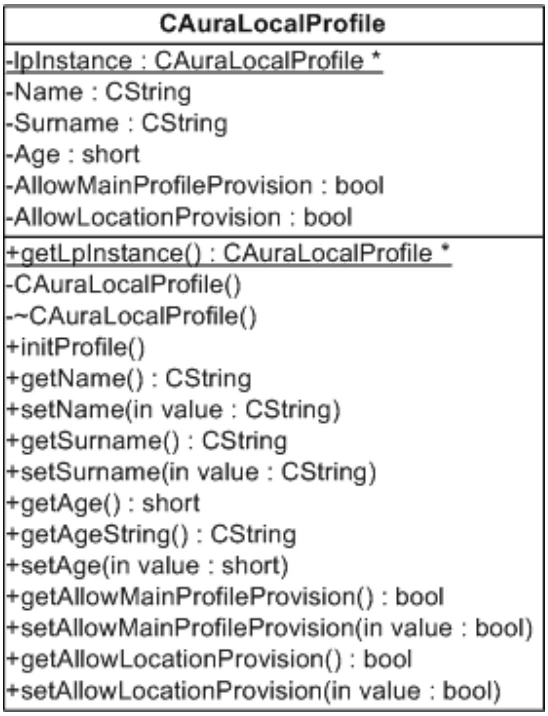
```
CScreenLib
+CScreenLib()
+~CScreenLib()
+DockControl(in hwndDlg : HWND, in nIDAffectedCtl : UINT, in nType : DockType = dtFill)
+OptimizeWidth(in hwndDlg : HWND, in cAffectedCtls : int, in nIDAffectedCtl : UINT, in
Parameter1 : ...)
+OptimizeHeight(in hwndDlg : HWND, in nIDAffectedCtl : UINT)
+AlignControls(in hwndDlg : HWND, in nType : AlignType, in cAffectedCtls : int, in nIDFixedCtl :
UINT, in nIDAffectedCtl : UINT, in Parameter1 : ...)
+MakeSameSize(in hwndDlg : HWND, in nType : SizeType, in cAffectedCtls : int, in nIDFixedCtl :
UINT, in nIDAffectedCtl : UINT, in Parameter1 : ...)
+ArrangeButtons(in hwndDlg : HWND, in nType : ButtonPlacement, in cAffectedCtls : int, in
nIDFixedCtl : UINT, in nIDAffectedCtl : UINT, in Parameter1 : ...)
+OptimizeWidthWithRightButton(in hwndDlg : HWND, in nIDPrimaryCtl : UINT, in nIDButtonCtl :
UINT, in bOptimizePrimaryCtlHeight : BOOL = 0)
```



CCompassDlg

-mbiCompass : SHMENUBARINFO
-auraMenuHwndCompass : HWND
-auraMenuCompass : HMENU
-m_staticCompassStatus : CStatic
-m_compassPort : CComboBox
-m_compassBaudRate : CComboBox
-m_compassTimeout : CComboBox
-m_compassHeading : CEdit
-m_compassPitch : CEdit
-m_compassRoll : CEdit
+refreshTimer : UINT_PTR
-IDD_COMPASS_DIALOG
+classCCompassDlg : CRuntimeClass

+CCompassDlg(in pParent : CWnd* = 0)
+~CCompassDlg()
#DoDataExchange(in pDX : CDataExchange*)
#OnInitDialog() : BOOL
#OnOK()
#OnTimer(in nIDEvent : UINT_PTR)
#OnSize(in nType : UINT, in cx : int, in cy : int)
#OnCompassMenuDeviceEnable()
#OnCompassMenuDeviceDisable()
#OnUpdateCompassMenuDeviceEnable(in pCmdUI : CCmdUI*)
#OnUpdateCompassMenuDeviceDisable(in pCmdUI : CCmdUI*)
#OnInitMenuPopup(in pPopupMenu : CMenu*, in nIndex : UINT, in bSysMenu : BOOL)
+initDialogItems()
+initializeMenuCompass() : bool



CAuraLocalProfile

-lpInstance : CAuraLocalProfile *
-Name : CString
-Surname : CString
-Age : short
-AllowMainProfileProvision : bool
-AllowLocationProvision : bool

+getLpInstance() : CAuraLocalProfile *
-CAuraLocalProfile()
-~CAuraLocalProfile()
+initProfile()
+getName() : CString
+setName(in value : CString)
+getSurname() : CString
+setSurname(in value : CString)
+getAge() : short
+getAgeString() : CString
+setAge(in value : short)
+getAllowMainProfileProvision() : bool
+setAllowMainProfileProvision(in value : bool)
+getAllowLocationProvision() : bool
+setAllowLocationProvision(in value : bool)

CCompassController

```
-compassInstance : CCompassController *
-compassParser : IPort
-gCompassStatus : COMPASSSTATUS
-m_cookie : DWORD
-compassParser_Created : bool
-compassParser_Advised : bool
-compassParser_Started : bool
-userPort : long
-userBaudrate : long
-userTimeout : long
-latestHeading : float
-latestPitch : float
-latestRoll : float
-csLatestData : CCriticalSection
-_interfaceEntries[] : AFX_INTERFACEMAP_ENTRY
#interfaceMap : AFX_INTERFACEMAP
#dispatchMap : AFX_DISPMAP
-_dwStockPropMask : DWORD
-_dispatchEntryCount : UINT
-_dispatchEntries[] : AFX_DISPMAP_ENTRY
+classCCompassController : CRuntimeClass
```

```
+getCompassInstance() : CCompassController *
-CCompassController()
-~CCompassController()
-OnFinalRelease()
+GetCompassStatus(in statusString : CString*) : short
+StartCompassWithDefaultValues() : short
+StartCompassWithCustomValues(in port : long, in baudrate : long, in timeout : long) : short
+StopCompassMeasurements() : short
+RetrieveComParams(in reqPort : long*, in reqBaudrate : long*, in reqTimeout : long*) : short
+RetrieveData(in reqHeading : float*, in reqPitch : float*, in reqRoll : float*) : short
+RetrieveEastNorth(in reqEasting : double*, in reqNorthing : double*, in reqSeaLevelAltitude :
double*, in reqOverEllipsoidAltitude : double*, in reqDatum : short*, in reqGrid : short*, in
reqZone : CString*) : short
+RetrieveSatellites(in itemNo : short, in reqID : int*, in reqSNR : int*, in reqElevation : int*, in
reqAzimuth : int*, in reqUsedForFix : bool*) : short
+RetrieveSpeedHeading(in speedType : short, in reqSpeed : double*, in reqHeading : double*, in
reqMagVar : double*) : short
-setCompassParserCreated(in trueFalse : bool)
-setCompassParserAdvised(in trueFalse : bool)
-setCompassParserStarted(in trueFalse : bool)
-isCompassParserCreated() : bool
-isCompassParserAdvised() : bool
-isCompassParserStarted() : bool
-checkSerialToolsLicense() : HRESULT
-createCompassParser() : HRESULT
-destroyCompassParser() : HRESULT
-initCompassParser() : HRESULT
-startCompassParser() : HRESULT
-stopCompassParser() : HRESULT
-OnRead(in Data : VARIANT*)
-OnDSR(in Value : VARIANT &)
-OnCTS(in Value : VARIANT &)
-OnRI(in Value : VARIANT &)
-OnDCD(in Value : VARIANT &)
-OnForceClose(in ErrorCode : VARIANT &)
-OnWritten(in BytesWritten : VARIANT &)
```

