Diese Dissertation haben begutachtet:

....

DISSERTATION

Mobilizing the Geospatial Web: A Framework and Conceptual Model for Spatially Aware Mobile Web Applications

zur Erlangung des akademischen Grades
Doktor der Technischen Wissenschaften
(Dr. techn.)

ausgeführt am
Institut für Rechnergestützte Automation
Forschungsgruppe Industrielle Software

der Technischen Universität Wien

unter der Anleitung von
Univ.Prof. DI Dr.techn. Thomas Grechenig

durch
Dipl. Ing. Rainer Simon
Rosasgasse 4/3/1
1120 Wien
E086 881, 9426036

Wien, im März 2008 ....

Kurzfassung

Online-Kartendienste wie *Google Maps*, *Yahoo Maps* oder *Microsoft Live Maps* und virtuelle Globen wie *Google Earth* oder *NASA World Wind* führten in den letzten Jahren zu einem sprunghaften Anstieg des öffentlichen Interesses an geographischen Themen und Anwendungen. Die frühe Verfügbarkeit kostenloser Authoring- und Programmierwerkzeuge zu diesen Diensten bewirkte innerhalb kurzer Zeit die Formierung einer großen Gemeinde begeisterter „Web Mapping“ Enthusiasten: von gewöhnlichen Benutzern, die geographische Inhalte unterschiedlichster Art – seien es geokodierte Fotos oder GPS-Aufnahmen von Fahrradstrecken – über das World Wide Web bereitstellten, bis hin zu versierten Hobby-Entwicklern, die mit „Map Mashups“ – teilweise aufwändigen, selbst entwickelten Geo-Informations-Anwendungen – bestehende Geo-Daten neuen Nutzungsmöglichkeiten und Benutzerschichten zuführten.

Mit steigender Verbreitung mobiler Web-Nutzung kann die Verknüpfung freier Geo-Daten und mobiler Interaktion zu ***mobilen Geo-Informations-Anwendungen*** als nächster logischer Schritt betrachtet werden. Doch während komfortable Werkzeuge die Entwicklung und Distribution von Geo-Informations-Anwendungen für Desktop-Browser inzwischen enorm vereinfacht haben, bleibt die Entwicklung mobiler Anwendungen – und insbesondere mobiler Geo-Informations-Anwendungen und ortsbezogener Dienste („location based services“) – die Domäne einer Hand voll qualifizierter Experten.

Diese Dissertation beschäftigt sich mit der Integration von Geo-Information in mobile Web-Anwendungen. Das Ziel der Arbeit ist es, die technischen Voraussetzungen zu identifizieren, die nötig sind, um die Diskrepanz zwischen bestehenden Online-Kartographie Werkzeugen und der Komplexität mobiler Applikationsentwicklung zu schließen: Ein Ansatz wird aufgezeigt, der die Entwicklung ortsbezogener Dienste vereinfacht, und so das Experimentieren mit dieser Art mobiler Interaktion auch Endbenutzern ermöglicht. Die Arbeit baut dabei durchgehend auf existierende Standards und Methoden auf. Die etablierte Architektur klassischer Web-Anwendungen wird um Komponenten erweitert, mit denen bestehende geographische Inhalte ohne Zusatzaufwand als mobiler, ortsbezogener Dienst nutzbar gemacht werden.

Der Funktionsnachweis erfolgt über mehrere Tests, in denen der Einsatz der neuen Dienst-Architektur mit unterschiedlichen Endgeräte-Konfigurationen demonstriert wird. Dabei werden auch die praktischen Auswirkungen in der Realität auftretender Störfaktoren, wie z.B. GPS-Fehler, abgeschätzt und illustriert. Insbesondere wird dabei auch den aktuellen Entwicklungen am Endgerätemarkt Rechnung getragen: Neue räumliche Interaktionsmethoden und Benutzerschnittstellen, die durch Mobiltelefone mit integrierten Orientierungssensoren ermöglicht werden, werden speziell diskutiert, und im Rahmen eines konkreten Fallbeispiels veranschaulicht.

Abstract

Over the last years, Web mapping services like *Google Maps*, *Yahoo Maps* or *Microsoft Live Maps*, as well as three-dimensional virtual globes like *Google Earth* or *NASA World Wind* have sparked an unprecedented public interest in geospatial information. The early availability of free content authoring toolkits and open programming interfaces to these applications has spawned an avid community of Web mapping enthusiasts: from casual users who produce and share different types of geospatial content as varied as geo-referenced photographs or GPS recordings of cycling routes, to technology-savvy hobbyist developers who create elaborate “map mashups” – interactive map applications that combine existing geospatial data and bring them to new use for a variety of purposes from education, to planning of day-to-day activities, to entertainment.

With the increased use of the Web from mobile devices, the combination of user generated geospatial content and mobile interaction towards user-generated *mobile geospatial applications* can be viewed as a next logical step. Yet, while sophisticated tools have made the development and distribution of geospatial Web applications for the large screen almost effortless, the development of mobile applications in general – and of mobile geospatial applications and location based services in particular – remains the domain of a few trained specialists.

This thesis is concerned with the integration of geospatial data in mobile Web applications. The goal of this work is to identify the technical requirements and architectural components necessary to bridge the gap between existing end user Web mapping tools, and the current complexity of mobile application development: an approach is proposed that simplifies the creation of location aware geospatial applications, thus allowing end users and hobbyist developers to experiment with this new form of mobile interaction. The thesis thereby builds on existing standards and methods throughout. The established Web application architecture is extended with software components that allow existing geospatial Web content to be re-used as a mobile, location aware application, without additional effort or the need for modifications.

The verification of the system is achieved through several experiments and tests. It is illustrated how the new architecture can be applied to a wide range of mobile devices with varying characteristics and capabilities, and how external influences such as GPS error affect the system’s operation. In particular, the verification takes into account recent trends on the mobile handset market: novel ways of spatial interaction and user interfaces made possible by *orientation aware* mobile phones which feature integrated GPS, compass and tilt sensors are discussed, and exemplified in a separate experiment.

Acknowledgements

I want to thank my advisor Prof. Thomas Grechenig for his supervision, and Prof. Dieter Merkl for co-supervising my thesis. Furthermore, I am grateful to many friends and colleagues who have accompanied me throughout my work. I thank: Martin Tomitsch and Thomas Költringer for their help and assistance. Peter Reichl for his encouragment and valuable input. Markus Kommenda and Horst Rode for their support, especially during the final phase of this thesis. All my colleagues at the *Telecommunications Research Center Vienna* for providing a stimulating and fun research environment, in particular Harald Kunczier and the members of the *Point to Discover* project team. Paul and Katja for the occasional neogeo/mobile-gadget discussions over a coffee (or beer), which have found their way into this thesis in many intricate ways. The anonymous reviewers of the research papers related to this thesis for their valuable feedback that has helped to shape and refine my work.

Moreover, I want to thank the Austrian Competence Centre Programme **K***plus* and the project consortium and cooperating partners of the *Point to Discover* project for funding this work. In particular I would like to thank *mobilkom austria* for providing the block model data used to trial my prototype system, and *Falter* for providing me with sample POI data.

Additionally, I want to express my deep appreciation for the vast community of spirited *neogeographers*, *map hackers*, and *mashup creators* out there: your creativity and ambitious efforts to contribute, collect, re-assemble, and communicate geospatial knowledge to the world in new ways – be it for profane purposes, for fun, or to raise awareness for global issues and interrelationships that may affect all of us – has inspired this thesis and continues to amaze me.

Most importantly, I am grateful to my family: my parents for supporting me, and giving me the chance to be where I am today, and my wife Helga and my son Florian. Without your endless patience and moral support this thesis would never have been possible.

Original Publications

Parts of the work described in this thesis have been published. This thesis only exploits parts of collaborative publications that are directly attributable to the author.

A rudimentary architecture concept for a generic application platform that enables pointing-based queries in geo-referenced content using a mobile phone in conjunction with GPS and orientation sensors were first published at the *3rd Symposium on Location Based Services and TeleCartography* in Vienna, Austria, November 2005. The publication was later included in a *Springer Lecture Notes in Geoinformation and Cartography* issue of the same name, which was published in 2007 based on the symposium contributions (Simon *et al.* 2007a).

The conceptual model behind the visibility-based query process proposed by the author, along with a first (at this stage, incomplete) version of the *Local Visibility Model* was published at the *6th International Symposium on Web and Wireless Geographical Information Systems* (W2GIS 2006) in Hong Kong, China, December 2006 (Simon *et al.* 2006), where it was awarded Best Paper and selected for inclusion in the *Transactions in GIS* Journal (Simon *et al.* 2007b).

A revised version of the *Local Visibility Model* was furthermore presented to the World Wide Web standardization community at the *16th International World Wide Web Conference* (WWW 2007) in Banff, Canada, May 2007 (Simon and Fröhlich 2007).

The key results of the functional evaluation of the prototype Geo-Wand, which forms part of Section 8.3, have been presented at the *4th Symposium on Location Based Services and TeleCartography* in Hong Kong, China, November 2007 (no printed proceedings); and has been submitted for inclusion in a special issue of the *Journal for Location Based Services* to be published in 2008, based on selected symposium contributions.

Content

1 Introduction

Geographical information is a valuable resource that forms the basis of decision making processes in application fields as varied as urban planning, environmental monitoring and analysis, and emergency response, to name but a few. Digital technologies for the visualization, analysis and storage of geographical data have reached a long tradition and a high maturity in these professional domains. Only recently, the same topics have been discovered by an entirely new audience: Web mapping services like *Google Maps*[1], *Microsoft Live Maps*[2] or *Yahoo! Maps*[3] and 3D virtual globes like *NASA World Wind*[4] or *Google Earth*[5] opened up the realm of geographical information and geo-visualization to the public, by making global map data and high-resolution satellite imagery available to everyone, for free. The intuitive, direct-manipulation user interfaces and the explorative, almost playful user experience these applications enabled, captured the imagination of non-professional, everyday Web users and lead to a change in the appreciation for the value and the utility of geographical knowledge for a variety of purposes – from education, to planning of day-to-day activities, to entertainment.

The most profound impact these new Web mapping technologies had, however, was not the fact that they simplified the *access* to geographical data. More importantly, they put the tools for *creating* maps into the hands of average users: Open application programming interfaces, well-documented content markup languages and free editing tools allowed anyone to create, distribute and share online maps without the need for professional training or extensive programming skills. By mapping all kinds of data – from road construction sites to travel destinations, from locations of bars and restaurants to bike or hiking routes; by geo-coding their photos on photo sharing sites; by creating their own “map mashups” (small, customized geo-information applications based on free Web mapping toolkits) – users have begun to add a spatial dimension to the World Wide Web, in a magnitude and breadth that has not existed before.

With the increased use of the Web from mobile devices, the convergence of Web maps and mobile Web applications towards *mobile geospatial Web applications* can be seen as a next logical step: Already, mobile-specific versions of established local Web search

---

[1] http://maps.google.com

[2] http://maps.live.com

[3] http://maps.yahoo.com

[4] http://worldwind.arc.nasa.gov

[5] http://earth.google.com

or driving direction services are becoming more commonly used; as are mobile mapping applications. Only recently, socially-oriented mobile applications are appearing on the market and are increasingly adopted by a young audience.

Unlike on the “large screen” Web, however, the development methods for the mobile platform lag far behind the sophistication of their desktop counterparts: The fragmentation of the mobile handset market – with devices varying widely in terms of screen size and processing power, as well as in the programming and markup languages they support – makes the creation of cross-device-compatible mobile applications a tedious and costly task. State of the art tools are complex; standardization is still in a state of evolution. Positioning technologies that enable *location aware* applications are inconsistently supported and implemented. Their integration often requires substantial software development expertise, and in many cases poses a high cost of entry to developers and users alike (LaMarca *et al.* 2005).

1.1 Motivation and Goal

The work presented in this thesis is motivated by the imbalance between the end-user-driven development methods of mapping applications on the World Wide Web, and the complexity of mobile application development. The thesis argues that novel engineering approaches are needed in order to leverage *mobile geospatial Web applications*: In the same way that open authoring languages and free mapping toolkits have fostered the formation of an avid community of mapping enthusiasts on the “large screen” Web, suitable tools and infrastructure support can mitigate the challenges that exist on the mobile platform, and open up the domain of *mobile geospatial* and *location aware* applications to the same broad, non-professional, enthusiast community.

Consequently, the goal of this thesis is to identify the engineering challenges that restrict the end-user-driven development of novel mobile geospatial and location aware applications. In order to address the arising issues, the thesis proposes an application framework that establishes interoperability between Web mapping toolkits and mobile application technologies by introducing a set of client- and server-side extensions to the existing Web application infrastructure. A prototype of the framework is implemented; and a series of tests and exemplary scenarios, where mobile devices with varying characteristics are used to validate the framework’s functionality, is presented.

1.2 Research Questions

This thesis contributes to the foundations of the *Geospatial Web* by addressing research issues related to the integration of geo-information into Web applications for mobile devices. State of the art Web mapping tools and relevant markup standards are investigated, and existing mobile application scenarios are discussed. The first research question of this thesis is therefore:

> *How can existing Web mapping tools and standards benefit the development process of mobile geospatial applications and location based services?*

Secondly, the thesis concentrates on engineering issues that are critical to the development of mobile geospatial applications: While some contemporary mobile devices feature large, high-resolution screens, powerful processors and built-in GPS receivers, most others lack high-end graphics and input/output (I/O) capabilities and client-side positioning equipment. The resulting compatibility problems, user interface restrictions, and the unpredictability of whether and how the location of the mobile device can be determined, introduces substantial complexity to the development process and creates the need for supporting infrastructure. Hence, the second research question of this thesis is:

> *How can the application programming interface be de-coupled from the varying technological characteristics of different mobile devices, as well as the implementation details of different positioning methods?*

Finally, special consideration in the design of the proposed application framework is given to recent developments on the mobile handset market. New hardware features that are currently finding their way into state of the art mobile phones – such as digital compasses and accelerometer-based tilt sensors – promise to change the way people navigate, explore, and interact with their physical environment: Location aware applications that exploit attitude information to realize *orientation aware interaction* have been subject of research for several years. Innovative user interface metaphors like the *GeoWand* – a portable system that allows users to identify geographic objects by pointing towards them (Egenhofer 1999) – represent a departure from traditional thinking of how mobile users can interact with geospatial data.

Indeed, empirical research on human-computer interaction (HCI) has confirmed that it is inappropriate to apply desktop user interface idioms to mobile user interfaces (Kristoffersen and Ljungberg 1999): In mobile situations, user actions are often driven by the external environment (Pascoe *et al.* 2000); therefore, interaction with the computing device is likely to take place as a secondary task, while users are primarily involved in interactions with the physical world (Holland *et al.* 2002). Consequently, it has been suggested that alternative interaction styles, which minimize the amount of necessary attention are preferable. For the case of geospatial applications – which inherently involve interacting with physical objects – the Geo-Wand's *pointing* metaphor, as well as other forms of orientation aware user interfaces, indeed promise to be suitable candidates for reducing the cognitive load, as early user evaluations indicate (Fröhlich *et al.* 2006).

Based on the observation that there is a clear trend towards integrating sensors into state of the art mobile phones that will soon enable orientation aware interaction, the third research question of this thesis is therefore:

> *What infrastructure and tool support can be given to foster innovative interaction concepts enabled by orientation aware mobile phones, without adding to the complexity and implementation overhead of the development process?*

1.3 Structure of the Thesis

The remainder of this thesis is organized into 9 sections:

Section 2 provides an introduction to the fundamentals of Web Mapping – the technologies and standards used to gather, author, distribute and present geospatial data on the World Wide Web. This section provides the technological background for the work presented in this thesis.

Section 3 presents an overview of the state of the art in the second field that is relevant to this thesis: location aware computing. History and related research are discussed; categories of location aware applications that currently exist on the market are presented and illustrated by representative examples. This section provides the background to this thesis with regard to the application scenarios and interaction design principles the proposed framework aims to support.

Section 4 covers general issues involved with the "mobilization" of the Geospatial Web, i.e. the re-use of geo-referenced data from the World Wide Web on mobile devices. The potential benefits and challenges are discussed, and an approach is proposed which addresses the identified challenges on three levels: on the *content authoring* level, the *software development* level, and the *interaction design* level.

- On the *content authoring* level, the approach aims for interoperability with established content encoding formats and standards, so that existing Web map content – as well as the expertise required to author and maintain it – can be re-used.
- On the *software development* level, the approach proposes infrastructure support to minimize the entry barrier and the development effort for developers of mobile geospatial applications through abstraction mechanisms that shield them from the details of cross-device portability and compatibility issues, and the complexity involved with the integration of location awareness technologies.
- On the *interaction design* level, the approach embraces experimentation with new orientation aware interaction metaphors: These can potentially circumvent the I/O restrictions experienced with contemporary mobile devices by enabling "real world browsing", i.e. the interaction with geo-referenced content through sensor-driven *directional* and *visibility based* spatial queries, as explained in detail in Section 3.

Section 5 discusses the structure and components of the established Web application architecture. It proposes a set of extensions that support the development of mobile geospatial Web applications, and explains how they relate to the three-level approach presented in Section 4.

Section 6 introduces the *Local Visibility Model*, an egocentric data model and XML encoding format for geospatial query results. The Local Visibility Model represents a key concept in the proposed system architecture, as it serves both as a user interface abstraction mechanism, and an enabler technology for sensor-driven, orientation aware interaction.

Section 7 describes the technical details of the system implementation. Particular focus is put on the functional modules of the *visibility query component*, which carries out the task of computing the Local Visibility Model from the content base and a digital block model of the environment; the *presentation and formatting component*, which can optionally transform the Local Visibility Model into different output formats, according to the client device's capabilities, and the *messaging interworking functions*, which serve as extensions to the presentation and formatting component that translate from the Web-based communication model to message-based communication formats like SMS, MMS or instant messaging.

Section 8 describes the functional verification of the system. The verification is based on several measures, which investigate how potential external influences affect the operation of the framework and the applications built on top of it under real world conditions. Three typical use cases are presented to exemplify different mobile user interfaces and interaction styles that are supported by the framework: The first example presents the case of location aware interaction using a GPS receiver for positioning. The second example depicts a legacy scenario, where the framework is used in conjunction with a low-end GSM phone, without GPS or the ability to access the Web. The third example illustrates a potential future scenario of orientation aware interaction, using a functional prototype implementation of a Geo-Wand.

Section 9 re-examines the scope and contributions of this thesis and discusses how they relate to – and where they differ from – existing work found in literature.

Section 10 concludes the thesis by summarizing the key results and contributions of the thesis. Their relation to the three research questions stated in section 1.2 is discussed, and potential areas of future work are highlighted.

2 Web Mapping

> *“For novices and geospatial experts alike, mapping technologies are undergoing as significant a change as has been seen since mapping first went digital. The prior introduction of Geographic Information Systems (GIS) and other digital mapping technologies transformed traditional map making and introduced an era of specialists in these new geographic technologies. Today, an even newer set of technological advancements are bringing an equally massive change as digital mapping goes mainstream. The availability of Global Positioning System (GPS), broadband Internet access, mass storage hard drives, portable devices, and – most importantly – web technologies are accelerating the ability to incorporate geographic information into our daily lives.”*
> (Mitchell 2005)

The history of digital mapping technologies spans more than three decades and dates back to the first Geographic Information Systems (GIS) in the 1970s (Wikle 1991). The growing relevance of the Internet as a global communication channel, and the World Wide Web as a publishing medium, has lead digital mapping technologies to evolve beyond their initial focus on a strictly professional audience: Industry standards and data exchange formats for the distribution of map imagery and geographical feature information across the Web have simplified the inclusion of maps into Web documents. New implementation techniques made possible by state of the art Web technologies such as JavaScript and XML have enabled more intuitive and fluid user experiences.

This section provides an introduction into the state of the art of Web mapping. It presents relevant standards and technologies, and discusses the evolution of digital maps from a professional cartography and analysis tool, towards a mass-audience publishing medium. Particular focus is put on the latest phase in this evolution, which happened mostly over the last two years: The notion of *participatory mapping* and the term *neogeography* – which has recently become popular to describe a paradigm shift from traditional map making towards mapping as a de-centralized community activity – are introduced, along with the *GeoStack* – an emerging reference model for the data management lifecycle of geospatial information on the World Wide Web. Due to the timeliness of the topic, scientific literature on the subject is scarce. The author therefore attempts to give a complete and accurate account of the latest developments in Web mapping by pointing to relevant essays, news articles and other online sources in addition to research articles.