CGpsController

```
-gpsInstance : CGpsController *
-gpsParser : INmeaParser
-gGpsStatus : GPSSTATUS
-m_cookie : DWORD
-gpsParser_Created : bool
-gpsParser_Advised : bool
-gpsParser_Started : bool
-actualPort : short
-actualFixType : short
-actualBaudrate : long
-actualTimeout : long
-userPort : short
-userDatum : short
-userGrid : short
-userBaudrate : long
-userTimeout : long
-gLatestPosition : IPosition
-gLatestSatellites : ISatellites
-gLatestMovement : IMovement
-csComStatus : CCriticalSection
-csLatestPosition : CCriticalSection
-csLatestSatellites : CCriticalSection
-csLatestMovement : CCriticalSection
#interfaceMap : AFX_INTERFACEMAP
-_interfaceEntries[] : AFX_INTERFACEMAP_ENTRY
-_dispatchEntries[] : AFX_DISPMAP_ENTRY
#dispatchMap : AFX_DISPMAP
-_dwStockPropMask : DWORD
-_dispatchEntryCount : UINT
+classCGpsController : CRuntimeClass
```

```
+getGpsInstance() : CGpsController *
-CGpsController()
-~CGpsController()
-OnFinalRelease()
+GetGpsStatus(in statusString : CString*) : short
+StartGpsWithDefaultValues() : short
+StartGpsWithCustomValues(in port : short, in baudrate : long, in datum : short, in grid : short, in timeout : long) : short
+StopGpsMeasurements() : short
+RetrieveComStatus(in reqPort : short*, in reqBaudrate : long*, in reqTimeout : long*) : short
+RetrieveLatLon(in reqLatitude : double*, in reqLongitude : double*, in reqSeaLevelAltitude : double*, in reqOverEllipsoidAltitude : double*, in reqDatum : short*, in reqGrid : short*) : short
+RetrieveEastNorth(in reqEasting : double*, in reqNorthing : double*, in reqSeaLevelAltitude : double*, in reqOverEllipsoidAltitude : double*, in reqDatum : short*, in reqGrid : short*, in reqZone : CString*) : short
+RetrieveSatellites(in itemNo : short, in reqID : int*, in reqSNR : int*, in reqElevation : int*, in reqAzimuth : int*, in reqUsedForFix : bool*) : short
+RetrieveSpeedHeading(in speedType : short, in reqSpeed : double*, in reqHeading : double*, in reqMagVar : double*) : short
-setGpsParserCreated(in trueFalse : bool)
-setGpsParserAdvised(in trueFalse : bool)
-setGpsParserStarted(in trueFalse : bool)
-isGpsParserCreated() : bool
-isGpsParserAdvised() : bool
-isGpsParserStarted() : bool
-checkGpsToolsLicense() : HRESULT
-createGpsParser() : HRESULT
-destroyGpsParser() : HRESULT
-initGpsParser() : HRESULT
-startGpsParser() : HRESULT
-stopGpsParser() : HRESULT
-OnGeneric(in objSentence : VARIANT &)
-OnGGA(in objGGA : VARIANT &, in objSentence : VARIANT &)
-OnGLL(in objGLL : VARIANT &, in objSentence : VARIANT &)
-OnRMC(in objRMC : VARIANT &, in objSentence : VARIANT &)
-OnSatellites(in objSatellites : VARIANT &)
-OnGpsFix(in objGpsFix : VARIANT &)
-OnMovement(in objMovement : VARIANT &)
-OnQuality(in objQuality : VARIANT &)
-OnComStatus(in objComStatus : VARIANT &)
```

CGpsDlg

-mbiGps : SHMENUBARINFO
-auraMenuHwndGps : HWND
-auraMenuGps : HMENU
-m_staticGpsStatus : CStatic
-m_gpsSatellites : CListBox
-m_gpsPort : CComboBox
-m_gpsBaudRate : CComboBox
-m_gpsDatum : CComboBox
-m_gpsGrid : CComboBox
-m_gpsTimeout : CComboBox
-m_gpsLatitude : CEdit
-m_gpsLongitude : CEdit
-m_gpsEasting : CEdit
-m_gpsNorthing : CEdit
-m_gpsZone : CEdit
-m_gpsHeading : CEdit
-m_gpsSeaLevel : CEdit
+refreshTimer : UINT_PTR
-IDD_GPS_DIALOG
+classCGpsDlg : CRuntimeClass

+CGpsDlg(in pParent : CWnd* = 0)
+~CGpsDlg()
#DoDataExchange(in pDX : CDataExchange*)
#OnInitDialog() : BOOL
#OnOK()
#OnTimer(in nIDEvent : UINT_PTR)
#OnSize(in nType : UINT, in cx : int, in cy : int)
#OnInitMenuPopup(in pPopupMenu : CMenu*, in nIndex : UINT, in bSysMenu : BOOL)
#OnGpsMenuDeviceEnable()
#OnGpsMenuDeviceDisable()
#OnUpdateGpsMenuDeviceEnable(in pCmdUI : CCmdUI*)
#OnUpdateGpsMenuDeviceDisable(in pCmdUI : CCmdUI*)
#OnGpsMenuOptionsSatellites()
+initDialogItems()
+initializeMenuGps() : bool

CProfileDlg

-manageLocalProfile : bool
-m_profileName : CEdit
-m_profileSurname : CEdit
-m_profileAge : CEdit
-m_allowMainProfileProvision : CButton
-m_allowLocationProvision : CButton
-m_latestLat : CEdit
-m_latestLon : CEdit
-m_latestTime : CEdit
-m_latestSeaHeight : CEdit
-m_latestDat : CEdit
-m_latestGri : CEdit
-IDD_PROFILE_DIALOG
+classCProfileDlg : CRuntimeClass

+CProfileDlg(in pParent : CWnd* = 0)
+~CProfileDlg()
#DoDataExchange(in pDX : CDataExchange*)
#OnInitDialog() : BOOL
#OnOK()
-readProfileVariables() : short
-writeProfileVariables() : short
+profileToManage(in isLocalProfile : bool)

CGpxParser

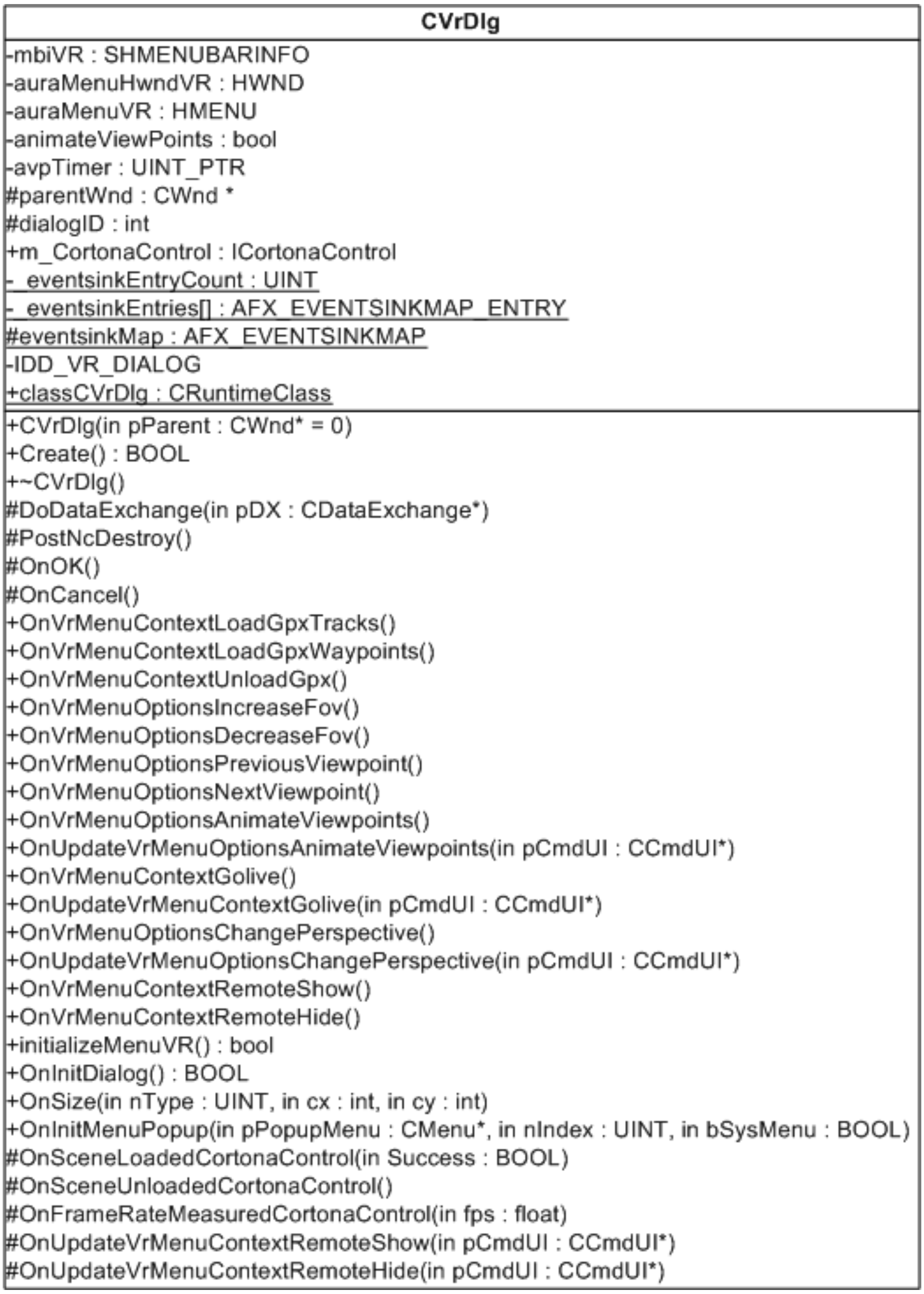
-startTime : SYSTEMTIME
-gpxDocument : CComPtr<IXMLDOMDocument>
-xmlNode : CComPtr<IXMLDOMNode>
-gpxNode : CComPtr<IXMLDOMNode>
-gpxTrack : CComPtr<IXMLDOMNode>
-gpxTrackSegment : CComPtr<IXMLDOMNode>
-gpxTrackPoint : CComPtr<IXMLDOMNode>