2.1 Geographic Information Systems

A multitude of varying definitions for what GIS are can be found in literature. Most of them, however, agree on the principal properties of a GIS: GIS are systems comprised of computer hard- and software used to *capture*, *store*, *manipulate*, *display* and *analyze* spatially-referenced data. GIS experienced their major growth period in the early 1980s with the wide-spread availability of affordable personal computers (Wikle 1991), and reached a state of maturity and saturation in the late 1990s (Egenhofer and Kuhn 1998). Tailored towards a consistently professional audience, GIS have traditionally been expensive and hard to use: adequate knowledge of geography, cartography and database management systems was a prerequisite; and frequently, only a single individual in a workplace had the expertise to use the software (Traynor and Williams 1997). Applications for GIS are varied and include, among other fields, urban planning, municipal facility management, transportation monitoring, scientific and environmental investigations, energy resources and waste management, telecommunications network planning, as well as planning and analysis tasks in geology, hydrology, epidemiology, archeology or defense.

With the growing proliferation of the Internet in the 1990s, interoperability of GIS across networks became an apparent requirement, leading to the formation of the Open Geospatial Consortium (OGC)[6] in 1994: Founded as a consortium of industry, academic and public members, the purpose of the OGC was to promote the development and use of open, interoperable standards and systems in the geo-information domain. In 2000 and 2002, respectively, the OGC published two specifications which are particularly relevant in the context of this thesis: the *Web Map Service* specification (WMS) and the *Web Features Service* specification (WFS).

2.1.1 WMS

The OGC Web Map Service specification defines an interface for the dynamic generation of map images, and their delivery over the Web. The WMS specification provides three operations, which can be invoked from a browser via HTTP (de la Beaujardiere 2004): *GetCapabilities*, *GetMap* and the optional *GetFeatureInfo*.

---

[6] http://www.opengeospatial.org/

- **GetCapabilities:** This operation provides service metadata, i.e. it returns an XML document describing the map server's information content and acceptable request parameter values. Information that is available through *GetCapabilities* includes, for example, maximum width and height values clients are permitted to include in a *GetMap* request; or information about the contents, coordinate reference systems, and appropriate scale ranges of the individual data layers stored on the map server.
- **GetMap:** The *GetMap* operation produces a map image in a pictorial format – such as PNG, GIF or JPEG (or occasionally in a vector graphics format such as SVG) according to a set of request parameters contained in the request URL. Mandatory request parameters include the width, height and requested file format of the map image; the list of data layers the map should display; the rendering styles used for each layer; and the bounding box of the geographical area depicted on the map image. Additional optional request parameters include e.g. a background color value, or time and elevation for layers (in case the layers vary across these dimensions).
- **GetFeatureInfo:** This operation provides WMS clients with information about a specific feature shown on the map image. The client thereby queries the server with a point on the map (in pixel coordinates), and indicates which data layer should be investigated. Since WMS is stateless, the defining parameters of the original map (such as width, height or bounding box) must be included in the request as well.

2.1.2 WFS

The OGC Web Features Service (WFS) specification defines an interface for the exchange of geographic feature information via HTTP: Unlike WMS, which produces a graphical portrayal of the geographic feature information, WFS returns the feature information itself – i.e. the data behind the map – encoded in XML. The XML grammar used to express the feature information is the *Geography Markup Language* GML (Portele 2007), another OGC standard described in detail in the next sub-section. Like WMS, the WFS specification provides several operations (Vretanos 2005):

- operations that describe the interface, such as *GetCapabilities* (which is similar to the *GetCapabilities* operation defined in the WMS specification) and *DescribeFeatureType* (which describes the structure of any feature type the service can provide);

- operations for the retrieval of features, such as *GetFeature* or *GetGmlObject*;
- operations such as *Transaction* and *LockFeature*, which are needed to modify (i.e. create, update or delete) geographic features on the server.

A WFS-complaint server can implement a read-only WFS (“basic WFS”), i.e. only support the interface description and retrieval operations; or it can offer a “transaction WFS”, which implements the transaction operations in addition to the description and retrieval operations (Percivall 2003).

2.1.3 GML

As explained above, the *Geography Markup Language* GML is an XML grammar for the modeling, transport and storage of geographic feature information, i.e. the geometry and properties of geographic features (Percivall 2003). According to the original GML specification (Portele 2007), a geographical feature is thereby defined as an abstraction of real world phenomena, associated with a location relative to the Earth.

```
...

<gml:Polygon>
  <gml:exterior>
    <gml:LinearRing>
       <gml:posList dimension="2">
         16.414249 48.232792 16.412683 48.232643
         16.411106 48.233929 16.409925 48.23329900000002
         16.409369 48.233757 16.411331 48.234821
         16.410946 48.23513400000001 16.412008 48.235764
         16.412811 48.235107 16.412704 48.234829
         16.413092 48.234512 16.413553 48.234486
         16.41395 48.23407 16.414442 48.23341400000001
         16.414356 48.233101 16.414249 48.232792
      </gml:posList>
    </gml:LinearRing>
  </gml:exterior>
</gml:Polygon>

...
```

Figure 1. GML 3.0 code sample

In its current version (3.0) GML defines a number of XML schemas for representing different types of geospatial phenomena: from simple 2D linear features and geometric primitives (such as points, lines or polygons), to features with complex, non-linear 2D or 3D geometry; features with 2D topology; or dynamic features with temporal properties and collections thereof (Percivall 2003). A basic GML syntax example that describes a polygonal area is shown in Figure 1.

2.2 “Neogeography” and the Next Generation of Web Maps

Until recently, interactive maps on the World Wide Web were primarily based on the OGC standards and architecture: relying on a WMS-compliant GIS, maps were rendered on demand at the server, and delivered to the client as an image embedded in a Web page. To move the viewport, zoom in and out, or change the layers displayed on the map, it was necessary to request a newly rendered map image (and Web page) from the server (Elson *et al.* 2007). In 2005 Google’s *Google Maps* pioneered a new class of Web maps: Instead of a strict client/server architecture, *Google Maps* followed a distributed approach that shifted much of the application logic into the client browser. Sophisticated JavaScript programming and asynchronous, background data transfer enabled seamless panning and zooming without the need for page reloads. (This Web development technique is now commonly referred to as “Asynchronous JavaScript and XML”, or AJAX[7].) Pre-rendered image tiles with anti-aliased fonts and map features were employed to improve the visual appearance of the map, since data layering was no longer handled on the server, and map images no longer needed to be rendered in real time (Elson *et al.* 2007).

As the authors of (Elson *et al.* 2007) explain further, the shift of application functionality from the server to the client had an unexpected side effect: Enthusiast Web developers soon reverse-engineered the mechanism that was used to overlay data on the map, and started to create their own applications, using their own geo-referenced data or data from alternative sources: The term “mashup” has since become customary to describe this new type of Web application that combines content and functionality from various sources, and integrates them under a single, browser-based user interface. Map-based mashups quickly gained popularity: therefore, most major Web map service providers

---

[7] J. J. Garrett (2005) AJAX: A New Approach to Web Applications, http://www.adaptivepath.com/publications/essays/archives/000385.php

have since released public application programming interfaces (APIs) in order to support developers in the creation of new mashups, without the need for reverse-engineering, or the danger of infringing upon the map service providers' intellectual property.

Arguably, the emergence of map mashups is a significant step in the history of digital mapping, viewed by some as a paradigm shift from its origin as a discipline for experts and central, government-funded agencies, towards a de-centralized, collaborative movement[8]. While participatory approaches to mapping and GIS are not new, and the debate on their impact on society had been lead by the GIS research community years before the emergence of the first mashups, the mashup phenomenon undoubtedly lead to an unprecedented upsurge in the public interest in geospatial technologies outside traditional academic and industry circles (Rouse *et al.* 2007): Grass-roots initiatives, such as the efforts to map storm damage and flooding in New Orleans in the aftermath of Hurricane Katrina using Google Maps and Google Earth[9]; the *OpenStreetMap*[10] project that aims to compile a free, global roadmap from user-submitted GPS recordings; *Tracks4Africa*, which follows a similar approach to map rural and remote parts of Africa[11], or Google's own initiative to map the streets of the Indian city of Hyderabad with the help of volunteers equipped with low-cost GPS units[12]; as well as the countless smaller-scale projects and Websites where users share geo-coded photographs (such as Panoramio[13] or Flickr[14]); or collaboratively annotate maps with information ranging from locations of bars[15], restaurants[16] or WiFi hotspots[17] to favorite running[18] or biking routes[19] are just some of the examples that indicate how participatory mapping is changing the way geospatial data is being produced, distributed and used.

---

[8] E. Ratliff (2007) Google Maps Is Changing the Way We See the World, *Wired Magazine*, issue 15.07, http://www.wired.com/techbiz/it/magazine/15-07/ff_maps

[9] D. M. Ewalt (2005) Google is Everywhere, *Forbes.com,* http://www.forbes.com/technology/2005/09/02/hurricane-google-map-rescue-cx_de_0902google.html

[10] OpenStreetMap, http://www.openstreetmap.org

[11] Tracks4Africa, http://www.tracks4africa.com/

[12] B. Forrest (2007) Google Uses Crowdsourcing To Create Maps In India, *O'Reilly Radar*, http://radar.oreilly.com/archives/2007/08/google_uses_cro.html

[13] Panoramio, http://www.panoramio.com/

[14] Flickr, http://flickr.com/map/

[15] MappyHour, http://mappyhour.nerl.net/

[16] Yelp, http://www.yelp.com/

[17] hotspotr: WiFi Cafes and Hotspots, http://hotspotr.com/wifi

[18] MapMyRun.com, http://www.mapmyrun.com/

[19] Bikely.com, http://www.bikely.com/

The term *neogeography* has recently become a popular synonym for this new form of end-user driven online geography: Paraphrased as a blend of "blogs with online maps" in a National Geographic News article[20], the Turner (2006) defines *neogeography* as a collection of technologies and "*tools that fall outside the realm of traditional GIS*" which "*combines the complex techniques of cartography and GIS and places them within reach of developers.*"

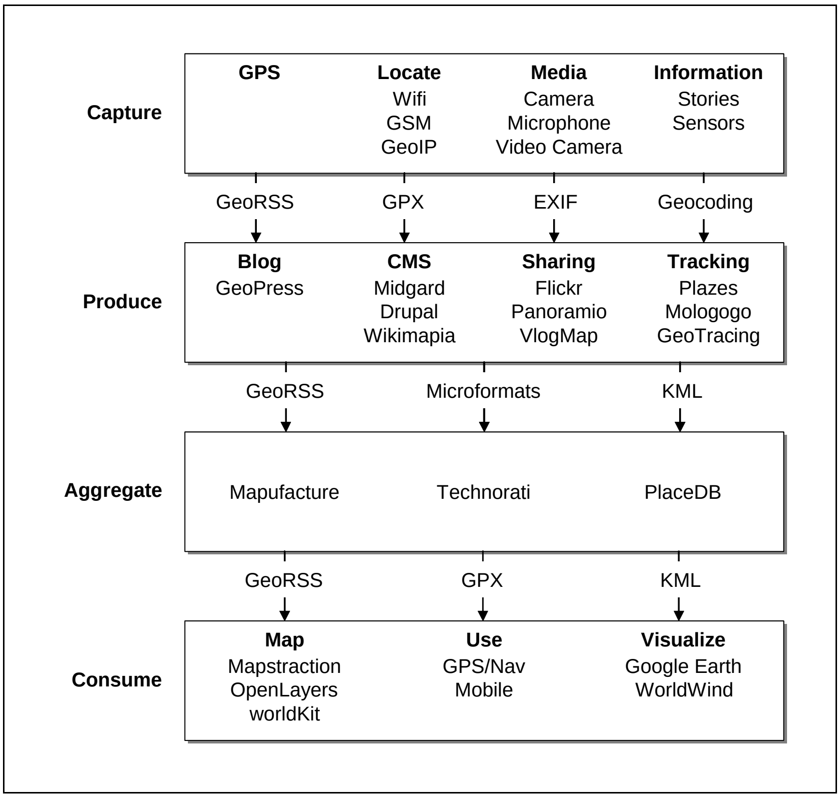


Figure 2. The *GeoStack* according to (Taylor 2006)

The definitions of what *neogeography* is are still vague; and the tools, standards, and interfaces that form its technological basis are fragmented. In order to harmonize the various technologies involved, the *neogeography* community is therefore working on the specification of the *GeoStack* – an agreed reference model that covers the entire life-cycle of geospatial data on the Web, from capturing to consumption (Turner 2006). The *GeoStack* is organized according to the various steps in the data management lifecycle,

---

[20] J. Jackson (2006) "Neogeography" Blends Blogs With Online Maps, *National Geographic News*, http://news.nationalgeographic.com/news/2006/04/0425_060425_map_blogs.html

and defines suitable exchange formats at the interfaces between them. Depending on the type of application, the data can take different paths through the stack. As shown in Figure 2, the *GeoStack* defines four steps:

- **Capture.** The capture step encompasses the technologies used to gather geospatial data and geo-coded media, such as GPS tracks or other arbitrary digital data that is referenced to a real-world location (e.g. images, audio, video, textual annotations or sensor data) using any form of positioning method (e.g. GPS or geo-location derived from known locations of nearby WiFi hotspots or GSM cell towers, or from the registered location of the assigned IP address).
- **Produce.** The produce step includes the tools used to author geo-referenced content and media, or to associate existing content and media with location information. Examples highlighted by Turner include *GeoPress*[21], a plugin to the popular blog engines *WordPress*[22] and *Moveable Type*[23] that allows users to add location information to their blog posts; *Midgard*[24] and *Drupal*[25], two open source content management systems, both of which offer support for import and export of geographic information; *Wikimapia*[26], a collaborative mapping project where users can freely annotate locations and areas on a world map; the popular photo-sharing sites *Flickr* and *Panoramio*, which both support geo-coding of photos; *VlogMap*[27], a service that shows the location of participating video bloggers on a map; the social networking services *Plazes*[28] and *Mologogo*[29], and the online geo-application toolkit *GeoTracing*[30], all of which allow users to broadcast their location to a Web map in real-time from their mobile phone.
- **Aggregate.** The aggregate step covers tools and services involved with the storing, filtering and re-distribution of original geospatial data. Examples given by Turner include *Mapufacture*[31], an aggregation service that creates Web maps

---

[21] http://georss.org/geopress

[22] http://wordpress.org/

[23] http://movabletype.org/

[24] http://www.midgard-project.org/

[25] http://drupal.org/

[26] http://wikimapia.org/

[27] http://community.vlogmap.org/

[28] http://beta.plazes.com/

[29] http://mologogo.com/

[30] http://www.geotracing.com/

[31] http://mapufacture.com/

from geospatial data in various formats; and the content feed aggregators *Technorati*[32] and *PlaceDB*[33].

- **Consume.** The consume step encompasses tools to map and visualize geospatial data in two- or three-dimensional form (examples presented by Turner include *Mapstraction*[34], an abstraction library that provides a common JavaScript interface to various Web mapping APIs; the open source JavaScript mapping library *OpenLayers*[35]; the open source Flash-based mapping application *worldKit*[36]; and the 3D virtual globes *Google Earth* and *NASA World Wind*), or to prepare and transfer it for use on mobile devices such as GPS personal navigation devices, car navigation systems or mobile phones.

As shown in Figure 2, different exchange formats are employed at the interfaces between the lifecycle stages, depending on the type of data that is transported. The following sub-sections describe the different formats in detail.

2.2.1 GEORSS

Geographically Encoded Objects for RSS, or GeoRSS, is an extension to the RSS online publishing format. RSS – short for *Really Simple Syndication* (since RSS version 2.0) – is a family of XML formats used on Web sites to notify readers of new content, news items or updates (Turner 2006). The motivation behind RSS is to provide a simple, brief, and structured description of new content that includes only key descriptive elements like author, date, title, narrative description, and a hypertext link, to help a reader decide what source materials are worth examining in more detail[37].

GeoRSS is a proposal for a standardized mechanism to describe the location of an RSS content source: it adds geospatial data types to RSS, such as points, lines, boxes or polygons. GeoRSS defines three types of *encodings*: *Simple*, *GML* and *W3C Geo*. The *Simple* encoding[38] was developed to be maximally concise and speed up the adoption and use of GeoRSS (Turner 2006), albeit at the expense of upwards compatibility. In the

---

[32] http://technorati.com/

[33] http://placedb.org/

[34] http://www.mapstraction.com/

[35] http://openlayers.org/

[36] http://worldkit.org/

[37] http://georss.org/overview

[38] http://georss.org/simple

Simple encoding, each of the four GeoRSS objects (point, line, box, polygon) requires only a single tag (compare Figure 3).

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:georss="http://www.georss.org/georss">
  <title>GeoRSS Example</title>
  <subtitle>A GeoRSS Simple Example</subtitle>
  <updated>2007-10-05T10:38:10Z</updated>
  <author>
    <name>Rainer Simon</name>
    <email>simon@ftw.at</email>
  </author>
  <entry>
    <title>Tech Gate</title>
    <link href="http://p2d.ftw.at/examples/georss"/>
    <summary>This is where the author’s Office is located.</summary>
    <updated>2007-10-05T10:38:10Z</updated>

    <georss:point>48.23266854182516 16.41315407848177</georss:point>

  </entry>
</feed>
```

Figure 3. GeoRSS Simple example

The *GML* encoding, on the other hand, is realized as a GML profile[39] and supports encoding of complex geographic geometry. An example for the GeoRSS *GML* syntax used to describe a point is shown in Figure 4.

Per default, GeoRSS uses the WGS 84 (*World Geodetic System* 1984) coordinate reference system. (This geocentric and globally applicable coordinate system describes locations based on their geographical latitude and longitude. It is most commonly known due to the fact that it is used by the GPS system.) While in the *Simple* encoding, WGS 84 is the only valid reference system, the *GML* encoding allows for other reference systems as well. The *W3C Geo* encoding, finally, exists due to historical use and is limited to describing points only (Turner 2006). Its rapid uptake has significantly motivated the GeoRSS standardization effort[40].

---

[39] http://georss.org/gml

[40] http://georss.org/w3c

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:georss="http://www.georss.org/georss">

  <title>GeoRSS Example</title>
  <subtitle>A GeoRSS Simple Example</subtitle>

  <updated>2007-10-05T10:38:10Z</updated>
  <author>
    <name>Rainer Simon</name>
    <email>simon@ftw.at</email>
  </author>

  <entry>
    <title>Tech Gate</title>
    <link href="http://p2d.ftw.at/examples/georss"/>
    <summary>The Tech Gate is the building where the author’s
    Office is located.</summary>
    <updated>2007-10-05T10:38:10Z</updated>

    <georss:where>
      <gml:Point>
        <gml:pos>48.23266854182516 16.41315407848177</gml:pos>
      </gml:Point>
    </georss:where>

  </entry>
</feed>
```

Figure 4. GeoRSS GML example

2.2.2 GPX

The *GPS Exchange Format* GPX is a widely adopted XML format for the exchange of waypoint, route and track data between GPS devices and applications (Turner 2006). Many handheld GPS navigation devices support the GPX format natively, and numerous applications are available for the manipulation, editing, conversion or visualization of GPX data[41].