+CGpxParser()
+CGpxParser(in st : SYSTEMTIME*)
+~CGpxParser()
+CreateGpxDocument() : CComPtr<IXMLDOMDocument>
+CreateXmlHeader() : CComPtr<IXMLDOMNode>
+CreateRootGpx() : CComPtr<IXMLDOMNode>
+CreateGpxTrack(in trackName : CString) : CComPtr<IXMLDOMNode>
+CreateGpxTrackSegment() : CComPtr<IXMLDOMNode>
+CreateGpxTrackPoint(in latitude : double, in longitude : double, in elevation : double, in magvar : double) : CComPtr<IXMLDOMNode>
+SaveGpxFile() : bool
+LoadGpxTracks(in thePath : CString) : bool
+LoadGpxWayPoints(in thePath : CString) : bool
+LoadGpxWayPointsAR(in thePath : CString) : long

CMapController

-mapInstance : CMapController *
-shapefileSuccessfullyLoaded : bool
-shapefilePathAndFileName : CString
-theMapControl : IMapControl *
-shFile : IShapeFile
-msFile : IMapShapeFile
-rotationScale : double
-zoomScale : double
-ZOOM_MIN : double
-ZOOM_MAX : double

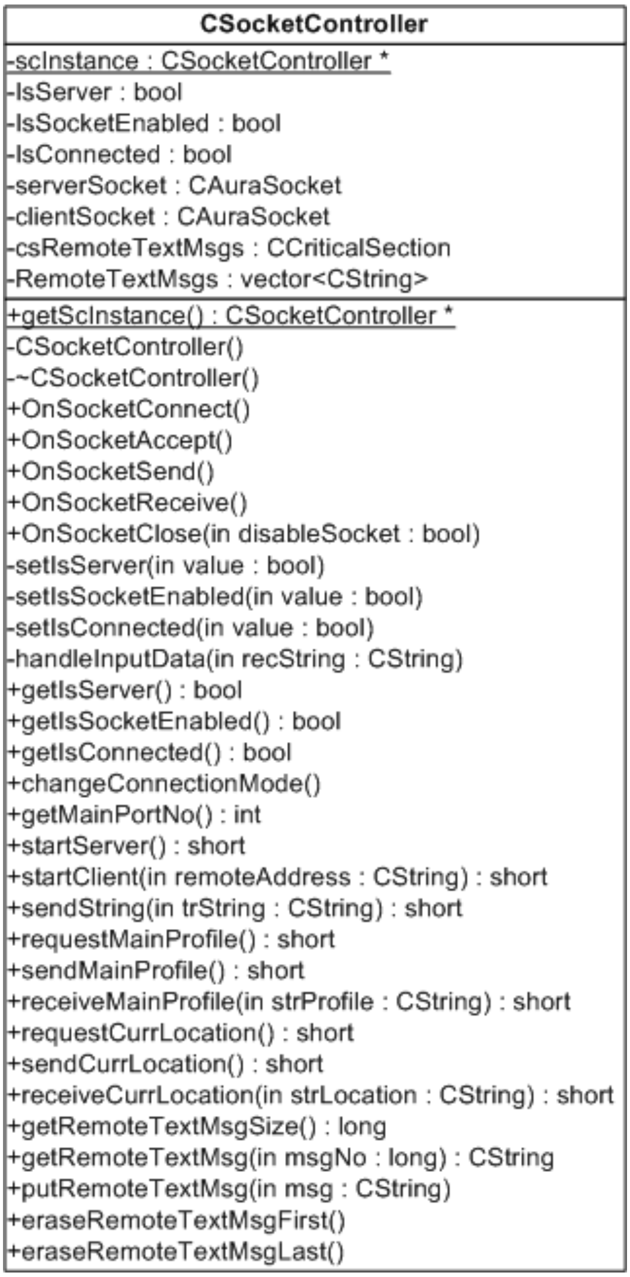
+getMapInstance() : CMapController *
+getMapInstance(in mapControlPointer : IMapControl*) : CMapController *
-CMapController()
-CMapController(in mapControlPointer : IMapControl*)
-~CMapController()
+IsShapefileSuccessfullyLoaded() : bool
+LoadShapefileIntoMapControl(in pathAndFileName : CString*) : bool
+InitializeMapEngine() : short
+TerminateMapEngine() : short
+MapZoomOut() : short
+MapZoomIn() : short
+RotateMapCounterClockwise() : short
+RotateMapClockwise() : short
+CentreMapOnPointBasedOnDatum(in longitude : double, in latitude : double, in datum : short) : short
+CentreMapOnPointBasedOnGrid(in easting : double, in northing : double, in datum : short, in grid : short) : short
-setShapefileSuccessfullyLoaded(in trueFalse : bool)
-checkGpsToolsLicense() : HRESULT



CVrDlg

-mbiVR : SHMENUBARINFO
-auraMenuHwndVR : HWND
-auraMenuVR : HMENU
-animateViewPoints : bool
-avpTimer : UINT_PTR
#parentWnd : CWnd *
#dialogID : int
+m_CortonaControl : ICortonaControl
-_eventsinkEntryCount : UINT
-_eventsinkEntries[] : AFX_EVENTSINKMAP_ENTRY
#eventsinkMap : AFX_EVENTSINKMAP
-IDD_VR_DIALOG
+classCVrDlg : CRuntimeClass