While the data types defined in GPX are syntactically similar to some of the data types defined in GML or GeoRSS (e. g. points or lines), GPX can also store GPS-specific metadata, such as the device that created the file or timestamp information, along with

---

[41] http://www.topografix.com/gpx_resources.asp

the geographical data. Figure 5 shows a GPX example that encodes a single track, comprised of 14 track points, each described in WGS 84 coordinate space.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<gpx xsi:schemaLocation="http://www.topografix.com/GPX/1/1
                         http://www.topografix.com/GPX/1/1/gpx.xsd"
     xmlns="http://www.topografix.com/GPX/1/1"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     creator="Rainer Simon" version="1.1">

  <trk>
    <name>GPX Example</name>
    <trkseg>
      <trkpt lon="16.41464134525965" lat="48.2328176133405"/>
      <trkpt lon="16.413075343360738" lat="48.232668618624906"/>
      <trkpt lon="16.41149835230176" lat="48.23395462393187"/>
      <trkpt lon="16.41031734695729" lat="48.233324627922094"/>
      <trkpt lon="16.40976135014426" lat="48.23378262979306"/>
      <trkpt lon="16.411723359154742" lat="48.234846623164174"/>
      <trkpt lon="16.411338361330436" lat="48.23515962445985"/>
      <trkpt lon="16.412400366616545" lat="48.23578962087108"/>
      <trkpt lon="16.413203362046918" lat="48.23513261816892"/>
      <trkpt lon="16.413096359892908" lat="48.234854618532594"/>
      <trkpt lon="16.413484357688404" lat="48.23453761722662"/>
      <trkpt lon="16.413945357719424" lat="48.23451161567188"/>
      <trkpt lon="16.414170175085335" lat="48.234098738732804"/>
      <trkpt lon="16.41464134525965" lat="48.2328176133405"/>
    </trkseg>
  </trk>
</gpx>
```

Figure 5. GPX example defining a track

2.2.3 KML

The *Keyhole Markup Language* KML is an XML format originally created by the company *Keyhole, Inc.* Acquired by Google in 2004, Keyhole, Inc. developed the predecessor to *Google Earth*. Accordingly, KML is designed to encode geographic data for viewing in a virtual globe browser[42]. KML provides geographic data types similar to those provided by GeoRSS or GML – such as points, lines or polygons – as well as arbitrary textured and non-textured three-dimensional objects. The most noteworthy difference to GML, however, is the inclusion of graphical styling information along with

---

[42] http://code.google.com/apis/kml/documentation/whatiskml.html

the geometric information: Available styling attributes include color, line width and opacity; as well as the possibility to add icons and labels (which may include HTML-formatted descriptive text), and to set preferred camera view positions for each feature.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.1">

<Document>
  <name>KML Example</name>

  <Placemark>
    <name>Tech Gate</name>
    <Point>
      <coordinates>16.41315407848177,48.232668541825,0</coordinates>
    </Point>
  </Placemark>

  <Placemark>
    <name>Route Donaucity</name>
    <LineString>
      <tessellate>1</tessellate>
      <coordinates>
        16.414249,48.232792,0 16.412683,48.232643,0
        16.411106,48.233929,0 16.409925,48.23329900000002,0
        16.409369,48.233757,0 16.411331,48.234821,0
        16.410946,48.23513400000001,0 16.412008,48.235764,0
        16.412811,48.235107,0 16.412704,48.234829,0
        16.413092,48.234512,0 16.413553,48.234486,0
        16.41395,48.23407,0 16.414442,48.23341400000001,0
        16.414356,48.233101,0 16.414249,48.232792,0#
      </coordinates>
    </LineString>
  </Placemark>
</Document>

</kml>
```

Figure 6. KML example defining a point- (POI) and
a line-shaped (path) placemark

In addition to *Google Earth*, KML is also a supported file format in the *ESRI ArcGIS Explorer* virtual globe browser[43] and NASA’s *World Wind*. Furthermore, a subset of

---

[43] http://www.esri.com/arcgisexplorer/

KML[44] is supported by *Google Maps* (including the *Google Maps for Mobile* mobile phone map client[45]). An example for the syntax of KML is shown in Figure 6. The example contains the same track as shown in the GPX example (Figure 5), and the point shown in the GeoRSS examples (Figure 3 and Figure 4).

2.2.4 GEOTAGGING FORMATS

Various other formats and conventions have evolved for associating location information with digital content and media of different types. Depending on the kind of data, and the format used to store and exchange it, different notations have become customary: for Web pages, location information in the form of a WGS 84 coordinate pair and/or a descriptive place name is frequently stored in the head of the HTML document as a `<meta>` tag; digital photographs can embed location in the EXIF header of the file (Turner 2006); various content upload and sharing services on the Web (such as Flickr or the video sharing service YouTube[46]) use their own conventions and tools for 'geotagging', i.e. adding location metadata to content during or after upload.

Microformats are another noteworthy recent effort to enrich arbitrary content included in Web documents with location information: Microformats define naming conventions for existing XHTML facilities such as CSS class attributes, in order to add semantic meaning to the words and elements contained in the document (Khare 2006): *adr* and *geo*, for example, are two Microformats that indicate a street address or a geographic coordinate, respectively (Turner 2006). Figure 7 shows a corresponding XHTML code sample that illustrates how these Microformats can be used to denote parts of the document that carry geographic information, without breaking the XHTML syntax.

2.2.5 GOOGLE MAPPLETS

Mapplets are a technology introduced by Google in 2007[47]. Mapplets are conventional map mashups built with the Google Maps JavaScript API, which conform to a specific XML syntax. This syntax allows them to be dynamically linked to the generic *Google Maps* viewer[48]: The premise of mapplets is to allow different mashups to be combined

---

[44] http://code.google.com/apis/kml/documentation/mapsSupport.html

[45] http://www.google.com/gmm/

[46] http://www.youtube.com/

[47] http://google-latlong.blogspot.com/2007/05/mashup-of-mashups.html

[48] http://www.google.com/apis/maps/documentation/mapplets/

on a single map, with each Mapplet acting as one layer of data that can be switched on and off like a data layer in a GIS or a virtual globe browser.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
                      "DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Microformats Example</title>
  </head>

  <body>
    <h1>Microformats Example</h1>
    <p>The office of the author is located at the Tech Gate:</p>

    <div class="adr">
      <div class="street-address">Donau-City-Str. 1</div>
      <span class="postal-code">1220</span>
      <span class="locality">Vienna</span>
      <div class="country-name">Austria</div>
    </div>

    <div class="geo">
      <span class="latitude">48.232668541</span>,
      <span class="longitude">16.41315407</span>
    </div>
  </body>
</html>
```

Figure 7. adr and geo Microformats example

In that sense, mapplets bridge the gap between data formats such as GPX or KML – which are declarative, and inherently suited for layered viewing – and map mashups, which are procedural applications, written in a scripting language, but are frequently used interchangeably to achieve the same result: to visualize a geographical data set comprised of spatial features like points, lines or polygons. Figure 8 depicts a code example for a simple Mapplet that draws a single marker for a point of interest.

Mapplets can be considered of interest in the context of this thesis insofar, as JavaScript and mashup development has become a well-established skill among the developer community. Hence, mashups are prevalent on the Web, and constitute a vast source of free geospatial data. Moreover, a wide range of tools has meanwhile become available that support the creation of mashups using drag-and-drop techniques, thus making mashup development even more accessible to casual Web users with no programming skills: Examples include tools like *CommunityWalk*[49], *Wayfaring*[50], *Frappr*[51] or *Platial*[52], as well as Google's own programming-free mapping tool *My Maps*, which is included with *Google Maps*.

```
<?xml version="1.0" encoding="UTF-8"?>
<Module>
  <ModulePrefs title="Mapplet Example"
              description="A simple Mapplet example"
              author="Rainer Simon"
              author_email="simon@ftw.at "
              height="400">
    <Require feature="sharedmap"/>
  </ModulePrefs>

  <Content type="html">
    <![CDATA[
      <h2>Example</h2>
      <script>
        // Set map center
        var map = new GMap2();
        var techGate = new GLatLng(48.232668541, 16.41315407);
        map.setCenter(techGate, 2);