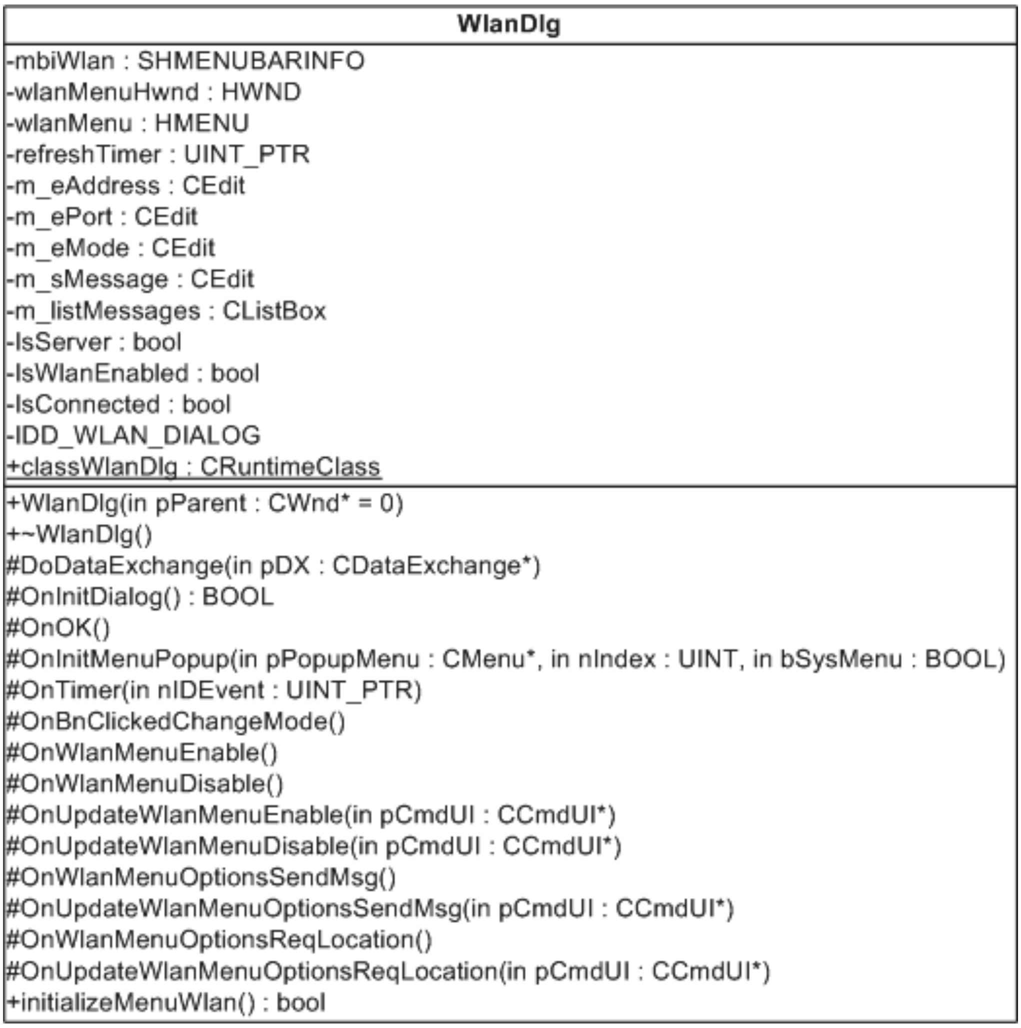
+CVrDlg(in pParent : CWnd* = 0)
+Create() : BOOL
+~CVrDlg()
#DoDataExchange(in pDX : CDataExchange*)
#PostNcDestroy()
#OnOK()
#OnCancel()
+OnVrMenuContextLoadGpxTracks()
+OnVrMenuContextLoadGpxWaypoints()
+OnVrMenuContextUnloadGpx()
+OnVrMenuOptionsIncreaseFov()
+OnVrMenuOptionsDecreaseFov()
+OnVrMenuOptionsPreviousViewpoint()
+OnVrMenuOptionsNextViewpoint()
+OnVrMenuOptionsAnimateViewpoints()
+OnUpdateVrMenuOptionsAnimateViewpoints(in pCmdUI : CCmdUI*)
+OnVrMenuContextGolive()
+OnUpdateVrMenuContextGolive(in pCmdUI : CCmdUI*)
+OnVrMenuOptionsChangePerspective()
+OnUpdateVrMenuOptionsChangePerspective(in pCmdUI : CCmdUI*)
+OnVrMenuContextRemoteShow()
+OnVrMenuContextRemoteHide()
+initializeMenuVR() : bool
+OnInitDialog() : BOOL
+OnSize(in nType : UINT, in cx : int, in cy : int)
+OnInitMenuPopup(in pPopupMenu : CMenu*, in nIndex : UINT, in bSysMenu : BOOL)
#OnSceneLoadedCortonaControl(in Success : BOOL)
#OnSceneUnloadedCortonaControl()
#OnFrameRateMeasuredCortonaControl(in fps : float)
#OnUpdateVrMenuContextRemoteShow(in pCmdUI : CCmdUI*)
#OnUpdateVrMenuContextRemoteHide(in pCmdUI : CCmdUI*)


CSocketController
-scInstance : CSocketController *
-IsServer : bool
-IsSocketEnabled : bool
-IsConnected : bool
-serverSocket : CAuraSocket
-clientSocket : CAuraSocket
-csRemoteTextMsgs : CCriticalSection
-RemoteTextMsgs : vector<CString>
+getScInstance() : CSocketController *
-CSocketController()
-~CSocketController()
+OnSocketConnect()
+OnSocketAccept()
+OnSocketSend()
+OnSocketReceive()
+OnSocketClose(in disableSocket : bool)
-setIsServer(in value : bool)
-setIsSocketEnabled(in value : bool)
-setIsConnected(in value : bool)
-handleInputData(in recString : CString)
+getIsServer() : bool
+getIsSocketEnabled() : bool
+getIsConnected() : bool
+changeConnectionMode()
+getMainPortNo() : int
+startServer() : short
+startClient(in remoteAddress : CString) : short
+sendString(in trString : CString) : short
+requestMainProfile() : short
+sendMainProfile() : short
+receiveMainProfile(in strProfile : CString) : short
+requestCurrLocation() : short
+sendCurrLocation() : short
+receiveCurrLocation(in strLocation : CString) : short
+getRemoteTextMsgSize() : long
+getRemoteTextMsg(in msgNo : long) : CString
+putRemoteTextMsg(in msg : CString)
+eraseRemoteTextMsgFirst()
+eraseRemoteTextMsgLast()

CVrController

-vrInstance : CVrController *
-theCortonaControl : ICortonaControl *
-theEngine : IEngine *
-the3DViewService : I3DViewService2 *
-sceneSuccessfullyLoaded : bool
-scenePathAndFileName : CString
-loadedModelType : MODELTYPE
-initRootNodeCount : long
+classCVrController : CRuntimeClass

+getVrInstance() : CVrController *
+getVrInstance(in cortonaControlPointer : ICortonaControl*) : CVrController *
+OnFinalRelease()
-CVrController()
-CVrController(in cortonaControlPointer : ICortonaControl*)
-~CVrController()
+IsSceneSuccessfullyLoaded() : bool
+LoadSceneIntoCortControl(in pathAndFileName : CString*) : bool
+InitialiseEngine() : short
+TerminateEngine() : short
+EnableControl()
+DisableControl()
+StartAddingStuff() : short
+StopAddingStuff() : short
+MoveCamera(in xEasting : float, in yNorthing : float, in zAltitude : float) : short
+RotateCamera(in inPitch : float, in inHeading : float, in inRoll : float, in realHeight : bool) : short
+AddCone(in xEasting : float, in yNorthing : float, in zAltitude : float) : short
+RemoveCones() : short
+AddSphere(in xEasting : float, in yNorthing : float, in zAltitude : float) : short
+RemoveSpheres() : short
+AddViewPoint(in xEasting : float, in yNorthing : float, in zAltitude : float, in Heading : float, in vpName : CString) : short
+AnimateViewPoints() : short
+RemoveViewPoints() : short
+IncreaseFov() : short
+DecreaseFov() : short
+MoveToPreviousViewPoint() : short
+MoveToNextViewPoint() : short
+AddLineBetween2Points(in xEasting1 : float, in yNorthing1 : float, in zAltitude1 : float, in xEasting2 : float, in yNorthing2 : float, in zAltitude2 : float) : short
+RemoveLinesBetween2Points() : short
+AddExahedronBetween2Points(in xEasting1 : float, in yNorthing1 : float, in zAltitude1 : float, in xEasting2 : float, in yNorthing2 : float, in zAltitude2 : float) : short
+RemoveExahedronBetween2Points() : short
+AddSignPost(in xEasting : float, in yNorthing : float, in zAltitude : float, in theText : CString) : short
+RemoveSignPosts() : short
+AddRotatingText(in xEasting : float, in yNorthing : float, in zAltitude : float, in theText : CString) : short
+RemoveRotatingText() : short
+AddFog(in visibilityRange : int) : short
+AddSubject(in xEasting : float, in yNorthing : float, in zAltitude : float, in subjectNo : short) : short
+MoveSubject(in xEasting : float, in yNorthing : float, in zAltitude : float, in subjectNo : short) : short
+RemoveSubject(in subjectNo : short) : short
-setupControl()
-getRootNodeCount() : long
-getLoadedModelType() : short
-setSceneSuccessfullyLoaded(in trueFalse : bool)



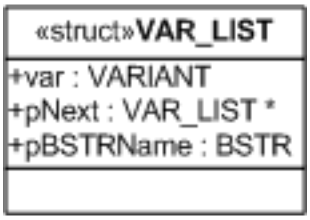
WlanDlg

-mbiWlan : SHMENUBARINFO
-wlanMenuHwnd : HWND
-wlanMenu : HMENU
-refreshTimer : UINT_PTR
-m_eAddress : CEdit
-m_ePort : CEdit
-m_eMode : CEdit
-m_sMessage : CEdit
-m_listMessages : CListBox
-IsServer : bool
-IsWlanEnabled : bool
-IsConnected : bool
-IDD_WLAN_DIALOG
+classWlanDlg : CRuntimeClass

+WlanDlg(in pParent : CWnd* = 0)
+~WlanDlg()
#DoDataExchange(in pDX : CDataExchange*)
#OnInitDialog() : BOOL
#OnOK()
#OnInitMenuPopup(in pPopupMenu : CMenu*, in nIndex : UINT, in bSysMenu : BOOL)
#OnTimer(in nIDEvent : UINT_PTR)
#OnBnClickedChangeMode()
#OnWlanMenuEnable()
#OnWlanMenuDisable()
#OnUpdateWlanMenuEnable(in pCmdUI : CCmdUI*)
#OnUpdateWlanMenuDisable(in pCmdUI : CCmdUI*)
#OnWlanMenuOptionsSendMsg()
#OnUpdateWlanMenuOptionsSendMsg(in pCmdUI : CCmdUI*)
#OnWlanMenuOptionsReqLocation()
#OnUpdateWlanMenuOptionsReqLocation(in pCmdUI : CCmdUI*)
+initializeMenuWlan() : bool

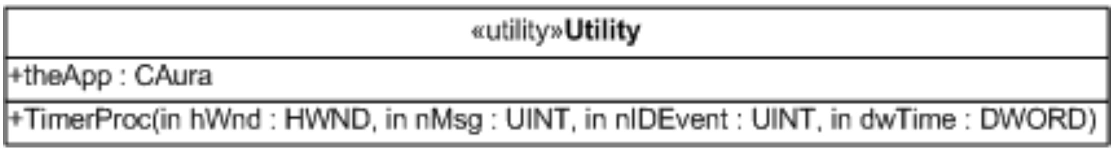
12.6 Appendix VI – Aura's Utility Structures

In this Appendix, the reader can view a representation of the 10 utility structures and enumeration data types that have been employed for the development of the system.