        // Add a marker
        var marker = new GMarker(techGate);
        map.addOverlay(marker);
      </script>
    ]]>
  </Content>
</Module>
```

Figure 8. Google Mapplets example

---

[49] http://www.communitywalk.com/

[50] http://www.wayfaring.com/

[51] http://www.frappr.com/

[52] http://www.platial.com/

3 Location Aware Computing

The previous section introduced the technologies, standards and interfaces that are relevant to the domain of Web Mapping. In essence, these can be regarded as the “host environment” that defines the technological boundary conditions for the work presented in this thesis. This section, on the other hand, discusses related work from a domain that is relevant to this thesis, not so much from a technological point of view, but rather with regard to the application scenarios and interaction design principles that influence and motivate this work: the domain of *location aware computing*.

Within the scope of this thesis, the term *location aware* thereby refers to the general ability of an application to infer information about the physical location of its user – either through sensors on the device, through external infrastructure, or a combination thereof – and to adapt its information offering, user interface or behavior in response to this information. The research issues associated with location aware computing span a wide range of topical areas – from societal issues, to questions of interaction design and usability, to engineering topics: Research has, for example, been conducted on the inherent privacy and anonymity issues that arise out of the technological ability to monitor and track peoples’ locations in real-time with low-cost equipment (Dobson and Fisher 2003), (Consolvo *et al.* 2005) and possible strategies to address them: such as the use of identity cloaking mechanisms (Beresford and Stajano 2003), by disassociating the identity of the user from accessed information (Ghinita e*t al.* 2007), or by designing systems and applications with de-centralized architectures, where users are in control of the location estimation (Priyantha *et al.* 2000) or disclosure process (Hong and Landay 2004).

With regard to the engineering issues, the reliability and accuracy problems of GPS in urban areas, where GPS signal shadowing occurs frequently, have been studied: Supporting methods have been devised such as *assisted GPS*, to enable GPS operation at lower satellite signal levels (Djuknic and Richton 2001); or *differential GPS* to increase the accuracy of the position estimate (Matosevic *et al.* 2006). Furthermore, alternative positioning methods have been proposed that promise to be better suited for urban or indoor environments than GPS, for example technologies that rely on known locations of wireless LAN (Borriello *et al.* 2005) or GSM cell tower radio beacons (Chen *et al.* 2006), or client-side systems based on *dead reckoning* – the process of estimating location by projecting heading and speed from a known past position, using data from various sensors such as electronic compasses, gyroscopes and accelerometers (Randell *et al.* 2003).

Other approaches alternatively seek to mitigate the impact of positioning inaccuracy on the user experience by informing the user about it rather than concealing it (Williamson *et al.* 2006), or by exploiting it at as a design resource that may, in fact, improve the user experience – for example in the context of multiplayer gaming (Benford *et al.* 2003).

The following sub-sections present history and relevant work from the field of location aware computing: Sub-section 3.1 introduces influential research projects which investigated scenarios and design principles for location- and orientation aware applications – the development of which this thesis subsequently aims to support; sub-section 3.2 discusses the state of the art in commercial location based applications, and presents the types of services that have become available on the market over the last years.

3.1 Research

Research on location aware computing has been actively pursued for several years. Noteworthy examples of location aware systems include the PARCTAB system (Want *et al.* 1995), *Cyberguide* (Long *et al.* 1996) and *GeoNotes* (Espinoza *et al.* 2001), all of which are presented in detail below, along with several other, related projects. Subsequently, an overview of research conducted on *orientation aware interaction* (i.e. work that has examined how location aware interaction can be augmented by using heading or tilt information gathered from sensors embedded in the mobile device) is given.

3.1.1 PARCTAB

Among the first research efforts to explore the notion of location aware interaction with handheld devices was the work conducted with the PARCTAB system. PARCTAB served as a testbed for the investigation of design and application issues of *Ubiquitous Computing* – a philosophy that originates at Xerox PARC. *Ubiquitous Computing* is based on the postulate that the "*most profound technologies are those that disappear*" (Weiser 1991) and aims to integrate computing infrastructure seamlessly into the environment "*by emphasizing context sensitivity, casual interaction and the spatial arrangement of computers*" (Want *et al.* 1995).

PARCTAB used palm-sized computing devices ("tabs") that were custom-built from commercial off-the-shelf hardware components. The tabs were designed for one-handed use, featured a 6.2 x 4.5 cm monochrome touch-sensitive display with a resolution of 128 x 64 pixels (corresponding to eight lines of text), three mechanical buttons, and a piezo-electric speaker for audio feedback. They were connected to the network via an infrared communication system and could be tracked by the system down to the resolution of a room. More than two dozen applications were developed and tested with about 40 people in order to investigate how location and other context information (such as time, the presence of other mobile or non-mobile computers, or the inferred presence of people), can be exploited to create systems that deliver information and messages more intelligently.

Among the key findings from their experiments with the PARCTAB system, the authors of (Want *et al.* 1995) report that users' interaction with the tabs was observed to be rather infrequent, and with short interaction times: Concluding, they therefore stress the need for more casual interfaces that enable the user to retrieve information at a glance, essentially replacing the *desktop metaphor* with a *wristwatch metaphor*.

3.1.2 CYBERGUIDE

Inspired by PARCTAB, and other related mobile computing projects such as the Berkley *InfoPad* project (Narayanaswamy *et al.* 1996), *Cyberguide* investigated the application scenario of mobile location aware personal guides. *Cyberguide* was realized using a commercial handheld device (the Apple MessagePad). Unlike the PARCTAB system, *Cyberguide* put primary focus on navigation and mapping functionality: as a key feature, it included a graphical map that could be scrolled and zoomed interactively. Two separate prototypes of the application were built: an indoor version that used infrared beacons for positioning (like PARCTAB); and an outdoor version where GPS was used to determine the location of the device.

In their discussion of future work, the authors of (Long *et al.* 1996) highlighted the need for a modifiable information base for *Cyberguide*, i.e. a mechanism that allows users to submit their own information and map annotations, as they are moving along, using the system. Their reference to this as "*a type of virtual graffiti*" later re-emerged as a common theme in several subsequent research projects conducted over the following years, such as *Graffiti* (Burrell and Gay 2001), *Urban Tapestries* (Lane 2003), or *InfoRadar* (Rantanen *et al.* 2004), to name only some of them.

3.1.3 GEONOTES

Among the earliest research projects that explored the concept of a “virtual information space overlaying the real-world”, where digital information is attached to physical objects or locations like virtual graffiti or digital post-it notes (Pascoe 1997), and mobile devices are used to interact with it, was *GeoNotes*: The system was initially implemented as a simulated system only (Espinoza *et al.* 2001), but later extended to be used on PDAs, using WLAN positioning; i.e. the user’s location was inferred from the physical location of the Wireless LAN hotspot the device was currently connected to (Persson *et al.* 2002).

In other noteworthy research efforts based on the notion of a real-world virtual information space – such as *Nexus* (Leonhardi et al. 1999), the *Lancaster Guide* project (Davies *et al.* 2001), or *Lol@* (Pospischil *et al.* 2002) – the mobile users’ primary role was to passively consume information. Content authoring was mostly expected to be performed by professional content providers such as art institutions, museums or tourist organizations. *GeoNotes,* on the other hand, centered on the vision of an open information space where ordinary users would equally act as consumers as well as contributors of geospatial data; so that it would grow, expand and develop with its users. In the words of the authors of (Espinoza *et al.* 2001) the information space would become more social, eventually reflecting the lives, concerns and social reality of its users – a vision that is in many respects reminiscent of the situation of the present day Web mapping and map mashup scene.

3.1.4 RESEARCH ON ORIENTATION AWARE INTERACTION

The research community continued to explore novel concepts for mobile interaction with physical space that go beyond location awareness: As mentioned in Section 1.2, innovative user interface metaphors like the Geo-Wand – a proposed portable device that allows users to obtain information about physical objects by pointing towards them – have been suggested as early as the late 1990. Along with the *Geo-Wand*, Egenhofer (1999) predicted an entire family of future geographical information assistants, which he termed *Spatial Information Appliances*: leveraging advances in GPS positioning and orientation sensor technology, these specialized tools for professional users and a public audience alike would enable interaction with geospatial information based on fundamentally different interaction metaphors. Examples given by Egenhofer include the *Geo Sketchpad* – a handheld location- and orientation-aware geospatial multimedia annotation device; the *Smart Compass*, a device which points users into the direction of certain points of interest; or the *Smart Horizon* that allows users to look beyond their real-world field of view.

Practical implementations of handheld applications that use heading sensors to extend location aware interaction have been demonstrated only in the last few years: Examples include the *M3I Navigation Platform*, a PDA-based pedestrian map and navigation system that is augmented with a GPS and magnetic compass, allowing users to identify landmarks by pointing towards them (Wasinger *et al.* 2003); *Real Tournament*, a location based multiplayer game where users collaboratively combat virtual opponents on a real world playing field using specialized game handsets equipped with a PDA, GPS and compass (Mitchell *et al.* 2003); or *gpsTunes*, a prototypical implementation of a PDA-based audio navigation tool that guides the user to a desired physical destination by adapting the balance and volume of the music currently played in the user's earphones, according to distance and heading from the destination (Strachan *et al.* 2005).

More recently, commercial efforts have emerged that make first steps towards bringing orientation-aware geospatial applications to the market: such as those of *GeoVector*[53] or *Intelligent Spatial Technologies*[54] (a company that originates from the research efforts and involved individuals of the original work on *Spatial Information Appliances*); and increased efforts are being made by several well-known mobile handset manufacturers to integrate navigation and sensor hardware – such as GPS and compass[55] or accelerometer-based tilt sensors[56,57] – into state of the art mobile phones.

3.2 Commercial Location Based Services

The *Global System for Mobile Communication* (GSM) family of wireless technologies represents the predominant mobile phone standard around the world, with a subscriber base exceeding 2.2 billion customers at the time of writing[58]. Due to the cellular architecture of GSM networks and their next-generation successor technologies like the *Uni-*

---

[53] http://www.geovector.com/

[54] http://www.i-spatialtech.com/

[55] GeoVector (2006) GeoVector and Cybermap Japan (Mapion) Deliver the World's First Pointing Based Solutions for Mobile Phones, *GeoVector press release* (http://www.geovector.com/press/mls.html)

[56] Samsung (2005) SAMSUNG Introduces World's First "3-dimensional Movement Recognition" Phone, *SAMSUNG's Digital World press release*, http://www.samsung.com/PressCenter/PressRelease/PressRelease.asp?seq=20050112_0000094230

[57] Nokia 5500 Sport (http://www.nokia.co.uk/nokia/0,,92079,00.html)

[58] GSM Association (2007) Subscriber statistics end Q1 2007, http://www.gsmworld.com/news/statistics/index.shtml

versal Mobile Telecommunications System (UMTS), they are inherently suited for providing localization information, which was understood even during the early days of GSM deployment (Drane *et al.* 1998). Regulatory pressure has been an additional major driving force behind localization technology in wireless networks: In the USA, the *Enhanced 911* (E911) requirement[59] mandates that emergency calls originating from mobile networks can be positioned with an accuracy of 500 feet (167m) in at least 95% of all cases (Uhlirz 2007); with a similar legislative initiative being pared in the EU under the name E112[60].

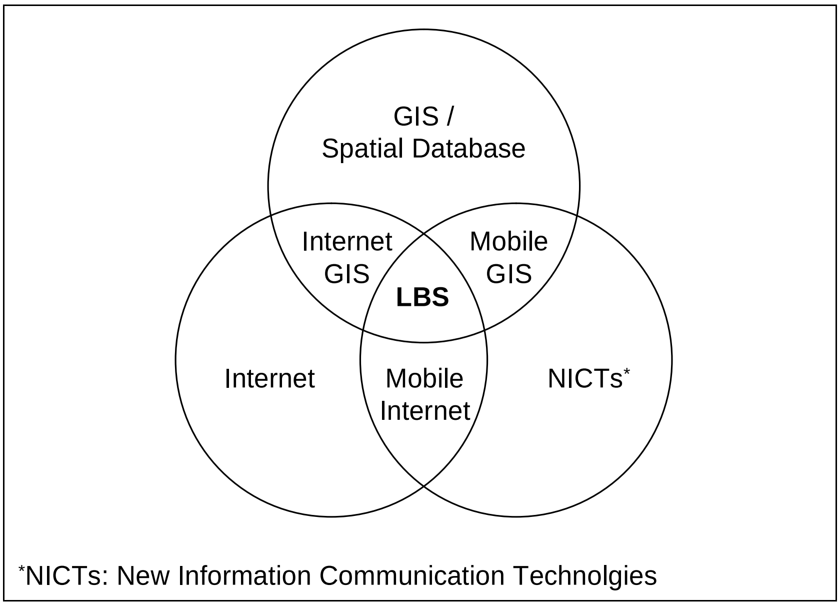


Figure 9. LBS as an intersection of technologies (Brimicombe 2002)

GSM network operators have since adopted *Location Based Services* (LBS) as an integral component of their service portfolio (Rao and Minakakis 2003): The authors of (Virrantaus *et al.* 2001) define LBS as information services that utilize "*the ability to dynamically determine and transmit the location of persons within a mobile network by means of their terminals*". From a system perspective, LBS can be considered an intersection of three technologies (Brimicombe 2002), as shown in Figure 9: they combine new information and mobile communication technologies (as access technologies) with

---

[59] http://www.fcc.gov/pshs/911/enhanced911/Welcome.html

[60] http://ec.europa.eu/environment/civil/prote/112/112_en.htm

GIS and spatial databases (as enabling technologies); and, more recently, the Internet (as transmission medium and communication infrastructure).

After the unexpected success of the Short Message Service (SMS) in GSM, LBS have been promoted as “the next killer application” since the first deployment phase of 3rd generation mobile communication systems in the late 1990s. Despite these expectations, wide-spread user acceptance and commercial success has, however, as yet been disappointing; and market forecasts have been overly optimistic (Uhlirz 2007). User studies nevertheless indicate that the demand for location based information is high in principle. Evaluations show that, while users are sensitive to the possible invasion of privacy that localization technologies may permit, their attitudes towards LBS are predominantly positive (Kaasinen 2003): People expect location aware information to be especially useful while in unfamiliar environments or in emergency situations. In particular, users appreciate access to information that is transient in nature, i.e. topical information that changes while they are on the move, such as traffic information.

More recently, mobile geospatial applications like maps, driving directions or local search are again being marketed as prominent features of top-of-the-line phone models such as the iPhone[61] or the Nokia N95[62]. The growing availability of low-cost GPS receivers has lessened the need for costly network-based positioning methods and alleviates the privacy problem, since localization is performed autonomously on the client device, rather than by an external, potentially untrusted entity. Their growing proliferation in more lower-end, consumer-oriented phones, combined with more affordable tariffs for mobile data and Internet services is expected to create a renewed momentum for LBS – and to promote new types of location based applications, such as social and community LBS[63,64], or location aware games[65,66].

Various attempts have been made to create classification schemes for LBS. No categorization has gained widespread acceptance so far, however. As suggested by the authors of (Virrantaus *et al.* 2001), the following sub-sections classify LBS according to their *functionality* and *utilization* of location information,. The classification attempts to capture the key application types that have emerged over the last years, illustrating them by concrete examples of specific products that are available on the market at the time of

---

[61] http://www.apple.com/iphone/internet/

[62] http://europe.nokia.com/phones/n95

[63] loopt (http://www.loopt.com/)

[64] Mologogo (http://mologogo.com/)

[65] The Shroud (http://www.shroudgame.com/)

[66] Locomatrix (http://www.futureplatforms.com/fp/clients/locomatrix/gps_gaming/)

writing. The list is not necessarily exhaustive, nor unambiguous: In fact, many applications today blur the boundaries between different application types by including elements from several categories. For example, local search applications may include navigation and wayfinding functionality; geo-annotation and mapping applications may provide media tagging or social and community features.

3.2.1 LOCAL SEARCH AND DIRECTORY SERVICES

Local search and directory services are a well-established class of LBS that are available on many network operators' service portals, or as SMS services. They provide users with information that is relevant to their current location, e.g. the city, district or street where they are located. Technically, these services utilize the location of the user to perform a spatially constrained search in a content database. The delivered information content may be static in nature (e.g. business listings such as nearby restaurants, tourist landmarks or various other facilities) or dynamic (e.g. the weather or public transport schedules); query responses may be textual only (e.g. in the case of SMS services) or include graphical maps as well.

Over the last years, several local search services have become available over the World Wide Web through 3rd party service providers who operate outside the mobile network operators' domain. As a means of positioning, these implementations either rely on client-side GPS positioning or, frequently, do not employ any form of automatic positioning at all: instead these (often free) services rely on the user to explicitly provide location information, e.g. in the form of a manually provided zip code or street address. It is disputable whether such applications truly qualify as "location based services": as stated above, the authors of (Virrantaus *et al.* 2001) stress the importance of "*the ability to dynamically determine […] the location of persons within a mobile network by means of their terminals*". In the strict sense, applications that rely on manually provided location hints could therefore not be considered LBS according to this definition. The author of this thesis, however, argues for a wider definition of LBS: According to Brimicombe's system view of LBS as an intersection of new information and mobile communication technologies, GIS, and the Internet, it can be considered permissible to regard these applications location based services. Similarly, Kaasinen (2003) distinguishes between *location based* and *location aware* services, where location based services are defined as "*services that are related as such or by their information contents to certain places or locations*"; and *location aware* services are considered a special sub-class of location based services "*that utilise the location of the user to adapt the service accordingly*". Building on these definitions, this thesis therefore specifically includes applications that rely on user-provided location hints such as zip codes or street addresses into its understanding of location based services. The terms *location based* and *location aware,* however, are used interchangeably: this is due to the fact that Kaasinen's definition of location based services, in the strictest sense, will include any non-mobile GIS or Web mapping application as well – which is strongly in contrast with common practice.

Practical examples for local search and directory services available at the time of writing include *Google Maps for Mobile, Microsoft Live Search Mobile*[67], *Yahoo Go*[68], or *Nokia Maps*[69], all of which support client-side GPS positioning as well as manually provided location (and all of which include navigation and wayfinding functionality in addition to local search as well).

3.2.2 NAVIGATION AND WAYFINDING

Navigation and wayfinding applications are a second class of location based service that has become commonly available: these applications provide users with driving or walking directions from their current location to a destination (which may either be provided manually, or as result of local search, etc.). Directions may be presented in various formats on the mobile device, e.g. as maps, as text, or as spoken turn-by-turn instructions.

Prominent examples of navigation applications include the *Wayfinder Navigator*[70], a map- and turn-by-turn-based navigation service for mobile phones (which is frequently marketed by mobile network operators as a re-named and re-branded product[71]); the *Verizon VZ Navigator*[72], a personal navigation application which includes local search and community features (such as sharing of maps); or *EZ NaviWalk*, a pedestrian navigation service offered by Japanese operator KDDI (Raper 2007a).

3.2.3 GEO-TAGGING AND COLLABORATIVE MAPPING

Local search, maps and navigation services are information services with a strict one-way information flow: data is provided by professional content providers, and the role of the user is limited to that of a consumer of content. Applications that encourage users to create and share their own geospatial content from their mobile devices, rather than

---

[67] http://livesearchmobile.com/windows_mobile.htm

[68] http://mobile.yahoo.com/go/local

[69] http://europe.nokia.com/A4509271

[70] http://www.wayfinder.com/?id=481&lang=en-UK

[71] http://www.a1.net/privat/a1navi

[72] https://vznavigator.vzw.com/index.html

just act as consumers – such as geo-tagging and collaborative mapping applications, where users create and exchange geographical information such as GPS recordings, annotations of points of interest or geographically referenced (“tagged”) media – have only emerged recently. Driven mainly by 3rd party service providers and *neogeography* initiatives, most of these services operate over the World Wide Web, outside the closed mobile network operators’ domains.

Two noteworthy examples for applications that allow users to record, annotate and share GPS tracks from their mobile phones are *SportsDo*[73], a GPS sports tracking application which allows users to share their tracking statistics over a Web portal; and the *Nokia Sportstracker*[74], which supports export of tracks and statistics in various formats, including GPX and KML. Two examples of mobile phone media tagging applications are Shozu[75], an application that can be used to upload photographs and videos to various online media sharing portals from the mobile phone, and automatically tag the uploaded media with GPS coordinates; or ZoneTag[76] an application with similar scope and features.

3.2.4 TRACKING

Among the more controversial uses of location awareness technology is tracking: tracking applications allow a third party to continuously monitor the location of one or more mobile clients from a desktop or browser application, or from another mobile device. The location of the monitored devices is either reported periodically, or on request by the monitoring third party.

Fleet management and vehicle tracking are common types of tracking applications offered by many mobile operators as part of their business application portfolio. More recently, tracking applications are also offered to private customers: examples include services for tracking stolen cars, as well as pet- or child-tracking services with special tracking-enabled phones (Schreiner 2007).

---

[73] http://www.sportsdo.net/

[74] http://research.nokia.com/research/projects/SportsTracker/index.html

[75] http://www.shozu.com/

[76] http://zonetag.research.yahoo.com/

3.2.5 SOCIAL AND COMMUNITY LBS

Social and community LBS are location based services that facilitate person-to-person communication within a group of peers or friends. Technically, the primary function of social and community LBS is often a tracking-like “buddy finder” functionality, where users can view the locations of friends within their peer group on a map. In general, however, social and community LBS follow a more privacy-observant mode of operation than traditional tracking services, where the monitoring party (i.e. the observer) is in control of the location disclosure process: in social LBS, users choose when and to whom to broadcast their location to. Typically, social and community LBS include a range of additional features from other application types such as local search or geo-annotation and media tagging.

Examples for existing social and community LBS include ***loopt***[77], a friend finder and collaborative mapping service that offers chat and messaging, tagging of favorite places and proximity alerts, as well as traditional social networking features like profile pages and journals; ***mologogo***[78], a friend locator and messaging service with local search functionality; or ***Buddy Beacon***[79] and ***Livecontacts***[80], two friend finder services with a more basic feature set.

3.2.6 LOCATION BASED ALERTS

Location based alerts or “push” services are services where the user’s location or proximity to another object triggers an event. Unlike the previously discussed application types (“pull” applications), push applications deliver information without an explicit, pro-active intervention on the part of the user. Examples of location based alerts are mostly found in location based marketing, where advertisements or discount coupons are sent to users when approaching a particular shop or restaurant (Virrantaus *et al.* 2001).

3.2.7 LOCATION BASED GAMES

A location based game is a computer game for a handheld device or mobile phone, in which the user’s physical location is used as a controlling element in the game mechan-

---

[77] http://www.boostmobile.com/boostloopt/

[78] http://www.mologogo.com/

[79] http://www.helio.com/#services_gps

[80] http://www.livecontacts.com/

ics, or where it influences the narrative of the game. Today, location based games are still considered a fringe activity that has yet failed to capture mass market attention (Rashid *et al.* 2006). There are only a few noteworthy exceptions such as the Japanese game *Mogi*[81], an urban treasure hunt game for mobile phones where users pick up virtual items located at real-world locations and trade with other players (Bell *et al.* 2006). The game was launched in 2003 and is still ongoing at the time of writing.

Further examples of existing location based games include *GPS Fishtrap*[82] a game where users deploy and pull "traps" at physical locations in order to catch virtual fish swimming by; *The Shroud*, a mobile role playing game with a 'GPS challenge' game mode where challenges are placed at nearby locations, and the user must find and complete them by physically moving there; and *The Journey*[83], a location based adventure game where the user advances the game's story by physically moving from one location to another in the real world.

---

[81] http://www.mogimogi.com/

[82] http://www.glofun.com/GPSFishtrap/GPSFishtrap_Introduction.htm

[83] http://journey.mopius.com/

4 Mobilizing the Geospatial Web

Sections 2 and 3 described the state of the art in two domains which – despite their close topical relationship – have so far remained separate worlds: the domains of end-user generated Web mapping applications, and of location aware mobile applications and LBS. This section discusses how the gap between those two domains can be closed, and how an end-user driven development process for LBS and new types of mobile geospatial applications can be enabled. First, it is discussed how existing geospatial Web content can benefit users on the go and, hence, why the mobilization of the Geospatial Web is a desirable goal. Secondly, some of the engineering challenges that currently pose a high barrier of entry into mobile application and LBS development are identified and investigated in detail. Finally, feasible strategies to overcome them are proposed, and the requirements for a generic application development framework for *spatially aware mobile Web applications* are derived.

4.1 Benefits

Without doubt, much of the geospatial data that is currently available on the World Wide Web is relevant for users on the go: Web maps, mashups, or virtual globe overlays available for download frequently contain dynamic data about real-time events that may be of interest in everyday situations, such as live public transport information[84], traffic flow data[85] or weather conditions[86]. Also, geospatial content on the Web is often much more niche-focused and potentially serves the needs of the intended target audience much better than generic operator-hosted LBS can: locations of independent bookstores[87], upcoming garage sales[88] or recently convicted crimes[89]; information about nearby sources of carbon emission[90] or other environmental pollution[91,92]; filming loca-

---

[84] HKL Public Transport Map (http://transport.wspgroup.fi/hklkartta/)

[85] Example: Google Maps with live traffic information (http://maps.google.com/?layer=t)

[86] guiWeather (http://www.guiweather.com/)

[87] Bookwormz (http://www.bookwormzonline.com/)

[88] Garage Sales, Yard & Estate Sales by Map (http://gsalr.com/)

[89] CrimeinDC.org: Crime Maps of Washington, DC (http://www.crimeindc.org/)

[90] Greenspace Carbon Emission Map (http://greenspaceresearch.uhi.ac.uk/greenspace/map.html)

[91] Find Pollution (http://findpollution.org/)

[92] Google Earth Gallery: pollutant data collected by the Commission for Environmental Cooperation (http://services.google.com/earth/kmz/cec_prtr_n.kmz)

tions of famous movies[93] or TV series[94,95]; or places with examples of early gothic architecture[96] are certainly not of interest to the average LBS user. Hence, it is more than unlikely that this information shows up in any network operator's service portfolio. However, it may be of interest to certain groups of people – and it is information that is readily available on the World Wide Web in the form of map mashups or virtual globe overlays. Mobilizing niche content that would otherwise be unprofitable and hence, unavailable in commercial services, can certainly be viewed as a benefit of the convergence between LBS and the Web.

Furthermore, geospatial datasets and applications on the Web are often edited and maintained by community initiatives or public institutions, rather than by a commercial entity, which may be inclined to provide biased information. Last but not least, the tighter integration of mobile access into the infrastructure of the geospatial Web not only benefits the information consumption process; it also benefits production and authoring: since mobile phones are always available to their users, geospatial data (such as map-annotations, geo-referenced photos or media) can be created spontaneously; real-world events can be captured the moment they happen, and information about them can be made available to a global audience instantly.

When a number of HCI experts from academia and industry were asked in a workshop about their views on how the convergence between mobile interaction and geospatial information can benefit users (Fröhlich *et al.* 2007), several recurring themes emerged in the discussion:

§ Support for *navigation* and *wayfinding* tasks while in unfamiliar environments was mentioned as a key benefit that can be enabled by spatially aware mobile computing devices. Desktop mapping toolkits like *Google Maps*, *Yahoo! Maps* or *Microsoft Live Maps* inherently support route planning and turn-by-turn guidance functionality; and mashup developers can easily integrate it with their own desktop applications. An integration of these existing Web development toolkits with the mobile development environment can therefore be considered highly desirable.