«struct»**VAR_LIST**

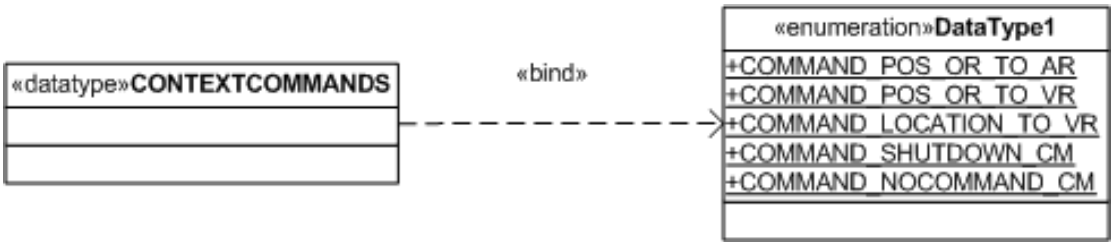
+var : VARIANT
+pNext : VAR_LIST *
+pBSTRName : BSTR

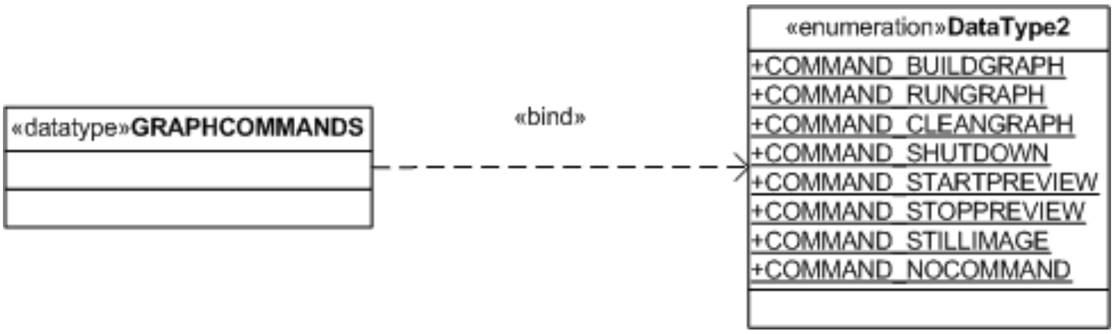


«utility»**Utility**

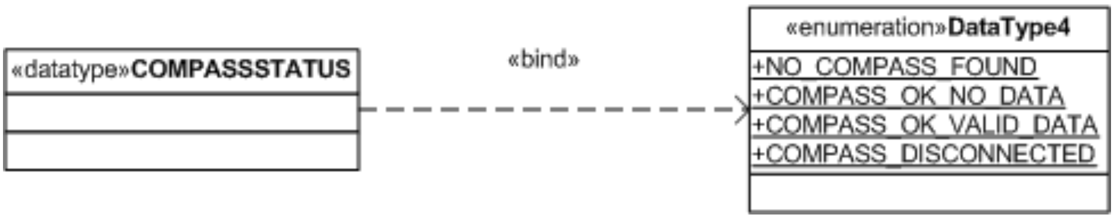
+theApp : CAura

+TimerProc(in hWnd : HWND, in nMsg : UINT, in nIDEvent : UINT, in dwTime : DWORD)

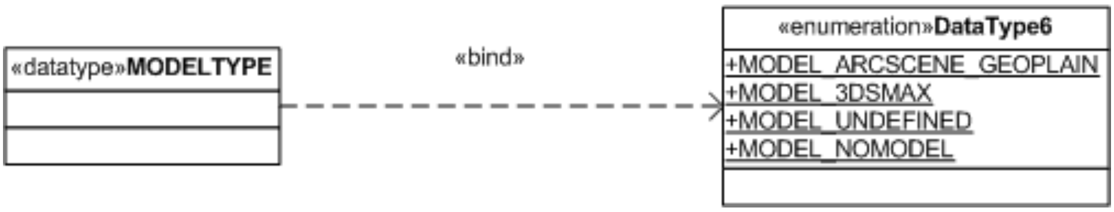

«datatype»CONTEXTCOMMANDS
«bind»
«enumeration»DataType1
+COMMAND_POS_OR_TO_AR
+COMMAND_POS_OR_TO_VR
+COMMAND_LOCATION_TO_VR
+COMMAND_SHUTDOWN_CM
+COMMAND_NOCOMMAND_CM


«datatype»GRAPHCOMMANDS
«bind»
«enumeration»DataType2
+COMMAND_BUILDGRAPH
+COMMAND_RUNGRAPH
+COMMAND_CLEANGRAPH
+COMMAND_SHUTDOWN
+COMMAND_STARTPREVIEW
+COMMAND_STOPPREVIEW
+COMMAND_STILLIMAGE
+COMMAND_NOCOMMAND

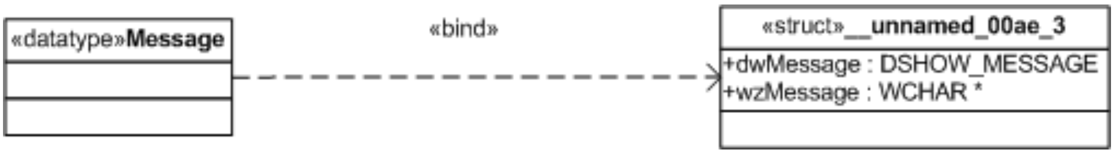

«datatype»DSHOW_MESSAGE
«bind»
«enumeration»DataType3
+MESSAGE_INFO
+MESSAGE_ERROR


«datatype»COMPASSSTATUS
«bind»
«enumeration»DataType4
+NO_COMPASS_FOUND
+COMPASS_OK_NO_DATA
+COMPASS_OK_VALID_DATA
+COMPASS_DISCONNECTED


«datatype»GPSSTATUS
«bind»
«enumeration»DataType5
+NO_GPS_FOUND
+GPS_OK_NO_FIX
+GPS_OK_FIX_OK
+GPS_OK_OLD_FIX
+GPS_DISCONNECTED


«datatype»MODELTYPE
«bind»
«enumeration»DataType6
+MODEL_ARCSCENE_GEOPLAIN
+MODEL_3DSMAX
+MODEL_UNDEFINED
+MODEL_NOMODEL


«datatype»PoiVectorStruct
«bind»
«struct»__unnamed_0095_2
+currentDistance : double
+vectorID : int