---

[93] The Google Maps Guide to Ghostbusters (http://www.ironicsans.com/2006/04/the_ultimate_interactive_googl_1.html)

[94] Dr. Who Locations (http://www.doctorwholocations.org.uk/)

[95] The Geography of Seinfeld (http://www.stolasgeospatial.com/seinfeld.htm)

[96] Early Gothic Architecture In France (http://jcm2044.net/Gothic/)

- Reference to how *social networking and community applications* can benefit from location aware mobile interaction was given several times: Finding of friends or family nearby, space-based recommendations or location based sharing of content were mentioned as examples for how spatial awareness can beneficially enhance the quality of person-to-person communication in mobile data services.
- Another, more general, recurring theme was the idea of "*expanding the interaction*" into the physical space, and the use of "*location as an interaction and input technique*": Much in line with the metaphors of *virtual graffiti* or *digital post-it notes* that had been expressed during the early days of research on location aware systems, the workshop participants envisioned examples of a "*real world Wikipedia*", where Wikipedia[97] articles are virtually attached to the physical objects they describe; or a "*real world Second Life*[98]" where the virtual playing field of a multiplayer computer game might overlay the real world environment around its users. Mixed reality techniques like the ones depicted in these examples (and potentially involving the use of orientation sensors) were seen as a beneficial enabler technology for future mobile applications that offer more compelling user interfaces and a superior user experience.

Summarizing, there are a number of benefits the mobilization of the *Geospatial Web* can bring to users: many of the information needs of users on the go – such as navigation or wayfinding support, different types of real-time geo-information, or niche content – are already covered be existing services on the Web. Moreover, the information provided is often maintained by communities of dedicated individuals or public institutions; and is in many cases made available for free. As experts approve, location aware interaction and mixed reality techniques furthermore promise to enable a considerably improved user experience over existing systems, in particular for entertainment and leisure applications such as in gaming or social networking. Last but not least, the convergence between mobile access and the *Geospatial Web* not only facilitates the consumption, but also the effortless production of geospatial content, by making its collection and distribution a seamless, instantaneous process.

---

[97] Wikipedia: The Free Encyclopedia (http://en.wikipedia.org/)

[98] Second Life (http://secondlife.com/)

4.2 Challenges

Despite these promising benefits that location aware mobile access adds to the utility of geospatial content on the Web for end users, there are considerable challenges to overcome for developers who wish to build the applications and tools to enable them: the implementation of location based services amounts to assembling "*a complex puzzle of disparate software, hardware, and connectivity components*" (Rao and Minakakis 2003). Traditionally, development of LBS has therefore been carried out in close cooperation with mobile network operators, either in contractual partnerships or through operator-specific 3rd party developer programs[99,100]. Today, no universal platform exists for the creation of mobile geospatial applications; no analogous mobile technology to Web mashups or mapplets is available to non-professional developers.

The following sub-sections discuss some specific problems that developers face today when designing, implementing and deploying mobile geospatial applications based on Web infrastructure: some of these problems are common to mobile application development in general; some of them are specific only to applications that deal with location aware presentation of geospatial content. The identified challenges subsequently serve as input to the application framework this thesis proposes in Section 5.

4.2.1 DEVICE FRAGMENTATION

Today, the Web is essentially a single-platform medium. Developers and user interface designers can center their creative efforts on the assumption that what they design and preview on their screens, is consistent to what users will perceive on theirs. In addition, the authoring philosophy of the web has always been implicitly developer-centric: In essence, a text-editor, knowledge of a few basic HTML tags and a browser for previewing the results are the only tools required to start developing. This predictability and simplicity has created a large community of non-technical, visually-oriented developers (Simon *et al.* 2005).

Mobile computing devices, on the other hand, differ substantially in form factor, screen size, resolution and aspect ratio, computing power, I/O capabilities and supported markup- and scripting standards: the high degree of predictability that exists on the desktop is no longer guaranteed. This first – and arguably foremost – challenge of mo-

---

[99] Verizon Wireless – The ZoN, developer Program,
http://www.vzwdevelopers.com/aims/public/menu/lbs/LBSLanding.jsp

[100] http://www.orangepartner.com/

bile application development, which is to ensure consistent presentation and optimized user experience despite device-specific differences, has been broadly acknowledged in the research community: A variety of new user interface authoring tools (Myers *et al.* 2000) and markup languages (Souchon and Vanderdonckt 2003) has been proposed to address the device fragmentation problem over the last years. In their note on *Authoring Techniques for Device Independence*[101], the World Wide Web Consortium (W3C) identifies three broad classifications of authoring techniques, which address the device fragmentation problem using different approaches: ***multiple authoring***, ***single authoring*** and *flexible authoring*.

Multiple Authoring. In the *multiple authoring* approach, the developer creates separate user interfaces for all types of devices that should be supported by the application. As a consequence, multiple authoring provides the maximum possible detail control over the result, since the developer can provide a tailor-made user interface for every device type and ensure the optimum user experience that is possible within the restrictions of the target device's display, I/O and computing capabilities. The obvious drawback of multiple authoring is the high production and maintenance effort, as well as a possible danger of inconsistency between different versions of the user interface. Also, there will not be a suitable user interface available for devices that were not explicitly addressed by the developer.

Single Authoring. In the *single authoring* approach, the developer provides a single implementation of the user interface that is valid for all platforms. Separation of the interface elements from their presentation is a basic prerequisite for the feasibility of this approach: by isolating the presentation information from the core description of the user interface elements, it is possible to generate alternative presentations based on the same core description (Trewin *et al.* 2003). Different approaches to single authoring are documented in literature, which address the device fragmentation problem at varying levels of sophistication: in the simplest case, single authoring can be achieved by providing a single description of the common parts of the user interface, and additional descriptions for the platform specific aspects; for example in the form of a common XHTML document together with platform-specific CSS or XSL stylesheets (Florins and Vanderdonckt 2004). While this approach ensures that user interfaces for different platforms are consistent at the level of information (a requirement which is also referred to as *thematic consistency*, and which is rated first in the W3C's *Mobile Web Best Prac-*

---

[101] http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/

tices[102] guidelines), it has been rightfully criticised for two reasons: first, there is still a high production effort required to create and maintain the device specific style information (Florins and Vanderdonckt 2004), which may in fact be in the same order of magnitude than the effort needed for *multiple authoring*. Second, it has been argued that the adaptability of such interface descriptions is limited insofar as they do not describe user interfaces in a sufficiently abstract form. They are therefore not applicable to arbitrary target platforms with distinctively different characteristics (Mueller *et al.* 2001), (Trewin *et al.* 2003): e.g. in the case of mobile devices vs. wall-sized displays (Myers *et al.* 2000).

An alternative approach to single authoring therefore aims for a higher degree of abstraction: *model-based* paradigms make use of high-level, declarative user interface specifications that define dialog and layout characteristics. From these high-level specifications, concrete platform-specific user interfaces are generated in an automatic or tool-supported process (Puerta *et al.* 1994) also referred to as *multiplatform generation* (Florins and Vanderdonckt 2004). Despite the fact that model-based approaches have a long tradition in academic research, they have never found wide acceptance in practice (Trætteberg *et al.* 2004). They have been criticised for the inherent lack of predictability that arises out of the high amount of automation involved, and a high threshold of use, since developers need to learn a new language for specifying the models (Myers *et al.* 2000).

Flexible Authoring. The *flexible authoring* approach is a combination of *multiple* and *single authoring*: For example, an application might follow a single authoring approach for most target devices, but offers specific tailor-made user interfaces for a selected subset of ‘preferred’ devices which should receive a particularly optimized user interface version; or for a certain range of target devices which differ radically from the majority of target devices; e.g. because they offer touch-screen operation rather than a typical 5-way-navigation button interface.

In addition to the aforementioned challenges, which are general to mobile interface design, the advent of mobile devices equipped with advanced navigation sensors introduces an additional degree of freedom to the user interface design process: Devices may potentially feature an arbitrary combination of sensors – from location-only (e.g. GPS) to full 3D orientation (e.g. by combining GPS with compass and tilt sensors). This complicates the development process, as the developer may not only need to adapt the visual

---

[102] J. Rabin and C. McCathieNevile (Eds.) (2006) *Mobile Web Best Practices 1.0*, http://www.w3.org/TR/mobile-bp/

appearance of the user interface, but may in fact need to adapt the interaction model: For example, a user interface based on direction – such as the Geo-Wand – is impossible to adapt to a device without integrated compass. The developer must therefore provide an alternative user interface that may differ substantially from the original one, such as a text- or map-based interface.

It can be summarized that the device fragmentation problem poses one of the key challenges for mobile application and user interface development in general. For the particular case of mobile geospatial applications, the emergence of mobile phones with varying navigation sensor sets introduces an additional uncertainty factor to the user interface design process: this uncertainty is potentially more differentiating than the factors that traditionally distinguish different mobile device types from each other – such as screen size or supported I/O mechanisms. This thesis therefore argues that a high-level abstraction mechanism at the user interface level is of particular importance for mobile geospatial applications: a common data model is needed for the delivery of local geospatial data, in order to avoid that application developers need to provide separate versions of their content for different types of location aware devices.

4.2.2 LIMITED COMPUTING CAPABILITIES AND BATTERY POWER

The limited computing resources of mobile devices represent a serious restriction to mobile application developers. Despite the fact that the reported design parameters of modern smartphones are on par with those of desktop computers of just a few years ago, the average state of the art mobile phone is still limited in practice: processors are comparably slow, and memory available for program execution is often restricted to a fraction of the total memory available on the phone (Simon 2006). This sets a boundary to the complexity of the applications and user interfaces the phone can support. It limits the interactivity and, consequently, the immersion that can be achieved with a given application. Correspondingly, the Web browsers which are pre-installed on mass-market mobile phones typically support only a very basic range of features: support for JavaScript is typically restricted to a minimum instruction set; often there is no scripting support available at all.

Furthermore, program execution, as well as the constant operation of the device display over longer periods of time, is power intensive: During the experiments with the PARCTAB system in the late 1990s, nominal use of the client device was considered to be in the range of 10 minutes per hour, 8 hours per working day. At this rate, the battery of the tab was easily able to last more than a day, with the typical tab only needing to be recharged once per week (Want *et al.* 1995). It can be expected that for present-day

LBS scenarios, the average time of use may far exceed this estimate. The demands on battery life will be proportionately higher. As explained by the authors of (Narayanaswamy *et al.* 1996), battery technology is, unfortunately, evolving at a comparably slow rate: the typical approach to improving battery life is therefore to re-engineer chips and systems for lower-voltage operation. As a further step, the authors suggest eliminating as much of the local computation on the client as possible, and migrate the power hungry tasks to servers in the network, where power consumption is of lesser concern.

Concluding, it can be summarized that due to memory, processing hardware and battery restrictions, mobile devices face serious limitations in terms of the application complexity and sophistication they can support; and the varying range of computing capabilities available on a different device models frequently represents an element of uncertainty in the development process that adds further to the device fragmentation problem described in Section 4.2.1. Approaches that shield developers from this uncertainty factor, as well as distributed application architectures that shift computationally intensive tasks into the network, are therefore desirable.

4.2.3 LIMITED I/O CAPABILITIES

It is reasonable to expect that the computing power, memory and battery life restrictions described above will become less severe as technology matures. There is, however, an inherent bound to the size that a handheld device can have, without becoming unhandy or impractical to the user. Thus, the screens of mobile devices – despite the fact that they are already beginning to rival their desktop counterparts in terms of display resolution – are likely to remain small in terms of their physical dimensions. Moreover, the size restrictions – combined with the fact that handheld devices are often used in dynamic environments and while users are moving – prohibit most input techniques that have become commonly accepted and successful on the desktop, such as the keyboard or the mouse (MacKay *et al.* 2005).

Consequently, a number of alternative mobile input styles have been devised. Often these are based on down-scaling an existing input technique: examples include small-sized full keyboards (occasionally in the form of a fold-out or slide-out keyboard), 18- or 12-button keypads that map multiple letters to a single button; scroll wheels or trackballs; touch-screen or stylus input. The setup which has, however, by far become the most common configuration until the time of writing is the combination of a 4-way directional controller element with a central ‘select’ button (this arrangement is also referred to as a “5-way controller” and is typically realized as a tiny joystick, or as five separate buttons, with four buttons arranged in rectangular fashion and the ‘select’ button in the center); a 12-button keypad; and two soft keys, i.e. two unlabeled, physical buttons below the display, whose function may change according to the current application context and is indicated by a label on the screen directly above the button.

The implications that the limited I/O capabilities have on the usability of mobile applications in general are an area of ongoing research (Kjeldskov and Graham 2003), (Zhang and Adipat 2005): For example, the effects that different levels of mobility (e.g. sitting, walking or standing) have on the execution of typical tasks performed on mobile devices like text entry (Mizobuchi *et al.* 2005), target selection or information navigation (MacKay *et al.* 2005) have been explored. Various possible mechanisms to overcome the challenges of limited screen real estate have been evaluated, including different 2D navigation techniques such as panning, zooming, overview & detail-, or focus & context views (Gutwin and Fedak 2004), (Burigat and Chittaro 2007), or through audio feedback to reduce the amount of required visual attention (Brewster 2002).

With regard to the domain of mobile geospatial applications and LBS, the discussed limitations raise various questions about how geospatial data can be effectively presented on handheld devices: for example about the amount of precision that digital maps need to provide in order to be useful (Agrawala and Stolte 2000); about the level of generalization that is appropriate for the design of maps for mobile devices (Dillemuth 2005); about how locations of interest that lie outside the viewport of the device can be best communicated (Burigat *et al.* 2006); or about how the relevance of geographical information can be assessed, so that mobile maps and LBS can provide information appropriately, "*presenting as much information as needed and as little as required*" (Reichenbacher 2007); and how this relevance can be modeled accordingly in an information system (Raper 2007b).

In addition to traditional maps, various design alternatives have been experimented with in the attempt to make navigation and location based information access on mobile devices more intuitive and usable: for example, Rakkolainen and Vainio (2001) presented a 3D city guide that supports mobile navigation with lifelike 3D models. They found that users perceived 3D representations as more recognizable than 2D maps because the visual similarity with reality helped them to find places in real life more easily. Similar findings were reported by Burigat and Chittaro (2005), who presented a prototype 3D tourist guide application on a PDA that uses GPS to present a location aware 3D visualization of the currently visited area. Other approaches to communicating geospatial information and navigational clues on resource- and I/O-constrained devices include the use of photographs (Cheung 2006) or panoramic landscape images (Miyazaki and

Kamiya 2006), textual, audio or symbolic directions (such as arrows), or combinations thereof (Burigat and Chittaro 2007).

Summarizing, it is concluded that the limited I/O capabilities of mobile devices raise a number of questions with regard to the cognitive and perceptual issues of human-computer interaction with mobile applications in general, and mobile geospatial applications in particular. The wide-ranging consequences these questions have on the usability of different geospatial presentation formats are, however, beyond the scope of this thesis. Consequently, the author attempts to make as few assumptions as possible on what constitutes a well-designed mobile geospatial user interface and what does not: Instead, the goal of this work is to provide a framework and conceptual model for prototyping of arbitrary forms of geospatial interaction; thus fostering the experimentation with new interaction styles and metaphors.

Despite this approach, the thesis is without doubt inspired by certain fundamental interaction principles which guide the design of the framework to be proposed in Section 5: In particular the thesis embraces the principles of orientation aware interaction and the notion of the *Spatial Information Appliance* introduced by Egenhofer (1999), as well as several fundamental assumptions about the HCI principles of mobile interaction, some of which have been identified as early as in the first ubiquitous computing research efforts in the late 1990s:

- that mobile interaction is often infrequent and with short interaction times, supporting the claim that mobile user interfaces need to be casual and enable users to retrieve information at a glance, "*replacing the desktop metaphor currently in use with a wristwatch metaphor*" (Want *et al.* 1995);
- that mobile interaction frequently happens as a secondary task while users are moving, again supporting the argument that user interfaces should provide low cognitive overhead and, furthermore, support one-handed operation.

4.2.4 LOCATION AWARENESS: INTEGRATION ISSUES

Positioning, i.e. the acquisition of information about the mobile client device's location, is obviously the foremost requirement for building location aware applications. Low-cost standalone Bluetooth GPS receivers, as well as PDAs and smartphones with integrated GPS have recently made it possible for developers to include location in their applications without resorting to costly mobile network operator-provided location services. Accordingly, location based services based on GPS positioning are becoming increasingly common on the market (Schreiner 2007). To date, however, it takes considerable effort to integrate GPS with mobile applications: Dedicated client-side software is required to obtain location coordinates and communicate them to the application server (Simon and Fröhlich 2007). Native code or Java development is, however, a skill that the majority of Web developers are unfamiliar with; the enthusiast community of Web map and mashup creators is therefore largely debarred from the development of GPS-enabled applications. This can potentially be seen as a major hindering factor for the quick mass-market adoption of mobile geospatial Web applications.

Irrespective of the implementation complexity involved with integrating GPS positioning into mobile applications, there is a potentially more severe barrier that hinders the development of location aware mobile applications: despite the growing proliferation of GPS-enabled devices, the majority of mobile phones is so far not equipped with GPS; and it is unreasonable to assume that average users will take the time, cost and effort required to obtain, connect and use an external GPS accessory. Furthermore, GPS is typically ill-suited for heavily urban or indoor environments, as has been explained in Section 3. In order to support more than just a minor share of mobile devices on the market, and offer robust service in all types of environment, developers will therefore need to support alternative positioning methods in addition to GPS, such as the network-centric positioning methods that are used by operators to determine the position of GSM phones in the network (Drane *et al.* 1998).

Beyond these operator-controlled positioning infrastructures, a number of alternative, de-centralized positioning techniques has recently become more widespread: these methods involve a client-side software component which listens for signals from nearby GSM cell towers, wireless LAN hotspots or other “signals of opportunity” (Raper *et al.* 2007) to determine the geographic location of the client device: Noteworthy efforts in this direction include *Place Lab*, a system that works by listening for transmissions of wireless networking sources such as 802.11 WLAN access points, fixed Bluetooth devices and GSM cell towers (“beacons”) and comparing their unique or semi-unique IDs against a *beacon database*, which lists locations for known beacons and which is partially cached on the mobile device (LaMarca *et al.* 2005); the aforementioned *ZoneTag* (compare Section 3.2.3), a mobile photo sharing application which uses an online directory of GSM cell tower IDs to infer the location of photos taken with a camera phone (Ahern *et al.* 2007); as well as several collaborative efforts on the World Wide Web which aim to build universal repositories and lookup services for WLAN hotspot IDs[103] or GSM cell IDs[104]; or commercial ventures with similar goals[105].

As a consequence, this thesis argues that in order to ease the integration of location awareness into mobile applications, it is necessary to address two concrete issues:

§ As GPS promises to become a widely available technology on mass market devices, the integration of GPS into the Web application environment must be seamless: In order to empower the quick adoption of mobile geospatial Web applications, a standardized format for communicating the location of a GPS enabled device from the mobile browser to the application server is needed, e.g. in the form of a HTTP header or query parameter.

§ For the immediate future, however, GPS can not be assumed to be universally available on mass market mobile phones. Moreover, GPS will continue to be ill-suited for certain environments such as indoor locations. It is therefore necessary to provide supporting infrastructure, which offers alternative mechanisms for inferring the user's location (e.g. from signals of opportunity) without introducing additional implementation overhead for application developers.

4.2.5 LEGACY COMPATIBILITY

As the authors of (LaMarca *et al.* 2005) note, location aware computing today is caught in an unfortunate cycle: due to the technical barriers faced by developers, few applications exist in practice. Hence, there is only a small user base for present-day LBS. On the other hand, infrastructure providers as well as handset manufacturers are not interested in investing in a field where there is little perceivable demand.

The key motivation behind this thesis is the assumption that a strong, enthusiast community of active LBS developers can break this cycle – by building a critical mass of applications, and by driving innovation in the field of location aware computing in a similar fashion that the *neogeography* community has innovated the field of Web mapping. In order to support this process, two goals must be met: first, the technical entry barriers – such as the device fragmentation problem or the challenges related to the integration of location awareness technologies – must be minimized, as discussed earlier. That way, LBS development becomes viable to a larger group of people. Secondly, it is

---

[103] http://www.wigle.net/

[104] http://gsmloc.org/

[105] Skyhook Wireless, How it Works, Overview, http://www.skyhookwireless.com/howitworks/

necessary to make LBS development more attractive to developers – by maximizing the potential audience they can reach as users: Today, the majority of mobile phone users still carry low-cost mobile phones without advanced presentation and connectivity features ("legacy phones"), rather than high-end smartphones. In the worst case, such devices cannot be assumed to provide any except the most basic features mandated by the GSM standard – i.e. the ability to place voice calls, and to send and receive SMS short messages. Consequently, it is necessary to foresee fallback mechanisms which make mobile geospatial applications accessible from such devices as well. Even if these fallback mechanisms induce a reduction of the perceived quality of the user experience, they will at least ensure that applications are immediately available to all mobile phone users, and not just the minority with compatible high-end devices.

4.2.6 LACK OF BEST PRACTICES

The relatively low number of existing LBS that are successful on the market (Raper 2007a), and the fact that user acceptance for LBS is still at marginal levels in practice (Uhlirz 2007) can be seen as an indication that the critical success factors for what distinguishes a successful LBS from an unsuccessful one are yet to be fully understood. For the near-term future, the development of a set of tried and tested guidelines for the various steps involved in the design of a mobile geospatial application will therefore be of particular importance. Efforts to agree on a set of commonly accepted best practices for mobile Web development in general have only emerged recently: such as the aforementioned ***Mobile Web Best Practices*** defined by the W3C's Mobile Web Initiative[106], or the ***Mobile Web Developer's Guide***[107] and the ***dotMobi StyleGuides***[108] published by the mTLD (mobile Top Level Domain), the appointed global registry for the ***.mobi*** domain (a top level domain which is dedicated to users who access the World Wide Web from mobile devices). Design guidelines that also take into account the intricate properties of location based applications – such as the usability problems discussed in Section 4.2.4, or the privacy and anonymity issues that are closely linked to many application scenarios in location aware computing – are yet missing, however. Unsurprisingly, 6 out of 12 research items that are identified by the authors of (Raper *et al.* 2007) as "urgent research needs" on a future LBS research agenda relate to these fundamental application design issues: architecture, interaction and user interface design, as well as privacy considerations (specifically: wayfinding interfaces for LBS/in-car navigation, interaction

---

[106] http://www.w3.org/Mobile/

[107] http://dev.mobi/node/197

[108] http://dev.mobi/node/423

design of LBS, map vs. virtual/augmented interfaces to LBS, LBS architectures and platforms, location privacy, location based social networking).

As already indicated in section 4.2.3, solving the wide ranging design issues that urgently need to be addressed in near-future research on location aware systems is beyond the scope of this thesis. Rather the goal of this work is to provide:

- a conceptual model and a possible architecture for future location based applications built on Web standards;
- an open experimentation platform that enforces as few restrictions on interaction and user interface design as possible, so that it can consequently serve as a tool for further research on evolving LBS design issues and best practices.

4.3 Approaches

The previous subsections discussed some of the benefits of mobile access to geospatial Web content, as well as some of the challenges that application developers are facing today when creating geospatial applications for mobile computing platforms. In this subsection, conclusions will be drawn from these discussions, and global requirements will be derived that consequently serve as input to the design of an application framework, which is presented in Section 5. The framework thereby addresses the issues identified in the aforementioned discussions on three different levels – each corresponding to different roles in the creation process of mobile geospatial applications: the *content authoring* level, the *software engineering* level, and the *interaction design* level. Likewise, each of the three levels pertains to a different target audience (who may or may not be represented by the same persons): *content creators*, who author and maintain the content base behind an application or service; *application developers* who provide the service functionality and business logic; and *user interface designers* who are responsible for the interaction design and user experience on the client device.

4.3.1 HARMONIZATION

As discussed in Section 4.1, numerous sources on the World Wide Web provide geospatial content that is relevant to mobile users. In order to make this existing information re-useable for mobile application developers, the foremost requirement for the framework developed in this thesis is therefore interoperability with the prevalent geospatial data formats on the Web: The *GeoStack*, which is currently emerging as a reference model to organize the various technologies and formats involved with the data management lifecycle of geospatial information (compare Section 2.2) already broadly acknowledges the role of mobile devices as a means to capture and record geographic content, and as a means to consume this information on the go – e.g. in the form of waypoints or route directions. A concrete engineering model, toolkit or library implementation that allows end-users to create their own mobile applications, rather than just use professionally developed products is, however, still missing in the *consume* step of the *GeoStack* (compare Section 2.2, Figure 2).