12.7 Appendix VII – Field-of-View Algorithm Implementation

This Appendix demonstrates the source code for calculating the Field-of-View polygon coordinates.

```
//Solution for all angles
#define PI
3.1415926535897932384626433832795028841971693993751058209749445923078164

//TESTING Variables
CString outputString;
//TESTING Variables

//Circle - Line intersection point coordinates
double xB1, xB2, xC1, xC2, xD1, xD2, yB1, yB2, yC1, yC2, yD1,
yD2;
//Input variables
double xGPS, yGPS;                    //metres;
double angleCompass;                  //degrees
//Constant variables
double fovAngle;                      //degrees
double r;                             //metres, length of
sides AB and AC, circle radius

//Intermediate Line variables
double w, w1, w2;                //degrees
double wTan, w1Tan, w2Tan;       //tan of degrees
double c, c1, c2;                //Y-axis intercept of
y=m*x+c


if (angleCompass == 90) angleCompass = 89.999999; //if
angleCompass = 90, then tan(angleCompass)=OO
if (angleCompass == 270) angleCompass = 269.999999; //if
angleCompass = 90, then tan(angleCompass)=OO


//Computation for Line XD
w = - (angleCompass - 90); //degrees
wTan = tan(w*PI/180);      //Conversion from radians to
degrees
c = yGPS - wTan*xGPS;      //metres, by using y=m*x+c

//Computation for Line YB
w1 = w + fovAngle/2;       //degrees
w1Tan = tan(w1*PI/180);    //Conversion from radians to
```

```
degrees
c1 = yGPS - w1Tan*xGPS;     //metres, by using y=m*x+c

//Computation for Line ZC
w2 = w - fovAngle/2;        //degrees
w2Tan = tan(w2*PI/180);     //Conversion from radians to
degrees
c2 = yGPS - w2Tan*xGPS;     //metres, by using y=m*x+c


//Intermediate Circle variables
double sub1, sub2, sub3;

//Substituting for line YB
sub1 = 1. + pow(wTan,2);
sub2 = 2.*wTan*c - 2.*wTan*yGPS - 2.*xGPS;
sub3 = pow(xGPS,2) + pow(yGPS,2) + pow(c,2) - 2.*c*yGPS -
pow(r,2);

//Results point coordinates of Line & Circle intersection
xD1 = ( -sub2 + sqrt( pow(sub2,2) - 4.*sub1*sub3) ) /
(2.*sub1);
xD2 = ( -sub2 - sqrt( pow(sub2,2) - 4.*sub1*sub3) ) /
(2.*sub1);
yD1 = wTan*xD1 + c;
yD2 = wTan*xD2 + c;

//Substituting for line YB
sub1 = 1. + pow(w1Tan,2);
sub2 = 2.*w1Tan*c1 - 2.*w1Tan*yGPS - 2.*xGPS;
sub3 = pow(xGPS,2) + pow(yGPS,2) + pow(c1,2) - 2.*c1*yGPS -
pow(r,2);

//Results point coordinates of Line & Circle intersection
xB1 = ( -sub2 + sqrt( pow(sub2,2) - 4.*sub1*sub3) ) /
(2.*sub1);
xB2 = ( -sub2 - sqrt( pow(sub2,2) - 4.*sub1*sub3) ) /
(2.*sub1);
yB1 = w1Tan*xB1 + c1;
yB2 = w1Tan*xB2 + c1;

//Substituting for line ZC
sub1 = 1. + pow(w2Tan,2);
sub2 = 2.*w2Tan*c2 - 2.*w2Tan*yGPS - 2.*xGPS;
sub3 = pow(xGPS,2) + pow(yGPS,2) + pow(c2,2) - 2.*c2*yGPS -
pow(r,2);

//Results point coordinates of Line & Circle intersection
xC1 = ( -sub2 + sqrt( pow(sub2,2) - 4.*sub1*sub3) ) /
(2.*sub1);
xC2 = ( -sub2 - sqrt( pow(sub2,2) - 4.*sub1*sub3) ) /
(2.*sub1);
yC1 = w2Tan*xC1 + c2;
yC2 = w2Tan*xC2 + c2;


if (0<=angleCompass && angleCompass<90){
     outputString.Format(L"xD1 = %f\nyD1 = %f", xD1, yD1 );
```

```
     MessageBox (outputString, L"Coordinates of point D",
MB_OK);
     outputString.Format(L"xB1 = %f\nyB1 = %f", xB1, yB1 );
     MessageBox (outputString, L"Coordinates of point B",
MB_OK);
     outputString.Format(L"xC1 = %f\nyC1 = %f", xC1, yC1 );
     MessageBox (outputString, L"Coordinates of point C",
MB_OK);
}

if (90<angleCompass && angleCompass<180){
     outputString.Format(L"xD1 = %f\nyD1 = %f", xD1, yD1 );
     MessageBox (outputString, L"Coordinates of point D",
MB_OK);
     outputString.Format(L"xB1 = %f\nyB1 = %f", xB1, yB1 );
     MessageBox (outputString, L"Coordinates of point B",
MB_OK);
     outputString.Format(L"xC1 = %f\nyC1 = %f", xC1, yC1 );
     MessageBox (outputString, L"Coordinates of point C",
MB_OK);
}

if (180<=angleCompass && angleCompass<270){
     outputString.Format(L"xD2 = %f\nyD2 = %f", xD2, yD2 );
     MessageBox (outputString, L"Coordinates of point D",
MB_OK);
     outputString.Format(L"xB2 = %f\nyB2 = %f", xB2, yB2 );
     MessageBox (outputString, L"Coordinates of point B",
MB_OK);
     outputString.Format(L"xC2 = %f\nyC2 = %f", xC2, yC2 );
     MessageBox (outputString, L"Coordinates of point C",
MB_OK);
}

if (270<angleCompass && angleCompass<360){
     outputString.Format(L"xD2 = %f\nyD2 = %f", xD2, yD2 );
     MessageBox (outputString, L"Coordinates of point D",
MB_OK);
     outputString.Format(L"xB2 = %f\nyB2 = %f", xB2, yB2 );
     MessageBox (outputString, L"Coordinates of point B",
MB_OK);
     outputString.Format(L"xC2 = %f\nyC2 = %f", xC2, yC2 );
     MessageBox (outputString, L"Coordinates of point C",
MB_OK);
}
```

12.8 Appendix VIII – Point in Polygon Algorithm Implementation

Source code for calculating if the origins of a point are inside the boundaries of a given polygon

```
/*************************************************************/
void PIP(double xUser, double yUser, double xp1, double yp1,
double xp2, double yp2, double xPOI, double yPOI)
/*************************************************************/
{
/* The points creating the polygon/TRIANGLE. */
double x[3];
double y[3];
double x1,x2;

/* Coordinates of the points */
x[0] = xUser;    y[0] = yUser;
x[1] = xp1;      y[1] = yp1;
x[2] = xp2;      y[2] = yp2;

/* The coordinates of the point */
double px = xPOI;
double py = yPOI;

/* How many times the ray crosses a line-segment */
int crossings = 0;

/* Iterate through each line */
for ( int i = 0; i < 3; i++ ){

/* This is done to ensure that we get the same result when the
line goes from left to right and right to left */
      if ( x[i] < x[ (i+1)%3 ] ){
           x1 = x[i];
           x2 = x[(i+1)%3];
      } else {
           x1 = x[(i+1)%3];
           x2 = x[i];
      }

/* First check if the ray is possible to cross the line */
      if ( px > x1 && px <= x2 && ( py < y[i] || py <=
y[(i+1)%3]) ){        static const double eps = 0.000001;

           /* Calculate the equation of the line */
           double dx = x[(i+1)%3] - x[i];
           double dy = y[(i+1)%3] - y[i];
           double k;

           if ( fabs(dx) < eps ){
                k = HUGE_VAL;         //INFINITY from math.h
           } else {
                k = dy/dx;
           }

           double m = y[i] - k * x[i];

           /* Find if the ray crosses the line */
           double y2 = k * px + m;
           if ( py <= y2 ){
                crossings++;
```

```
           }
      }
}

printf("The point is crossing %d lines", crossings);
if ( crossings % 2 == 1 ){
      printf(" thus it is INSIDE the polygon");
} else {
      printf(" thus it is OUTSIDE the polygon");
}

return;
}
```

12.9 Appendix IX – Complete Source Code of Aura

The full *Source Code* that has been developed for this project is attached to this Appendix. The native code includes all classes for the implementation of *Aura* on a Windows Mobile device, any additional DirectShow Mobile filters developed for the Augmented Reality interface and the first attempts for introducing an advanced 3D rendering pipeline for representing virtual elements in AR. The provided version is *Aura* v0.6.53.

The *Source Code* can be found in digital format, accessible by Microsoft Visual Studio 2005 or later, on the CD that accompanies the Thesis.

12.10 Appendix X – Final Evaluation Digital Content

In this Appendix, the reader can find the digital data that was developed for the purpose of the *Extensive Evaluation*. The digital content consists of the files that describe the routes and the waypoints, which the participants came across. Furthermore, the digital 3D model, which represents the environment that the *Evaluation* took place, including the applied textures, can be found in this section.

The *Evaluation Content* can be found in digital format on the CD that accompanies the Thesis. The waypoint and route information are provided as GPX (.gpx) and as KML (.kml) documents for easier browsing. The 3D scene is a VRML (.wrl) document compatible with most VRML viewers and the textures are plain computer images (.jpg).

The folders contain two sets of files – those that describe the route clockwise (R1) and vice versa (R2).

12.11 Appendix XI – Final Evaluation Information Sheet

This Appendix illustrates the *Participant Information Sheet* that was distributed to the people who engaged in the *Extensive Evaluation* of *Aura*.

The *Participant Information Sheet* can be found in digital format on the CD that accompanies the Thesis. It was created with Adobe Acrobat v7.0 (.pdf), but is accessible by v6.0 onwards.

12.12 Appendix XII – Final Evaluation Consent Form

This Appendix illustrates the C*onsent Form* that was distributed to the people who participated in the *Extensive Evaluation* of *Aura*.

The C*onsent Form* can be found in digital format on the CD that accompanies the Thesis. It was created with Adobe Acrobat v7.0 (.pdf), but is accessible by v6.0 onwards.

12.13 Appendix XIII – Final Evaluation Questionnaire

This Appendix presents the *Questionnaire* that was provided to the participants of the *Extensive Evaluation* of *Aura*.

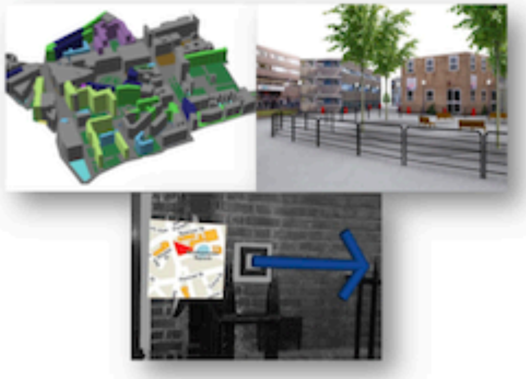
The following section presents a reduced version of the *Questionnaire*. Due to length restrictions, Parts 4 and 5 of the original document have been omitted because they are identical to Parts 1 and 2 respectively. The full *Questionnaire* can be found in digital format on the CD that accompanies the Thesis. It was created with Adobe Acrobat v7.0 (.pdf), but is accessible by v6.0 onwards.

CITY UNIVERSITY LONDON

School of Informatics

Department of Information Science

Aura End-User Testing



Document information	
Author(s)	Stelios Papakonstantinou
Version	6.0
Release Date	30 September 2010
Contact Details	stelios@soi.city.ac.uk
Revision	Final

Aura Testing Evaluation Sheet

Participant Name:		
Participant ID:		
Age:		
Sex:	Male	Female
Occupation:		
Email:		
Date:		
Weather conditions: (e.g. sunny / overcast / raining)		
Visibility conditions: (e.g. clear / fog)		

Part 1: Previous Experience

1. Have you had any previous experience with Virtual Reality or 3D applications?

1 (None)	2	3	4	5 (Expert)

2. Please name some, if any.

3. Have you had any previous experience with Augmented Reality or video see-through applications?

1 (None)	2	3	4	5 (Expert)

4. Please name some, if any.

5. Have you had any previous experience with real time, Context-Aware systems (SatNav, User Profiling, etc.)?

1 (None)	2	3	4	5 (Expert)

6. Please name some, if any.

7. To what extent do you use a Smartphone in your daily activities?

1 (Not at all)	2	3	4	5 (Very much)

8. What is the level of your Computer skills?

1 (None)	2	3	4	5 (Expert)

9. How often do you play Computer Games?

1 (Not at all)	2	3	4	5 (Very much)

10. What is the level of your Sense of Direction skills?

1 (Very low)	2	3	4	5 (Very high)

11. How often do you use way-finding applications?

1 (Not at all)	2	3	4	5 (Very much)

Part 2: Wayfinding Task 1

Before Task

Route Followed:	Clockwise (A)	Counter clockwise (B)
Selected Interface:	Virtual Reality (A)	Augmented Reality (B)

Task Performance

Start Time (GPS – HH:MM:SS)	
End Time (GPS – HH:MM:SS)	

Minimum Distance for Task Completion	
Distance Covered by User (meters)	

Observations

Part 3: Wayfinding Task 1 - Questionnaire

Usefulness

1. This interface provided effective support for my wayfinding task.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

2. The functionalities provided by the interface are sufficient for the task that I had to accomplish.

TRUE	FALSE

3. The interface helped me notice features of the environment, which I would not normally do.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

4. With this interface, I can search for a person or building effectively.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

5. With this interface, I can find my way to an unknown place effectively.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

6. The interface helped me make fewer mistakes while achieving my goal.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

7. I understood the layout of the locations presented to me.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

8. I understood when I reached the end of the task.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

9. The content in the Virtual Environment was helpful in informing me about my current task.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

10. The user interface was helpful in informing me of my current task.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

11. The interface increased my interest and motivation to complete the task.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

12. During the task, where were you looking at or where was your attention drawn to, most of the time?

1 (Physical environment)	2	3	4	5 (Virtual environment)

13. Can you think of any other mobile applications that this interface could have?

Ease of Use

14. The design of this interface is user friendly, compared to other mobile applications.

TRUE	FALSE

15. The interface is easy to use, compared to other mobile applications.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

16. I felt comfortable while using the interface, compared to other mobile applications.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

17. I was able to interact quickly with the interface, compared to other mobile applications.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

18. Interaction in the Virtual Environment was easy to me.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

19. Interaction in the Physical Environment was easy to me.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

20. I did not experience nausea, dizziness or discomfort while using this interface.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

21. The interface did not restrict my physical behaviour.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

22 I did not need more information about the environment in order to understand my position.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

Ease of Learning

23. I did not need a lot of time to learn how to use this interface, compared to the duration of the task that I had to accomplish.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

24. The interface is not complicated to use, compared to the provided functionalities.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

Satisfaction

25. I felt confident that the system was giving me correct instructions.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

26. I felt in control while using this interface.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

27. I would use the same or a similar technology again.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

28. I enjoyed navigating using this interface.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

29. The interface is exciting/entertaining to use.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

30. I would like to use this interface in my profession.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

31. I would like to use this interface in other contexts than my profession.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

32. Rate the realism level of this interface.

1 (Very Low)	2	3	4	5 (Very High)

33. Rate the immersion level of this interface.

1 (Very Low)	2	3	4	5 (Very High)

34. How much would you pay for an application that implements this interface?

System Performance

35. My physical position was accurately represented in the Virtual Environment.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

36. My physical orientation was accurately represented in the Virtual Environment.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

37. The update speed of this interface did not trouble me.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

38. The response of the interface to my manual input (touch screen) was prompt.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

39. The response of the interface to my physical changes (location, orientation) was prompt.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

40. The interface can support well multiple user interaction.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

41. The Smartphone display size and resolution supported this interface well.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

42. The text style (colour and size) was easy to read.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

43. The graphics (colour and size) were easy to comprehend.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

Part 6: After Both Tasks - Questionnaire

1. I was more motivated than usual for accomplishing the task by using a Smartphone.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

2. Which egocentric perspective did you prefer?

Virtual Reality	Augmented Reality

3. Which visualisation perspective did you overall prefer?

Egocentric	Oblique	Allocentric

4. Which interface do you prefer for accomplishing your wayfinding tasks?

Virtual Reality	Augmented Reality

5. How much would you pay for an application that implements both interfaces?

6. Please give us some comments on how we can improve the system and what do you believe that the current novelty and drawbacks of the system are.

12.14 Appendix XIV – Final Evaluation Data

The raw *Data* that has been collected from the participant replies, as well as from the use of *Aura* during the *Extensive Evaluation* is presented in this Appendix.

The *Data* file includes the objective and subjective responses retrieved from 23 individuals that participated in this *Evaluation* task. Any features that can identify the individuals have been intentionally removed due to privacy protection issues.

The *Data* can be found in digital format, accessible by IBM SPSS v17.0.0 (.sav) or later, on the CD that accompanies the Thesis.

12.15 Appendix XV – Final Evaluation Results

The *Statistical Analysis Results* generated by the data collected from the participant replies, as well as from the use of *Aura* during the *Extensive Evaluation* is presented in this Appendix.

Initially, this section presents a *Statistical Exploration* of all subjective and objective data variables. The *Shapiro-Wilk* test verified for which variables the distribution of our sample was normal. The variables that validated the assumptions of *Parametric Tests* occurred further *T-Tests* to compare the differences between the two interfaces. The variables that did not validate the *parametric* assumptions were compared against each other by using the *Wilcoxon* test and the *McNemar* test. The last set of replies that related to the overall use of the system did not sustain any further post-hoc analysis. The *Statistical Analysis Results* that were included in the main body of the Thesis concerned the data variables tha produced significant results. The full set is provided in this Appendix.