On the content authoring level, this thesis therefore stresses the need for harmonization with the emerging standards that comprise the *GeoStack*, so that existing geospatial Web content, as well as the tools and the expertise to produce, maintain, aggregate and distribute it, can be re-used; and content authors do not need to go through extra effort to make their content suitable for location aware access. Consequently, the framework shall act as an integration platform for different geospatial formats and content sources such as Virtual Globe overlays or Mapplets. Furthermore, the framework's role shall be to complement the *GeoStack*, by providing the appropriate toolset for end-user-generated mobile geospatial applications that is yet missing.

4.3.2 ABSTRACTION

As detailed in Section 4.2, many of the engineering challenges involved with the development of mobile geospatial applications relate to the varying characteristics and capabilities of mobile devices. With regard to the software engineering issues, a first important requirement is therefore the introduction of an abstraction mechanism at the user interface level that avoids the necessity for excessive "multiple authoring" effort (compare Section 4.2.1), i.e. which prevents that application developers must provide a large number of device specific output channels for their application.

In addition, the abstraction mechanism shall suitably address the challenges associated with the problems of limited computing and I/O capabilities discussed in Section 4.2.2 and Section 4.2.3, respectively: it should suit "thin client" and legacy architectures, where processing and user interface composition is performed primarily at the server (for example in the case of a typical browser-based application, where the user interface is realized as a WML, HTML or XHTML page, or in the case of an SMS-driven legacy scenario, where the user interface is restricted to the exchange of short text messages). At the same time it should not limit developers in creating richer, more interactive user interfaces on more sophisticated devices; for example sensor driven interfaces based on the notion of the *Spatial Information Appliance*. Note that the terms *cross-platform scalability* and *generality* have previously been used by the author of this thesis to refer to the ability to equally support thin as well as rich clients, and to the ability to support various alternative forms of geospatial user interfaces beyond maps (Simon *et al.* 2007b), without the need for extra software engineering effort on the application side.

A second important software engineering requirement, which can be derived from Section 4.2.4, is the need for an abstraction mechanism at the API level that hides the implementation and integration details of the employed positioning technology: even if the client device does not feature GPS, the framework should be able to provide a "best effort" position estimate using alternative methods. This process should be transparent, i.e. the application need not necessarily be aware of the fact that the device does not feature integrated GPS, and that the framework uses other means to infer its location.

4.3.3 PROTOTYPING

As discussed in Section 4.2.3, applying classical desktop user interface metaphors to size- and I/O-constrained devices poses considerable problems with regard to the usability of mobile applications. Beyond these user interface related problems, there is a general lack of established guidelines and best practices also in other relevant areas of application design (e.g. with regard to the design of appropriate privacy mechanisms and location disclosure policies) as detailed in Section 4.2.6. The author argues that in order to stimulate the development of more usable mobile geospatial applications, and to foster the evolution of a commonly agreed set of LBS best practices, tools and APIs for flexible and rapid prototyping are crucial: Already during the earliest research on Ubiquitous Computing it was found that "*experimental infrastructure that allows users to prototype new application ideas*" is capable of acting as "*a catalyst in generating new ideas in the area of Ubiquitous Computing*" and, hence, can inspire further, novel applications (Want *et al.* 1995). The same view is shared by experts from the HCI community, who acknowledge the need for prototyping as a means to demonstrate the benefits of mobile interaction with geospatial information, and to discover "*what people want to interact with and how*" (Fröhlich *et al.* 2007).

Moreover, location based applications are a class of applications that is heavily distributed and componentized by nature (Rao and Minakakis 2003). They are dependent on various associated services such as wireless connectivity and positioning, each of which may be subject to physically imposed limitations like intermittent availability or poor signal quality. Due to these reasons, many LBS as designed would not perform acceptably in practice; and, as Raper (2007a) has stated, the arising performance issues can only be solved by compromises on data models and user interfaces once the system as a whole can be evaluated in use.

Consequently, the framework developed within this thesis shall act as an experimentation toolkit for rapid prototyping of various kinds of mobile geospatial applications: It should pose a low barrier of entry to designers previously not involved with the design of geospatial user interfaces and, in particular, non-professional and hobbyist designers. It should provide a *low threshold*: i.e. it should allow developers to achieve first results and basic prototypes quickly, without confronting them with a steep learning curve. At the same time, it should provide a *high ceiling*: i.e. it should not constrain experienced designers in producing sophisticated applications.

In particular, the framework's goal is to enable and encourage user interface designers to best exploit the capabilities of devices to provide an enhanced user experience (a requirement which is listed under the keyword of *capabilities*, and which is rated second in the W3C's Mobile Web Best Practices guidelines). Novel orientation aware interaction metaphors are considered of particular importance in this context: They are becoming more and more common on the market (compare Section 3.1.4), and offer a potentially promising answer to the I/O restrictions of contemporary mobile geospatial interfaces (Fröhlich *et al.* 2006), (Fröhlich *et al.* 2007).

5 A Development Framework for Mobile Geospatial Web Applications

The previous sections have explained the motivation for this thesis: that it is a desirable goal to close the gap that exists between end-user generated Web mapping applications, and the domain of mobile location aware applications. This section discusses the current Web application architecture and proposes the addition of four distinct extension components in order to achieve this goal: a *positioning support component*, a *content aggregation component*, a *query component for ego-centric spatial queries*, and a *presentation and formatting component*. The internal architecture of the extension components is described; the interfaces and the processing flow between them are explained; and their relation to the three engineering approaches defined in Section 4.3 is discussed.

5.1 Extending the Web Application Architecture

Web applications evolved from *Web sites* or *Web systems*. While *Web sites* are distributed hypermedia systems that enable users to access documents stored on the file systems of other computers using a browser, *Web applications* extend this model by adding business functionality; i.e. a *Web application* can be described as a Web system that uses a *Web site* front end for executing server-side business logic from the browser (Conallen 2002). As explained in detail by Conallen (2002), Web applications use *enabling technologies* to make their content dynamic, and allow users to affect the state of the business logic on the server, e.g. by entering a varied range of data such as text, list or check box selections, or by submitting binary and file data from their browser.

Today, two major approaches to Web application enabling technologies exist: compiled modules and interpreted scripts. Compiled modules are loadable binaries executed by the server. The modules have access to APIs that provide information about incoming requests, and produce HTML output that is sent back to the client in response. Essentially, compiled modules intercept the request/response processing flow in front of the normal Web server's processing. Popular implementations of this approach include Microsoft's Internet Server API (ISAPI), Netscape Server API (NSAPI) and Java Servlets (Conallen 2002). The second category of enabling technologies – scripted pages – are HTML pages that contain scripts, special tags or tokens that can be interpreted by an application server. The scripts can interact with objects on the server; the resulting output is embedded back into the HTML page. Popular implementations of this approach include Java Server Pages (JSP), Microsoft's Active Server Pages (ASP) or PHP (Conallen 2002).

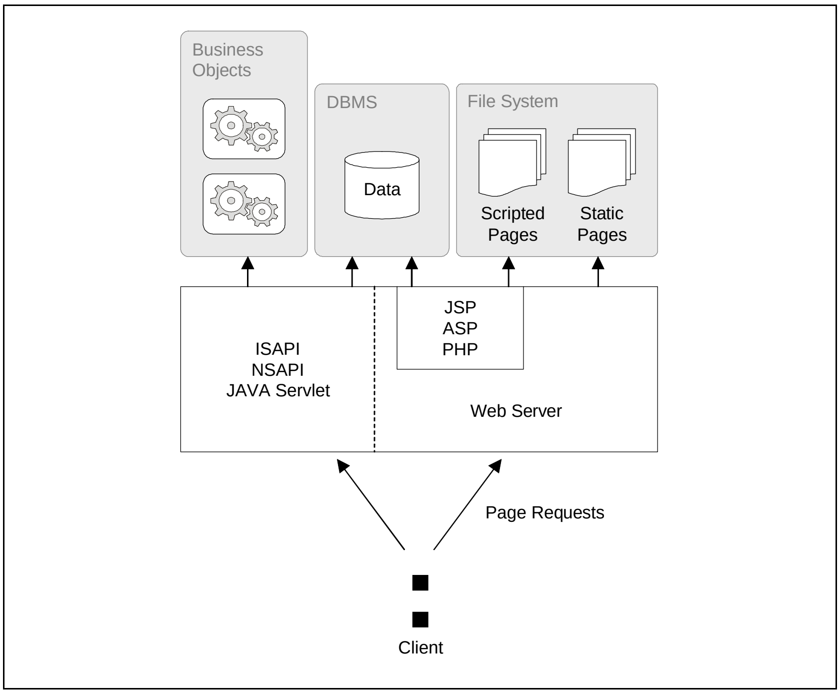


Figure 10. Web server and Web application enabling technologies

Drawn from the information provided by Conallen (2002), Figure 10 shows how client browser, Web server, and enabling technologies together comprise the state of the art Web application architecture. This model forms the basis for the mobile geospatial Web application framework proposed in this thesis: the framework is implemented as a set of business objects that reside in the server domain, and which perform actions that are common to the processing flow of mobile geospatial applications – such as spatial querying of data sources in different encoding formats, or device-specific user interface adaptation and formatting. The following sub-section describes the individual business objects, their internal structure, and the interfaces between them in detail.

5.2 Framework Overview

In order to support the development of mobile geospatial applications based on Web infrastructure, this thesis proposes the following four additions to the established Web application architecture:

- A **content aggregation component** that mediates between different formats for POI and waypoint data on the World Wide Web, such as for example KML, GeoRSS or GPX; and which carries out transformations between different geographical reference systems, if necessary. Accordingly, this component enables interoperability with the *GeoStack*, thus satisfying the first global requirement defined in Section 4.3.
- A **query component** that handles the process of selecting the relevant sub-set of geospatial information from the information base, according to the user's location and other request parameters. Taking into account the third global requirement defined in Section 4.3 – prototyping support – special consideration in the design of the query engine is thereby given to the properties of sensor-driven interaction metaphors such as the Geo-Wand: Unlike existing systems, the proposed framework uses a unique query mechanism that includes information about the *visibility* of geographical features, as seen from the query location. The details of the query process and the *Local Visibility Model* (LVis, pronounced "Elvis") – a novel XML data format used to encode the query result – are presented in Section 6.
- A **presentation and formatting component** which optionally transforms the query result into a device-specific presentation format (such as a Web page adapted to the screen size of a specific mobile phone model); or which translates the result into a textual format for legacy devices communicating over a messaging medium such as SMS. Along with the LVis query result format, this component is instrumental in fulfilling the requirement of *user interface abstraction* as defined in Section 4.3.2.
- A **positioning support component** that serves as an abstraction layer for different positioning technologies: In case the client device does not feature GPS, the positioning support component will make best effort to infer a geo-coordinate using alternative mechanisms, for example by attempting to geo-code a user-provided street address or landmark name. This component directly addresses the requirement of abstraction at the positioning API level, as defined in Section

4.3.2 and serves as an important enabler for achieving compatibility with legacy devices, as discussed in Section 4.2.5.

Figure 11 depicts the internal architecture of the proposed framework. It shows how the query component forms the central element of the framework, and how the Local Visibility Model acts as the common base-format for different types of output channels: it is either sent directly to the mobile client, where it can be interpreted by different types of spatially aware browsers (e.g. a Geo-Wand or a Smart Compass), or custom “spatially enhanced applications” (for example an orientation aware mobile game). Alternatively, the LVis is channeled through the presentation and formatting component, and possibly through one the different messaging interworking functions (IF), where it can be transformed into various formats, according to the capabilities of individual client device.

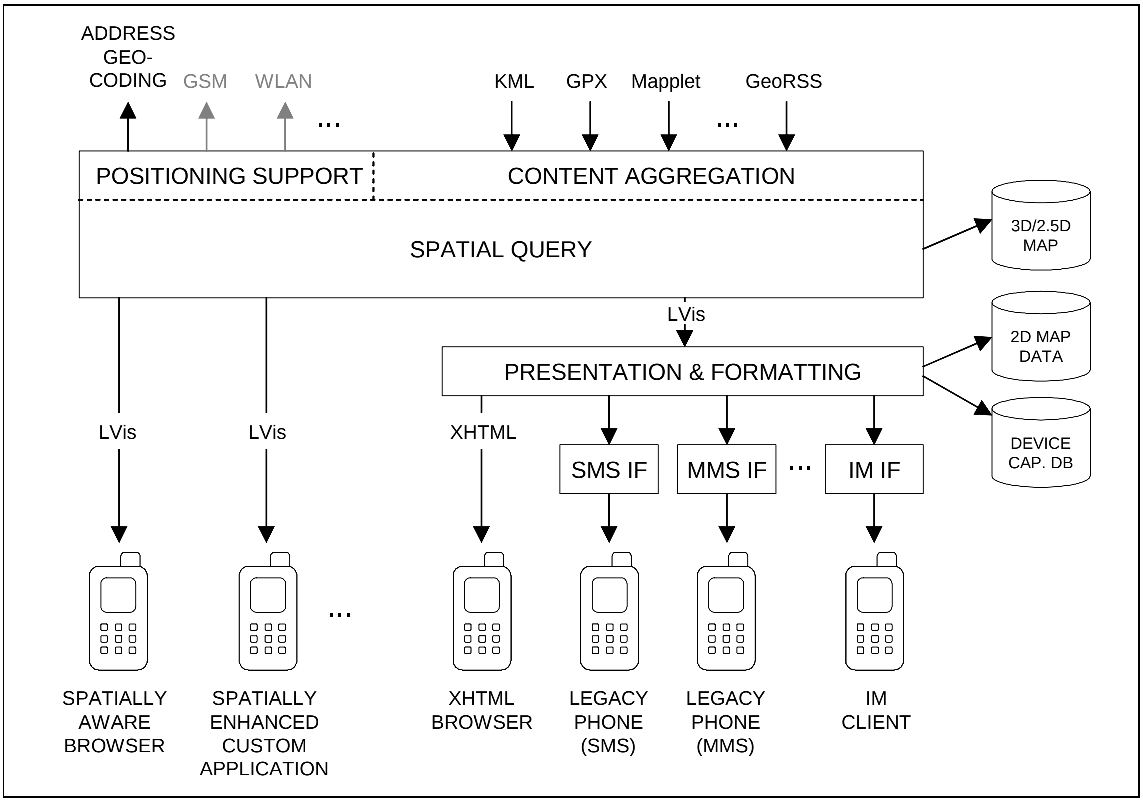


Figure 11. Framework architecture overview

With regard to the established Web application architecture discussed previously (compare Figure 10), the query engine resides in the server domain as a compiled module. It handles incoming HTTP requests in lieu of the Web server. Unlike the Web server, however, which will always produce HTML in response, the query engine responds with an LVis, and may use the presentation and formatting component and messaging interworking functions (which reside in the server domain as business objects, together with the business objects that implement positioning support and content aggregation) to optionally produce alternative, device-specific output. Further elements shown in Figure 11 include:

- a three-dimensional model of the environment, which is required by the query engine for the visibility-based query process (refer to Section 6 for more details);
- a map image data source used by the presentation and formatting component to generate graphical maps from LVis query results;
- a repository storing device capability information in order to enable the presentation and formatting component to carry out device-specific formatting tasks (e.g. map sizing).

The following sub-sections present all of the components depicted in Figure 11 in detail.

5.2.1 CONTENT AGGREGATION COMPONENT

Section 2 discussed the various data formats that have evolved in the domain of Web mapping to encode geospatial content of different types. The individual formats differ in terms of their thematic scope (e.g. a path or location described in GML represents an arbitrary spatiotemporal entity, while GPX on the other hand, implies that the content represents a recorded GPS track; i.e. it inherently associates specific semantics with the content) as well as in their syntactic feature set (e.g. GeoRSS encodes only the content, while KML may include graphical styling information as well). Nonetheless, the formats described in Section 2 share a common base feature set, in the sense that each format in general provides the syntax to express:

- **locations**, such as navigation waypoints or other points of interest (this applies to all formats, including the GeoRSS W3C Geo encoding and Microformats);
- **paths** consisting of line segments;
- and **polygons** (this applies to GML, GeoRSS Simple and GML encodings, KML, as well as Mapplets).

Based on this “least common denominator”, the function of the *content aggregation component* is therefore to translate between various syntax formats used to express geographic feature information, and allow the framework’s central element – the query component – to seamlessly integrate with the *GeoStack*.

5.2.2 QUERY COMPONENT

Spatial databases are database systems that offer support for spatial data types such as those provided by the data formats of the *GeoStack* (i.e. points, lines or polygons) in their data model and query language. Query operations supported by spatial databases are based on *spatial relationships*: *Spatial relationships* make it possible to select elements from the database with a given relationship to a *query object*, e.g. all elements inside a window. As explained by Güting (1994), relationships can be of topological nature (such as *inside*, *adjacent*, or *disjoint*); they can be directional (e.g. *above* or *North of*), or metric (e.g. "distance<100"). Accordingly, spatial databases organize geographical data in a *panoptic* or birds-eye view, which is suitable for these types of spatial relationships; and which is also useful for generating maps – a form of presentation which is inherently panoptic.

A mobile user, however, faces a uniquely different view of the world: instead of adopting a birds-eye view, the mobile user is physically immersed in the geographical region that is associated with the spatial dataset or search space. A *deictic* frame of reference, i.e. a reference frame that is relative to each individual user, can therefore be considered more appropriate for the mobile context than an *extrinsic* reference frame that is established independently of the situation and orientation of the observer, as noted by Gardiner and Carswell (2004). This claim is in accordance with Raper[109], who criticizes that the panoptic view of present day spatial information systems neglects "*the individual perspective in favour of the 'map' overview*". The same opinion is shared by Frank (2003), who proposed the inclusion of egocentric spatial relationships into the query language of spatial databases.

In an article which discusses potential research directions in cartographic representation, the authors of (Fairbairn *et al.* 2001) stress the need to "*advance ways of transforming information about the world into models suitable for digital and cartographic representations that are designed for, and facilitate, effective visualization. Such models should ensure fitness for use; should draw on research into the cognitive issues that surround increasingly personalized and flexible possibilities for map use with an expanded range of map forms; should respond to the state of the art in the realm of interfaces; and should drive (and respond to) developments in the field of databases and geocomputation.*" Building on these recommendations, this thesis therefore proposes a novel query mechanism for supporting alternative forms of mobile interaction with geospatial data:

---

[109] J. Raper (2002) GIScience 2002 Keynote, http://www.soi.city.ac.uk/~raper/research/GIScience2002-OHs-pub.ppt

- Based on an *egocentric* query principle, the query mechanism delivers a spatial model of the user's individual perspective on the result space, rather than a panoptic overview, as demanded by Raper, and the authors of (Frank 2003) and (Gardiner and Carswell 2004).
- Attempting to satisfy the requirements stated by the authors of (Fairbairn *et al.* 2001), the query result data model is flexible and suited for a variety of graphical and non-graphical user interfaces and representation formats, in particular those based on the orientation aware interaction metaphors of *Spatial Information Appliances*. Thus, the data format serves both as a means to achieve user interface abstraction (as demanded in Section 4.3.2) as well prototyping support for sensor-driven interfaces (as demanded in Section 4.3.3).

In order to provide the reader with a complete understanding of how the proposed query mechanism implements these requirements, Section 6 is entirely dedicated to the discussion of the LVis – the query result data format which forms its core conceptual element.

5.2.3 PRESENTATION AND FORMATTING COMPONENT

As discussed previously, the I/O capabilities and limitations of the target device may allow for (or mandate) different ways of presenting the query result to the user: e.g. in the form of 2D or 3D maps with different levels of generalization and abstraction, as panoramic landscape images or textual descriptions. The role of the *presentation and formatting component* is to act as a server-side adaptation layer that transforms the generic LVis into a suitable device specific representation. It has to be noted that the *presentation and formatting component* is optional in the application request/response path: As shown in Figure 11, the framework may also return the LVis directly to the client device over the wireless link, where it can be processed locally, e.g. to realize sensor-driven user interfaces. For most state of the art mobile phones however, it is preferable to pre-process the generic LVis query result on the server before returning it over the wireless interface. Not only does this reduce the amount of (potentially battery intensive) local processing; it also eliminates the need for dedicated software that must be installed on the client device, since the query result can be viewed in the phone's native Web browser.

The primary functionality provided by the *presentation and formatting component* is therefore to translate the LVis into mobile-friendly XHTML or WML markup and/or a graphical map image. Since the LVis itself does not contain map data, but only the georeferenced content to be overlaid on the map, an external data source that provides base map images, e.g. a WMS-compliant GIS, is needed (see Figure 11). In order to optimize map-size, as well as other presentation and layout characteristics of the result for the various different mobile devices, the *presentation and formatting component* furthermore relies on a device description repository, i.e. a lookup directory for the key characteristics of the various mobile phone models found on the market (e.g. screen width and height, supported markup standards and image formats).

5.2.4 MESSAGING INTERWORKING FUNCTIONS

Generally, it cannot be assumed that all mobile users will be readily able to access the World Wide Web from their devices: in many cases, technical limitations of the device (or the lack of a suitable mobile phone contract) will prohibit access to Internet services; in other cases where access to the Web is possible in principle, users might deliberately choose not to use data services in order to save costs (e.g. in case the user is traveling in a foreign country and would face high data roaming charges). The *messaging interworking functions* are the framework's primary support components that enable service access in such cases, using various alternative communication channels. For example, an *SMS interworking function* can respond to incoming SMS queries by translating the LVis query result into an appropriate SMS reply. Alternatively, an *MMS interworking function* can provide similar functionality, however with the added possibility of replying with a map image instead of a text-only reply. Further possible options for message-based interaction include e-mail or instant messaging, two communication formats which are increasingly supported natively by state of the art mobile phones, and which may be preferable to some users due to cost reasons, as described above.

5.2.5 POSITIONING SUPPORT COMPONENT

As explained in Section 4.2.4, the recent availability of low-cost GPS receivers reduces the dependence on costly network operator-provided positioning infrastructure. It enables developers to build and deploy location based applications at virtually no entry costs, using standard Web technologies and tools. This new GPS-based design scheme aligns well with the practices and spirit of the *Neogeography* and mashup scene, which is primarily driven by enthusiast communities or individuals with no upfront commercial interest. Consequently, the framework proposed in this thesis is, above all, designed with a GPS-enabled target audience in mind.

However, as also explained in Section 4.2.4, the number of GPS-enabled devices is still limited today. This seriously restricts the potential target audience that developers can reach with an application that depends on GPS. The *positioning support component* addresses this problem by providing interoperability mechanisms for non-GPS-enabled devices: it offers services to the framework that can attempt to infer the location of the device from various alternative data it may include with a request in lieu of a WGS 84 coordinate. For example, the positioning support component may offer an address geo-coding service; i.e. the user can manually provide a street address, or the name of nearby landmark to indicate a location, which the service will attempt to translate into a geographical coordinate. Since this process remains transparent to the application, the *positioning support component* fulfills a key role with regard to the second approach stated in Section 4.3.2 ("abstraction"), by de-coupling the positioning technology from the developer API.

The *positioning support component* is designed in a modular fashion, so that additional support services can be added when available: Examples for possible services depicted in Figure 11 are a WLAN geo-coding service that attempts to infer the client device's location from SSID(s) of nearby wireless LAN hotspots the client browser might provide with the request, or a GSM geo-coding service that does the same for cell IDs of nearby GSM cell towers. As discussed in Section 4.2.4, geo-coding and geo-location lookup tables for this kind of data are increasingly becoming available over the World Wide Web. A concrete implementation of the framework can therefore potentially make use of existing 3rd party services to realize these support modules. In principle, further possible modules are conceivable that interface with a network operator's commercial LBS infrastructure, so that the location of legacy devices subscribed in the respective network can be determined automatically (including in cases where legacy devices access the framework through SMS).

5.2.6 SPATIALLY AWARE BROWSERS AND APPLICATIONS

As discussed previously, the proposed framework and the Local Visibility Model are designed with a particular class of mobile geospatial applications in mind: applications that rely on *spatially aware* user interfaces, based on sensor-driven, orientation aware interaction principles. In accordance with the third global requirement stated in Section 4.3, prototyping support for such types of interfaces is a key goal of the framework's architecture and functional components.

The author of this thesis expects that two classes of spatially aware client applications are likely to emerge with the growing proliferation of orientation-sensor-enhanced mobile phones:

- Custom applications that include spatially aware interface paradigms into the interaction design of the application, but which operate entirely on closed, application-specific content; e.g. orientation aware games, orientation enhanced social and community applications, or sophisticated (e.g. pointing-enabled) geotagging and collaborative mapping applications (compare Section 3.2).
- Generic spatially aware browsers used to search for, and access arbitrary georeferenced content openly available on the Web in various encoding formats. Besides basic querying and visualization functionality, these browsers may also offer additional navigation features (i.e. route instructions from the users current location to a particular location in the query result), as well as different display modes which, depending on the task, may be better suited to provide a wide area overview (e.g. a map) or improved situational awareness with regard to the real-world location of geo-referenced content in the immediate physical vicinity (e.g. a handheld augmented reality interface; compare Section 6.1.2).

As depicted in Figure 11, the author proposes the LVis as a suitable generic data model for supporting both classes of client applications, and the various types of user interfaces they may implement. The following Section therefore illustrates the concepts behind the LVis, and how they relate to the previously defined requirements of user interface abstraction and prototyping support in detail.

6 The Local Visibility Model

This section introduces the Local Visibility Model – or LVis, in short – the XML format used by the framework's query engine to encode the results of an ego-centric spatial query. The LVis represents a key element of the framework, and a primary conceptual model for its interaction design philosophy: it acts both as a user interface abstraction mechanism, and an enabler technology for sensor-driven, orientation aware browsing. It allows developers to embrace the innovative features offered by novel mobile devices as they become available on the market; however without limiting them when developing for more conventional devices that do not feature advanced sensor features. Content authors who are not interested in developing their own, custom mobile user interfaces will not have to deal with the LVis at all, since the transformation to visual output is either handled by the framework's *presentation and formatting component*, or by a generic spatially aware browser installed locally on the mobile device. Mobile application developers and interaction designers, on the other hand, who want to create a customized browser or an otherwise location- or orientation aware mobile application, will need to be aware of the LVis syntax and its spatial modeling metaphors.

This section goes into the details of how the LVis expresses geographical features in the user's vicinity, such as nearby buildings and locations of interest. It is explained how the LVis serves as a common base model for a variety of presentation formats – from text-based user interfaces on legacy devices, to rich, animated sensor-driven GUIs that will be possible with future mobile phones. Finally, limitations that the LVis currently faces are discussed, and possible directions for future extensions are pointed out.

6.1 Spatial Modelling Metaphors

A Local Visibility Model is an XML document produced by the framework's query component in response to a query request. It represents a simplified abstraction of the environment around the query location, including building geometry as well as point of interest data, i.e. the locations of geo-referenced content contained in the query result. The LVis is novel insofar as it includes explicit information about the *visibility* of geographic features from the query location, thus providing inherent support for visibility-driven browsing metaphors such as the Geo-Wand. Furthermore, since the LVis is XML-based and relies on standard units and measurements (i.e. meters and decimal degrees), meaningful textual output can be derived from it by simply styling the XML accordingly, for example using XSLT. Due to the fact that the LVis includes a polar coordinate notation, orientation-based user interfaces that would normally require transformations from a Cartesian reference system to polar coordinate space (such as the Smart Compass) can be realized with considerably reduced development effort.

In order to keep the barrier of entry low for mobile application developers, the LVis uses two simple spatial modeling metaphors to express the physical arrangement of the query result space: *points of interest* (POI) and *billboards*. This section explains the details of these spatial modeling metaphors. It describes the syntax used to express them in XML, and illustrates how the LVis can be used as a common foundation for realizing various different user interface types and interaction styles.

6.1.1 CONTENT MODEL: POINTS OF INTEREST

The result of a traditional geospatial query, as it is produced by spatial databases or geospatial search engines on the Web, is a set of references to geo-coded content. The references are typically point-shaped, i.e. they can be represented by a single geographical coordinate. Therefore, the data points contained in the search result can be visualized with graphical markers on a map, along with a list of associated hyperlinks. This common way of representing geospatial query results on the Web is also referred to as the map-and-hyperlink architecture (Tezuka *et al.* 2006).