The *Statistical Analysis Results* can be found in digital format, accessible either by IBM SPSS v17.0.0 (.sav) or Adobe Acrobat v6.0 (.pdf), on the CD that accompanies the Thesis. The folder contains two sets of files that *Statistically Explore* the data variables – those that examine the *Frequencies* of string variables and those that *Explore* numerical variables. Furthermore, the T-Tests have been stored to different files than the Wilcoxon tests.

12.16 Appendix XVI – Potential Commercialisation Model

There are several scenarios for possible commercialisation and each one has its own characteristics attached. Each scenario is dedicated to a certain group of users that may need to adopt the framework functionalities for certain reasons. The second part of this chapter presents a potential route for commercialisation, which, although not unique or definitive, it can support the initial activities required in order to positively influence the commercialisation prospect.

In order to identify the most fitting (vertical) market that is going to offer the best potential for achieving the business requirements which have been identified earlier (i.e. maximised income, low risk, limited resources and portability options), we examine the potential applications for context-sensitive Augmented Reality, according to the following factors that can be linked to Porter's Five Forces model.

1. Monetisation potential;
2. Technical Feasibility;
3. Barriers to Entry;
4. Threat of Substitutes;
5. Buyer Power;
6. Supplier Power;
7. Degree of Rivalry.

Vertical	**Monetisation Potential**	**Technical Feasibility**	**Barriers to Entry**	**Threat of Substitutes**	**Buyer Power**	**Supplier Power**	**Degree of Rivalry**	**Summary**
1. Technology Licensing	Low to Medium	N/A	Affiliation with OEMs & mobile operators	Low	Very High OEMs in-house developments	N/A	N/A	1. Establishing IP Rights 2. OEMs in-house development
2. Urban Navigation	Medium	Medium Content Availability (POI Database), 3D Graphics	Medium Technology, Marketing, Distribution	Low 2D Maps, 3D Maps, Text Directions, PNDs	Medium	N/A	Low (e.g. Wikitude)	1. Large Market 2. Resource Availability 3. Context Accuracy
3. Spatial Search	Medium	High Content Availability (POI Database)	Medium Content, Marketing, Distribution	Very High Text & Voice Search, LBS	Medium	Very High Stakeholder participation	Very High Web & LBS businesses	1. No further Implementation required 2. High Competition 3. Resource Availability
4. POI Querying	Medium	Medium Content Availability (POI Database), Networking Resources	Medium Content, Marketing, Distribution	Very High Text & Voice Search, LBS	Medium	High Stakeholder participation	Medium (e.g. Layar, Wikitude, Sekia)	1. No further Implementation required 2. Resource Availability
5. Marketing	High	Medium Networking Resources, Ad Contents	High Affiliation with OEMs & mobile operators, Technology	Medium Text Ads, Banner Ads, SMS Ads	Medium	Medium	Low No competition in rich-media advertisement	1. Large Market 2. No Competition
6. Confined space Entertainment	Low to Medium	Medium Content Availability (POI Database), Networking Resources	Medium Affiliation with Stakeholders, Technology, Marketing	Low	Medium	Very High Stakeholder participation	Low	1. Customised Application
7. Open space Entertainment	Medium	Medium Networking Resources	Medium Technology, Marketing, Distribution	Low	Medium	High User participation	Low	1. User Social Networking
8. Virtual Surveillance and Exploration	Low to Medium	Medium Content Availability (POI Database), Networking Resources	Medium Technology, Marketing, Distribution	Low	Medium	High Stakeholder participation, User participation	Low	1. User Networking 2. Resource Availability

Table 12-1: Comparison of potential commercialisation applications for context-sensitive AR

Although every identified application of the developed framework may be commercially viable, the previous table shows that there are several factors, which may influence their immediate delivery. Thus, a dedicated plan is required, which will integrate most solutions and propose the best way to commercialise and maximise the benefits of the research and technology. The researcher and the enterprise unit of the university must adopt a strategic plan in order to become capable of producing new business agreements with various-sized businesses, without discarding any application. This way, there is a chance that the spin-off company, which will probably be formed, will be able produce the highest monetisation potential by combining all available applications and getting established as a strong body in the context-sensitive AR market. Although the aforementioned applications can be built independently, this is not the suggested solution. The main reason is that the technical requirements for each specific application may be either fundamentally different or they may overlap with those of another application. Thus, we are proposing a 3-phase progressive plan, which can utilise the results of this research and also takes into consideration the difficulties that may be faced by potential application developers. Following this plan, the technical solutions provided for each application, can be constructively fed to the applications developed during the subsequent phase. As a result, the technical effort is reduced and only new application-specific components need to be introduced in each phase.

12.16.1 Phase One (I)

In this phase, there are two vertical options that may be explored. These are the *Technology Licensing* and the *Mobile Marketing* solution. Technology licensing is the easiest way to make profit out of the framework, but it is definitely not the best solution because our spin-off company relinquishes all rights of the developed product and further collaboration cannot be guaranteed with the potential buyer. Furthermore, this option would render the rest impossible, because we would not possess exclusive ownership of the framework any more, in order to exploit it for our benefit. In contrast, the marketing solution can sustain a product, which has high commercialisation potential. The main reason is that the technical deliverables can be easily achieved because context accuracy is not a determining factor and because the contents of the ads are created by the advertising agency and not by our spin-off company. The most time-consuming factor of this business proposition is to establish relations with the other stakeholders, which include the advertising agency and the mobile operator. Another big advantage of this option is that there are not any competitors offering the business model that we are proposing. As a result, after careful negotiations with the participating stakeholders, this option can be considered as the most appropriate to invest in, during the first phase of our plan.

12.16.2 Phase Two (II)

During this phase of the commercialisation plan, two specific applications should be developed and promoted. These are *Spatial Search* and *POI Querying*. Both can have the form of COTS products and can be distributed through various channels. After negotiating agreements and licensing out the applications to the device manufacturers and/or mobile operators, these applications can be pre-installed on selected smartphones before being released to the end-users. The evident advantage of this distribution channel is that the number of smartphones that are being sold globally is continuously increasing, thus rendering this option very attractive. The disadvantages of this channel are that the consumers tend not to use most pre-installed applications, which decreases the supplier power. Additionally, the negotiations and the process of pre-installing external applications on new devices is quite long, mainly because supplementary versions of our software need to be developed, which must be compatible with the various of models that are being offered by the selected OEM or operator. The other distribution channel that can be adopted during this phase is the promotion of the application through a dedicated application store. Currently, there are three kinds of application stores that are operating. These include the device and platform manufacturers' (e.g. HTC, Microsoft), the mobile operators' (e.g. Vodafone, 3) and Independent Software Vendors (ISV) (e.g. Handango) e-shops. Compared to the pre-installation distribution method, application stores need a much shorter time frame in order for the application to reach the market, which provides various benefits for the software developers.

12.16.3 Phase Three (III)

The third phase is probably the most time consuming because the applications, which need to be developed and distributed, supply the most advanced features and need specific customisations in order to render them attractive for the general public. Namely, the applications that may press forward during this phase are: *Urban Navigation*, *Confined and Open-Space Entertainment* and *Virtual Surveillance and Exploration*. Urban navigation is a complicated solution that needs to integrate several characteristics, which have been identified during the research and by the LOCUS project as well. The effectiveness and usability of this application needs to reach high quality standards because there are several competitors with unquestionable expertise, who have offered solutions that have already been on the market for several years. Therefore, a COTS product needs to present distinct advantages compared to the existing solutions and this may be accomplished by investigating potential agreements with other stakeholders, such as car manufacturers who have started looking into this kind of technology. The *Confined Entertainment* solution is a specialised application, which can be developed only after establishing a strong relationship and agreement with an external body or organisation. The reason is that we need to develop unique features, which will complement an event or place. After the first implementation of this application, new versions, which may serve other stakeholders, will be much simpler and faster to develop. Finally, the *Open-Space Entertainment* and *Virtual Surveillance and Exploration* solutions target either independent users or groups, which require such services. Once more, the quality of the developed product needs to be more than satisfactory and this can happen only by implementing all advanced features that have been discussed in this report. Both applications need to work flawlessly in any part of the world and, similarly to the *Urban Navigation* solution, core GIS data is a vital component, which must be quantified according to various perspectives.