```
<poi id="poi026" name="Snack Bar" href="http://www.asnackbar.com/"
     descr="Snack bar near subway station"
     adr="Isidro-Fabela-Promenade"
     lng="16.4142" lat="48.23221" alt="178"
     d="117" hdg="95" elev="5" />
```

Figure 12. LVis `poi` (“Point of Interest”) XML element

Point-shaped geo-referenced content of the described type is represented in the LVis as a so-called *point of interest* (POI). Since the LVis includes a polar coordinate notation, each POI is described by its distance, heading and elevation relative to the query origin. This way, a textual description of the POI (e.g. “There is a Snack bar located 200 meters to the North”) can be produced without the need for complex computations. The code sample in Figure 12 shows the XML syntax for the **`poi`** element: The **`id`** attribute is a unique identifier for the element. The **`name`** attribute provides a short name the interface designer can use to create descriptive elements, e.g. a graphical label. The **`descr`** attribute provides a longer textual description of the POI, which can, for example, be used as help text or tool tip. The (optional) **`href`** attribute contains a URL associated with the POI. The (optional) **`adr`** attribute provides a street address for the POI in textual form. The **`lng`**, **`lat`** and **`alt`** attributes denote the longitude, latitude and altitude of the POI, i.e. its absolute geographical location. The attributes **`d`**, **`hdg`** and **`elev`** describe the relative location of the POI with regard to the user: **`d`** denotes the distance from the user, expressed in meters. **`hdg`** is the compass heading of the POI, i.e. the direction the user must turn to face the POI. The heading is always expressed as an integer value in decimal degrees and lies in the range between 0 and 359. The **`elev`** attribute represents the elevation (or pitch) angle of the POI, i.e. the altitude at which the user must point to aim directly at the POI. In the example from Figure 12, the user would therefore need to face towards the East direction (at a heading of 95 degrees), and point almost horizontally (at a 5 degrees upwards angle) to aim directly at the POI.

As mentioned above, the LVis carries intrinsic information about the visibility of geographical content: therefore, only POI that are directly visible from the query origin are included in the LVis per default. If the query engine is configured to also include hidden POI in the LVis, these POI will be expressed by a different XML element – the **`hpoi`** (‘h’ for hidden) element – which has the same attributes as the **`poi`** element.

6.1.2 GEOMETRY MODEL: *BILLBOARDS*

Unlike a traditional map-and-hyperlink Web search result, which consists only of points of interest, the LVis also includes the actual geometry of the environment in the query result. In other words, the query does not only return the content in the search space, but also a simplified three-dimensional geometrical representation of the 3D search space itself along with the content. The motivation for this approach is twofold:

- Locally stored geometry is a necessary pre-requisite for real-time user interfaces: If the client has information about the three-dimensional structure of the vicinity, it can react to changes from integrated sensors without the need to re-query the server. For example, the client might be able to re-compute distances, headings or the visibility of points of interest locally, as the user moves through the environment. (This distribution of distance- and visibility-computation between the mobile client and the application server can be compared with the distributed approach that was introduced by AJAX-driven Web mapping toolkits, where the shift of some user interface functionality, e.g. local management of map tile movement and layering through JavaScript, has enabled more responsive, real-time user interfaces in the browser.)

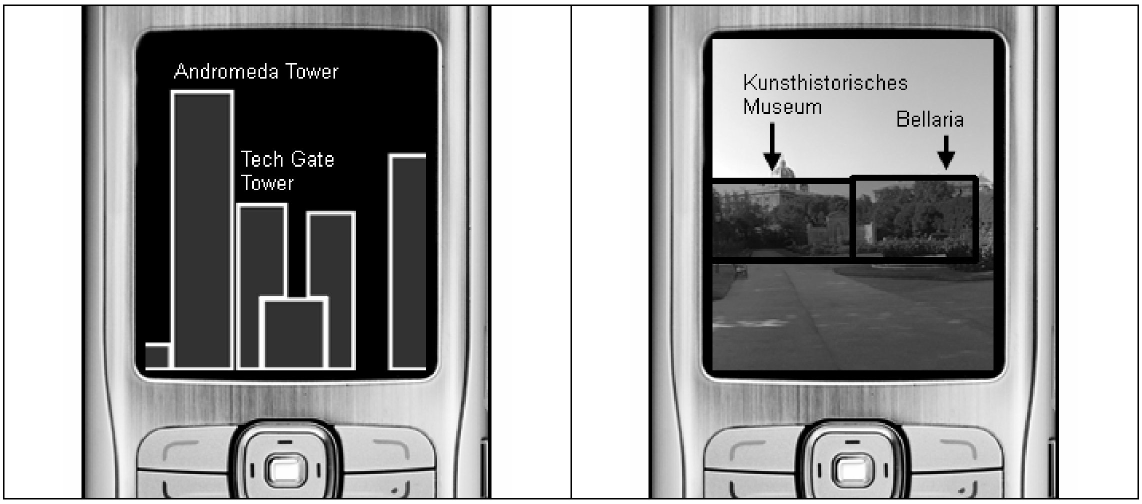


Figure 13. Using environment geometry in the user interface (concept designs): skyline and simple AR concepts

- Local knowledge of the environment geometry can be exploited on the mobile device as an interface design resource, in particular for user interfaces based on sensor-driven interaction metaphors: Figure 13 shows two concept designs for user interfaces that could be implemented with information about the geometry of the surrounding environment. The example in Figure 13, left, shows an interface design that presents a schematized skyline view of the user's vicinity. Figure 13, right, shows a more advanced realization of the same concept: using sophisticated positioning techniques such as differential GPS and sufficiently accurate heading and tilt sensors, a basic *augmented reality* (AR) interface might be envisioned for future devices, where labels are superimposed on the phone camera image to describe buildings and POI nearby.

Traditional geographic data formats like GML or KML describe geometry using polygons or geometric primitives in Cartesian 2D or 3D space: They are therefore not human-readable as such and require complex client-side pre-processing to produce visual output. Since this conflicts with the design goal of keeping the LVis simple, presentation-agnostic and, in particular, suitable for textual output, a more basic, abstracted metaphor is introduced: The LVis models the environment using a *billboard* metaphor. Billboards approximate a geographic feature, such as a building, by a flat, rectangular wall that faces directly towards the user. As in the case of POI, billboards are only included in the LVis for (parts of) buildings that are visible from the query origin.

Essentially, the LVis can therefore be thought of as a 360-degrees panoramic "cardboard cutout" version of the immediate vicinity, much like a movie set where the environment is not made up of solid buildings, but instead only of building facades. Since the LVis notation includes polar coordinates for each billboard, a textual description of a nearby building (e.g. “There is an Office Building 150 meters to the South”) can be directly derived from the XML code without the need for arithmetic computations.

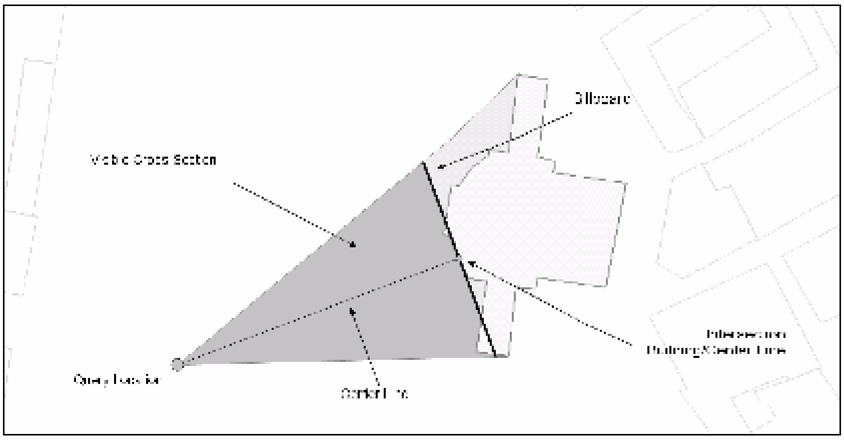


Figure 14. LVis billboard

Figure 14 illustrates how a billboard is generated from the geometrical shape of the building it represents: The image shows a top-down view on a building (“building footprint”). The visible cross section the building occupies in the user’s field of view is indicated by the shaded cone that extends from the query location to the outer edges of the building footprint. The billboard is then defined as a rectangle which:

- stands upright, i.e. normal to the ground;
- is exactly as wide as the visible cross section of the building;
- is located at the intersection point between the center line of the visible cross section and the building footprint;
- faces directly towards the user, i.e. is placed at a 90-degree angle to the cross section’s center line;
- is as high as the building it represents.

Due to its simple geometrical structure, a billboard can be unambiguously described through very few parameters: the location of its geometrical center (“anchor point”), its horizontal and vertical width, and descriptive metadata associated with it. Figure 15 shows an XML code sample for the LVis **`billboard`** element: The element has all the attributes of the **`poi`** element (**`id`**, **`name`**, **`descr`**, **`href`**, **`adr`**, **`lng`**, **`lat`**, **`alt`**, **`d`**, **`hdg`** and **`elev`**).

```
<billboard id="bldg019" name="Tech Gate"
           href="http://www.techgate.at/"
           descr="Tech Gate Office Building"
           adr="Carl-Auboeck-Promenade"
           lng="16.4094" lat="48.232" alt="183"
           d="42" hdg="125" elev="28"
           hwidth="41" vwidth="18" />
```

Figure 15. LVis `billboard` XML element

Since the billboard also has a spatial extension, there are two additional attributes: the **`hwidth`** and **`vwidth`** attributes: While **`d`**, **`hdg`**, and **`elev`** denote the distance, heading and elevation of the billboard's anchor point, the **`hwidth`** and **`vwidth`** attributes provide the horizontal and vertical width of the billboard, respectively. Like **`hdg`** and **`elev`**, both attributes are expressed in decimal degrees, with a range of values between 0 and 180 degrees. In order to better illustrate the relation between the attribute values of the **`billboard`** element and the corresponding geometrical arrangement, Figure 16 shows a representative example.

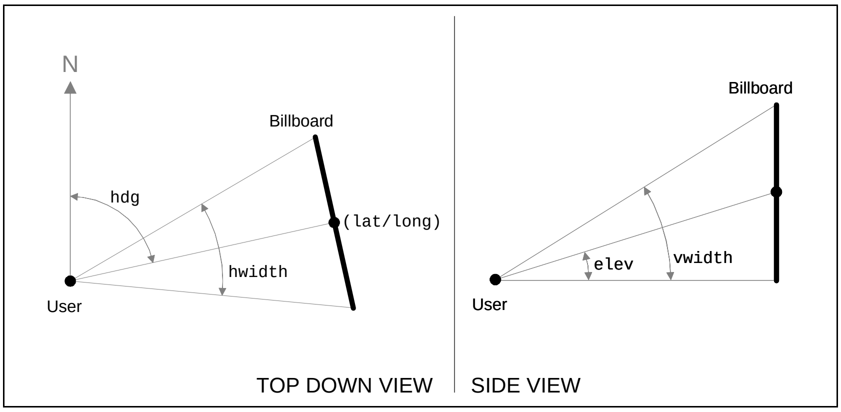


Figure 16. `billboard` XML attributes and geometrical arrangement

6.2 Limitations

Due to the strong amount of abstraction and simplification that the LVis billboard metaphor introduces, there are certain limitations in practice. Some of these limitations are discussed below, along with possible strategies to address them appropriately, if possible.

6.2.1 DISTANCE MISMATCH

Figure 17 illustrates a limitation of the LVis that is due to the way the location of a billboard's anchor point is defined: As explained earlier, the anchor point is placed at the intersection point between an imaginary line in the center of the angular interval covered by the building in the users' FOV ("cross section center line"), and the building footprint (compare Figure 14).

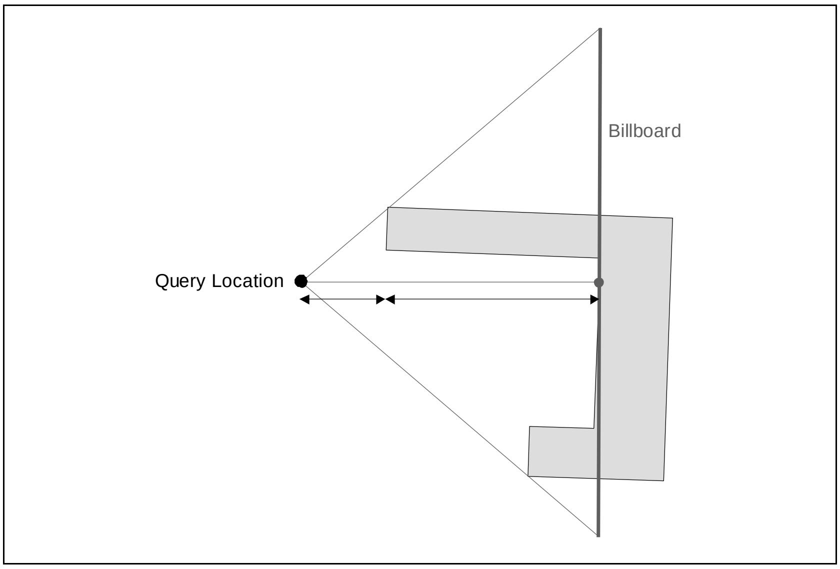


Figure 17. Distance mismatch example

In cases of irregularly shaped buildings, this definition can lead to the placement of billboards at a distance that is likely to be considered 'too far away' by the user. In practice, however, this source of error has been found to be of minor relevance by the author, in particular in comparison with the physical distance errors introduced due to GPS inaccuracy. (Refer to Section 8 for a detailed evaluation of the effects of GPS error on billboard placement.)

6.2.2 ARCHED BUILDINGS

As explained, a billboard is defined as a rectangular wall, described by the location of an anchor point and a horizontal and vertical extension. By definition, it is therefore not possible to correctly describe arched buildings: For example, the building depicted in Figure 18 will be modeled as a solid billboard in the LVis. Hence, all POI behind that billboard will be considered hidden, even though they might actually be visible to the user through the passageway in reality. Therefore, this limitation may be a relevant source of error for user interfaces based on the Geo-Wand principle.

Figure 18. Arched building

6.2.3 BUILDINGS COVERING MORE THAN 180 DEGREES OF THE FOV

Per definition, a billboard cannot model a building that surrounds the user, i.e. which covers more than 180 degrees of the user's field of view. In the query engine implementation presented in Section 7, this inherent limitation of the billboard metaphor has been addressed with the following strategy: Every building that covers more than 180 degrees of the user's view is always modeled using three equal-width billboards, instead of one. This way, all buildings that cover between 180 and 360 degrees or the user's field of view can be expressed, as shown in Figure 19.

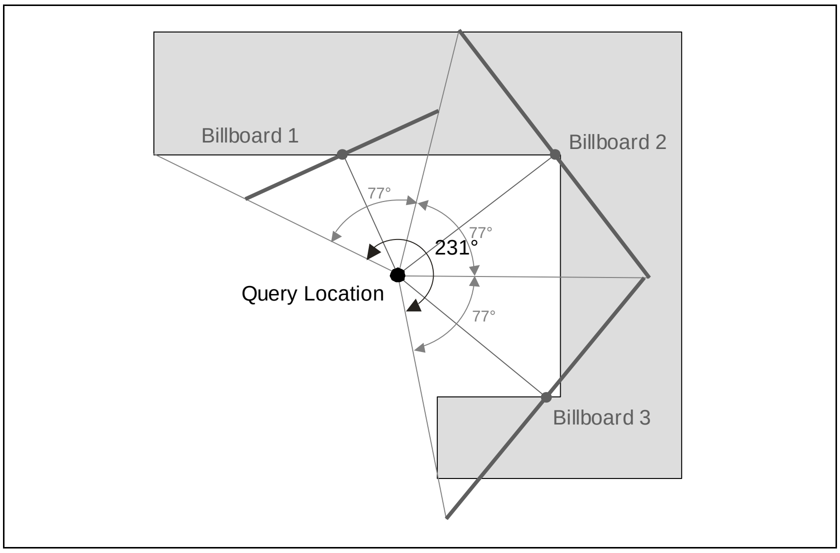


Figure 19. Building covering more than 180 degrees of the FOV

6.2.4 PERSPECTIVE DISTORTION

Perhaps the most relevant limitation of the billboard modeling principle with regard to its real world applications is perspective distortion: the LVis generally omits shape and perspective in favor of simplicity and compact syntax. Hence, billboards rarely resemble the buildings they represent in practice, except for their location and their approximate dimensions. Due to the fact that billboards are always aligned with the user's viewing plane (i.e. they face directly towards the user), the part of the building that is further away from the user will always be depicted too high. Likewise, the part of the building that is closer to the user than the anchor point will always be depicted too low.

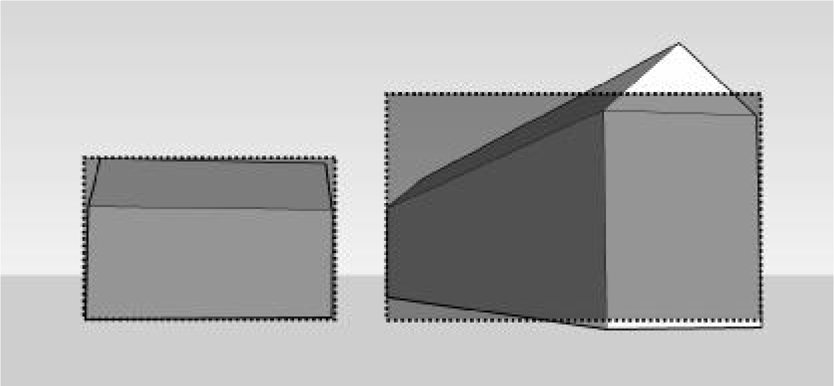


Figure 20. Perspective distortion

Figure 20 shows this distortion effect, which is less perceivable for buildings viewed frontally (Figure 20, left building), or for buildings that are far away; but which gets more severe for nearby buildings viewed from an acute angle (Figure 20, right building).

6.3 Possible Extensions

In its current state, the LVis represents a basic conceptual model for encoding information about the geometry of the surrounding environment in a simple, compact form. In order to determine whether the modelling concepts used in the LVis are truly effective in supporting the design and implementation of more usable mobile geospatial interfaces; and whether additional syntax elements besides points of interest and billboards are desirable, experimentation and prototyping will be crucial, as discussed in Section 4.3.3.

While the creation and evaluation of a larger set of prototype user interfaces is beyond the scope of this thesis, two potential directions for extensions to the LVis have already been identified by the author, based on early feedback to his work: the inclusion of a *level of detail* concept to mitigate shape and perspective distortion effects, and the possible addition of navigation elements to the LVis syntax.

6.3.1 LEVELS OF DETAIL

A fundamental design goal of the LVis was to keep modeling concepts and syntax as simple as possible. As a particular consequence, it therefore does not include true 3D geometry data, as explained in Section 6.1.2. First feedback to the author's work has already shown, however, that detail geometry is considered a crucial feature. In fact, it may be a key enabler for particularly compelling forms of geospatial interfaces in the future, such as augmented reality interfaces. Furthermore, as can be concluded from the LVis' limitations discussed above, the overly minimalist billboard model may generally be problematic for nearby buildings, where it frequently leads to various distortion effects (e.g. distance mismatch, perspective distortion, or the problems involved with the depiction of buildings that cover more than 180 degrees of the user's field of view).

Obviously, it is possible to solve these distortion problems by extending the definition of the billboard, e.g. so that it does not necessarily need to face towards the user, but may be oriented at an arbitrary angle; or by allowing billboards of general (and not necessarily rectangular) shape. This, however, will require additional descriptive parameters in the LVis syntax, thus making the XML code less compact. Worse, it will add complexity to the design of the conceptual model and thus jeopardize the inherent human-readability of the LVis. A hybrid *level of detail* approach, where 3D geometry is included for nearby geographic features, while far-away features are modeled as billboards, appears to be a suitable compromise between simplicity and detail which may prove useful in the future.

6.3.2 NAVIGATION PRIMITIVES

As further feedback to the author's work has shown, way-finding information and walking directions are considered a particularly desirable feature in many types of mobile geospatial applications. In order to enable mobile application developers to easily include this functionality seamlessly into their applications, a promising future extension to the LVis is the addition of a new **`<route>`** syntax element that describes a directed path towards a particular point of interest or billboard. A necessary pre-requisite for this feature is the availability of road network data in the map data source used by the query engine (compare Figure 11).

7 Implementation

This section describes the technical details of the framework implementation. The implementation is thereby meant to serve as a proof of concept: Focus is put on instructiveness and on illustrating the concepts and functionality that are introduced by the novel architecture components proposed in this thesis. Performance and scalability of the algorithms behind these components has been of lesser concern in the implementation.

The implementation was produced in the Java programming language, using a setup that follows well-established practices for Java-based Web applications: the open source *Apache Tomcat*[110] was chosen as Web server and Servlet container to host the framework logic; the open source relational database system *PostgreSQL*[111] (with the *PostGIS*[112] spatial database extension) was used to store the environment block model required by the query engine. Other open source libraries and external online services the implementation relies upon (such as e.g. the open source *JH Labs Java Map Projection Library*[113], which handles coordinate transformations; or the *Yahoo Map Image API*[114] that is used to obtain map base images for the presentation and formatting component's map module) are mentioned and described in the respective sub-sections.

Please note that part of the implementation has been produced by the author as part of the Telecommunications Research Center Vienna's *Point to Discover*[115] research project.

7.1 Content Aggregation Component

Presently, *points of interest* are the only type of content the Local Visibility Model allows. The task of the *content aggregation component* implementation is therefore to extract point of interest data from sources of geo-referenced content encoded in various data formats, and provide it to the query engine. As the implementation serves as a proof of concept, it does not support all formats defined in the *GeoStack* (compare Sec-

---

[110] http://tomcat.apache.org/

[111] http://www.postgresql.org/

[112] http://postgis.refractions.net/

[113] http://www.jhlabs.com/java/maps/proj/

[114] http://developer.yahoo.com/maps/rest/V1/mapImage.html

[115] http://p2d.ftw.at

tion 2), but only those that were actually used for testing of the framework, such as KML and several proprietary formats; e.g. the query result format of the online geographical database and Wikipedia search engine *geonames.org*[116] (Simon and Fröhlich 2007), and a format used in-house by a project partner within the *Point to Discover* project consortium.

Since all data used for testing is based on XML, integration is straightforward: The implementation consists of an XML parser (using the standard *Java API for XML Processing* JAXP), which – depending on the complexity of the XML schema of the data format – extracts relevant data either by sequentially traversing the XML tree (e.g. in case of the *geonames.org* XML format); or through XPath statements (using the XPath implementation of the open source *Apache Xalan* Java XSLT processor[117], e.g. in the case of KML). The *content aggregation component* is thereby implemented so that it can either operate on local files or directly on remote data over HTTP, thus enabling location based mobile access to existing 3rd party content on the World Wide Web.

7.2 Visibility Query Component

As explained in Section 6, the LVis that is produced in response to a query carries information about the *content* (in the form of points of interest) as well as the *geometry* (in the form of billboards) that is visible from the user's location. Consequently, a geometrical model of the environment is required in order for the query engine to be able to compute the LVis. In case of the implementation produced in this thesis, the query component relies on a 2.5-dimensional model of the environment: i.e. each building in the model is represented by a two-dimensional footprint polygon, which is extruded by a height value ("block model"). Until recently, the cost of producing such models by conventional means such as surveying has been prohibitive. However, current technologies such as automatic or semi-automatic building reconstruction from high-resolution aerial imagery (Nevatia and Price 2002) and/or LIDAR (LIght Detection And Ranging) scans (Hu *et al.* 2004) have greatly reduced the cost of gathering of large-scale three-dimensional environment data, making its use for a variety of purposes – from urban development to mobile phone radio network planning – more widespread.

The query component implementation uses two different block model datasets: First, a small sample dataset that was produced manually by the author for testing purposes.

---

[116] http://www.geonames.org/export/wikipedia-webservice.html

[117] http://xml.apache.org/xalan-j/

The dataset covers an area of roughly 600 x 600 meter around the author's office premises, an architecturally rather unusual business district in the North of Vienna, Austria. A basic, custom-developed JavaScript drawing tool was implemented to approximate building footprints on a Google Maps' satellite images and extract the footprint polygons' vertex coordinates in WGS 84 coordinate space. The height of the buildings was known in some cases (e.g. through data made publicly available on the World Wide Web by the municipality[118]), and estimated for the remaining buildings using photographs. A second, professionally surveyed block model was provided by a project partner of the *Point to Discover* project: the properties of this block model conform to those of the manually-produced dataset, i.e. buildings are modelled as polygon footprints, with a single height parameter per footprint. The model covers an area of roughly 2.4 x 2.4 km in the center of Vienna's inner city district. Both block models are stored together in a *PostgreSQL*/*PostGIS* spatial database.

The query process itself is conducted in three stages: In the first stage, the query component retrieves the block model data for the region around the query location from the database, using a standard bounding box query. This coarse pre-selection is necessary in order to limit the operating data set for the subsequent visibility algorithms, which are otherwise too computationally expensive to operate on the entire block model. In the second stage, the query component computes the visibilities for the POI in the operating data set: This is done using a basic line of sight algorithm, as explained in sub-section 7.2.1. In the third stage, the query component computes the billboards for the visible buildings, as explained in Section 7.2.2. In order to graphically explain the query process, section 7.2.3 illustrates a concrete query example with visualizations for each processing stage involved.

The algorithms that implement the core functionalities of the query component are based on analytical geometry: A dedicated geometry library that handles basic geometry calculations such as line intersections in 3D space was implemented by the author for this purpose, using the Java 2D API[119] as a foundation. As explained, the algorithms, as well as the 3-stage query process, aim to illustrate the principle of operation, rather than to achieve the highest possible performance: A demonstration of how the performance of visibility computation can be increased by several orders of magnitude by using a tree-indexed version of the environment block model and a sampling-based query ap-

---

[118] http://www.wien.gv.at/stadtentwicklung/donaucity/

[119] http://java.sun.com/products/java-media/2D/

proach (essentially amounting to the implementation of a spatial database management system with native visibility query support) can be found in (Maierhofer *et al.* 2007).

7.2.1 POI LINE OF SIGHT ALGORITHM

In order to compute which POI are visible from the query location, the implementation uses a basic line of sight algorithm: for each POI in the operating dataset, the line between the query location and the POI location ("line of sight") is checked for intersections with the buildings in the environment model. A POI is considered visible if:

- there is no intersection between the line of sight and any of the buildings in the environment model, or if
- the POI is located inside a building, and this building is the only one that intersects the line of sight.

Figure 21 illustrates this rule by an example: only those three POI are considered hidden whose line of sight is blocked by another building. As explained above, intersections between lines of sight and the 2.5D building blocks are computed based on the functionality provided by the Java 2D API.

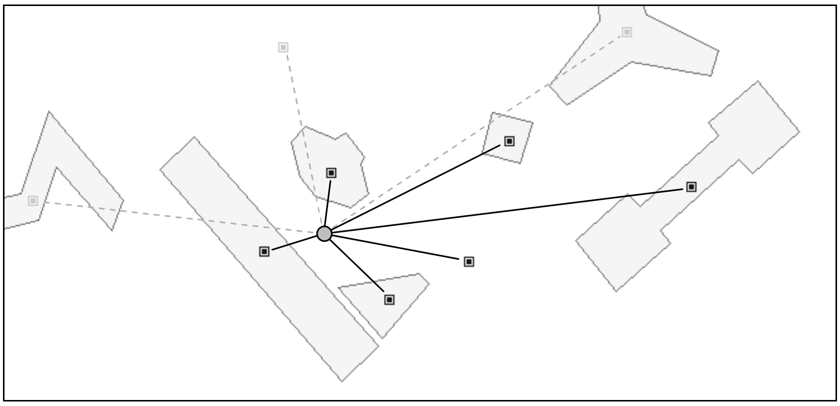


Figure 21. POI visibility: line of sight calculation

7.2.2 BILLBOARD COMPUTATION ALGORITHM

Visibility problems for 3D geometry are a classical topic of computational geometry. The authors of (Sutherland *et al.* 1974) have classified different methods and algorithms for hidden surface (or line) determination: Generally, algorithms are either image-accurate or object-accurate. The fundamental difference is that image-accurate algorithms take advantage of the raster-characteristics of digital displays. They compute the visibility using a sampling approach. Object-accurate algorithms on the other hand, compute the visibility of geometric primitives in object space analytically, which is, in general, substantially more processing intensive.

In order to provide a more instructive insight into the process of how the billboards are computed for buildings visible in the user's field of view, the query component implementation follows an object-accurate approach nevertheless, which is the reason for the separate pre-selection process performed in the first query stage. In order to further reduce the computational complexity, so that near real time response times are achieved, the billboard computation algorithm does not operate on the actual 2.5D building blocks. Instead, it replaces each building in the operating data set by a "cross section rectangle" in a first step. The cross section rectangle is a rectangle that is standing upright (i.e. normal to the ground), and which is as high and as wide (as seen from the user's perspective) as the building it replaces. An example of this step is shown in Figure 22: The black dot in Figure 22 represents the query location, relative to which the cross section rectangles are computed.

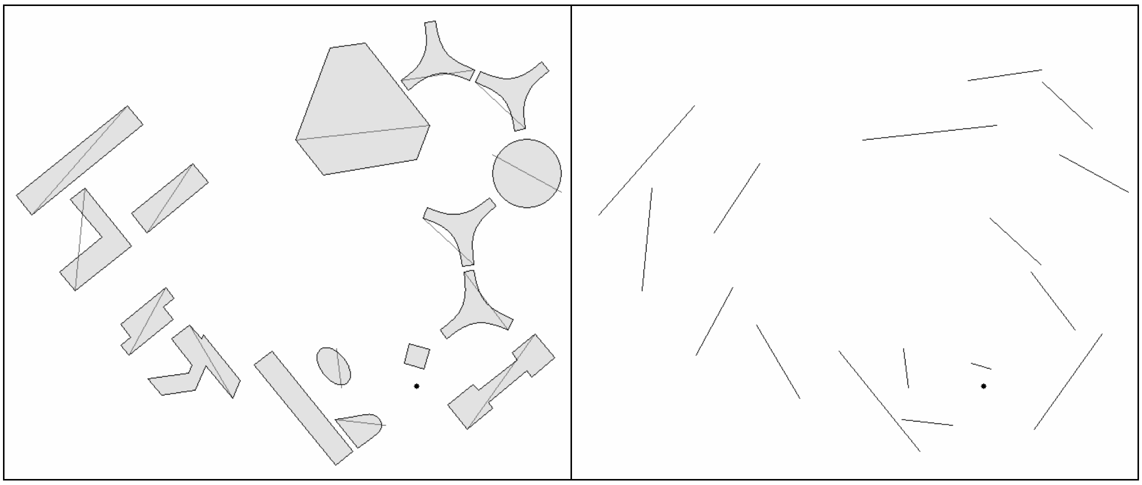


Figure 22. Operating data set: full block models vs. cross section rectangles (top-down view)

Each cross section rectangle is then checked for full or partial occlusions against every other cross section rectangle ("occluder"). Thanks to the simplification from 2.5D blocks to cross section rectangles, the occlusion detection process is simplified to a case differentiation: In order to compute occlusions among two cross sections, it is sufficient to check for occlusions only of their lower and their upper edges, respectively. The following four occlusion cases are possible:

- **Full occlusion case:** The upper edge of the cross section under test is fully obscured by the occluder. Therefore, the entire cross section must be fully obscured, and can be ignored in the occlusion detection process from now on (compare Figure 23a), as well as in the subsequent billboard generation process.
- **No occlusion case:** The currently tested upper edge is not obscured by the occluder at all. In this case it is tested whether the lower edge is also not occluded. If this is the case, the currently tested wall is not obscured by the occluder, e.g. because the occluder is further away than the wall, or does otherwise not interfere with the tested cross section.
- **Partial occlusion (trim or split) case:** The upper edge is partially obscured by the occluder. In this case, the visible segment(s) of the upper edge are determined. The accordingly trimmed cross section is used for further occlusion detection from then on (compare Figure 23b). In cases where the occluded cross section extends beyond both sides of the occluder (as shown in Figure 23c), the occluded cross section is left unchanged for the remaining occlusion detection process; it is, however, split into two cross sections for the subsequent billboard generation process.
- **Bottom occlusion case:** Only the lower edge of the tested wall is obscured by the occluder – either fully (Figure 23d) or partially (Figure 23e) – but not the upper edge. Therefore, the building is still visible, but it is partially occluded by a lower building in front of it. In this case, the cross section is left unchanged for the remaining occlusion detection process and the subsequent billboard generation process.

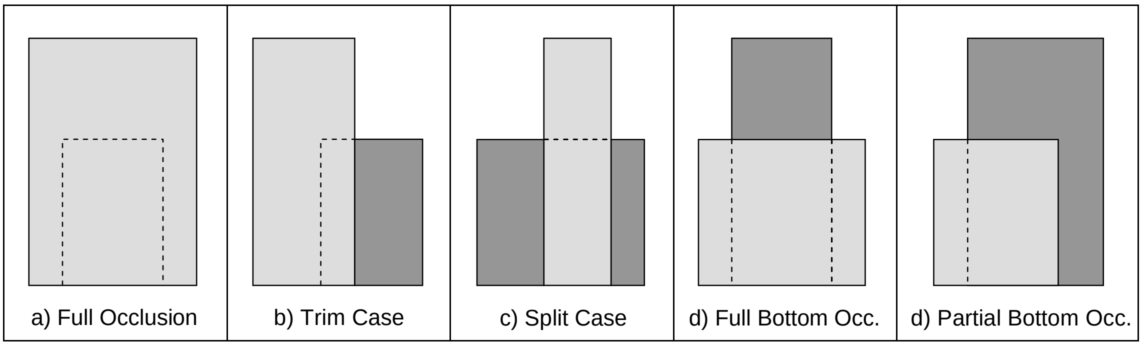


Figure 23. Occlusion cases (dark rectangle: cross section under test
light rectangle: occluder)

Once all cross sections have been tested for occlusions, and only the appropriately trimmed or split cross sections remain, the actual billboards are computed for each cross section: As explained in Figure 14 in Section 6.1.2, the billboard is computed so that it is exactly as wide as the angle the visible cross section occupies in the user's field of view; is located at the intersection point between the center line of the visible cross section and the building footprint; faces directly towards the user; and is as high as the building (i.e. the cross section rectangle). As an example, Figure 24 shows the cross sections of an operating data set before (Figure 24, left) and after (Figure 24, center) the occlusion detection process; as well as the billboards that result from the visible cross sections (Figure 24, right).

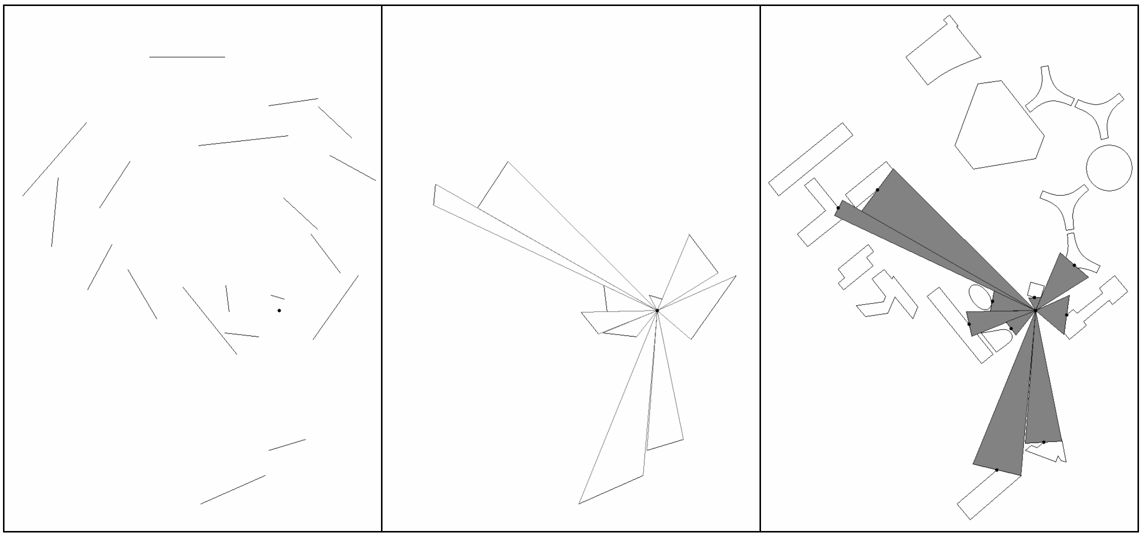


Figure 24. Cross sections before (left) and after (center) occlusion detection; billboards generated from remaining cross sections (right)

7.2.3 QUERY EXAMPLE

For a comprehensive understanding of the complete visibility query process and the structure of the resulting LVis, a concrete query example is presented below, step by step. The individual steps are visualized in two ways: For a detailed view, screenshots are shown of each step as it is visualized in the framework's proprietary GUI component that was implemented by the author. Alternatively, the same scene is shown as 3D geometry, visualized in *Google Earth*. This viewing mode allows a better assessment of the three dimensional structure of the presented data, and how it relates to the physical world.

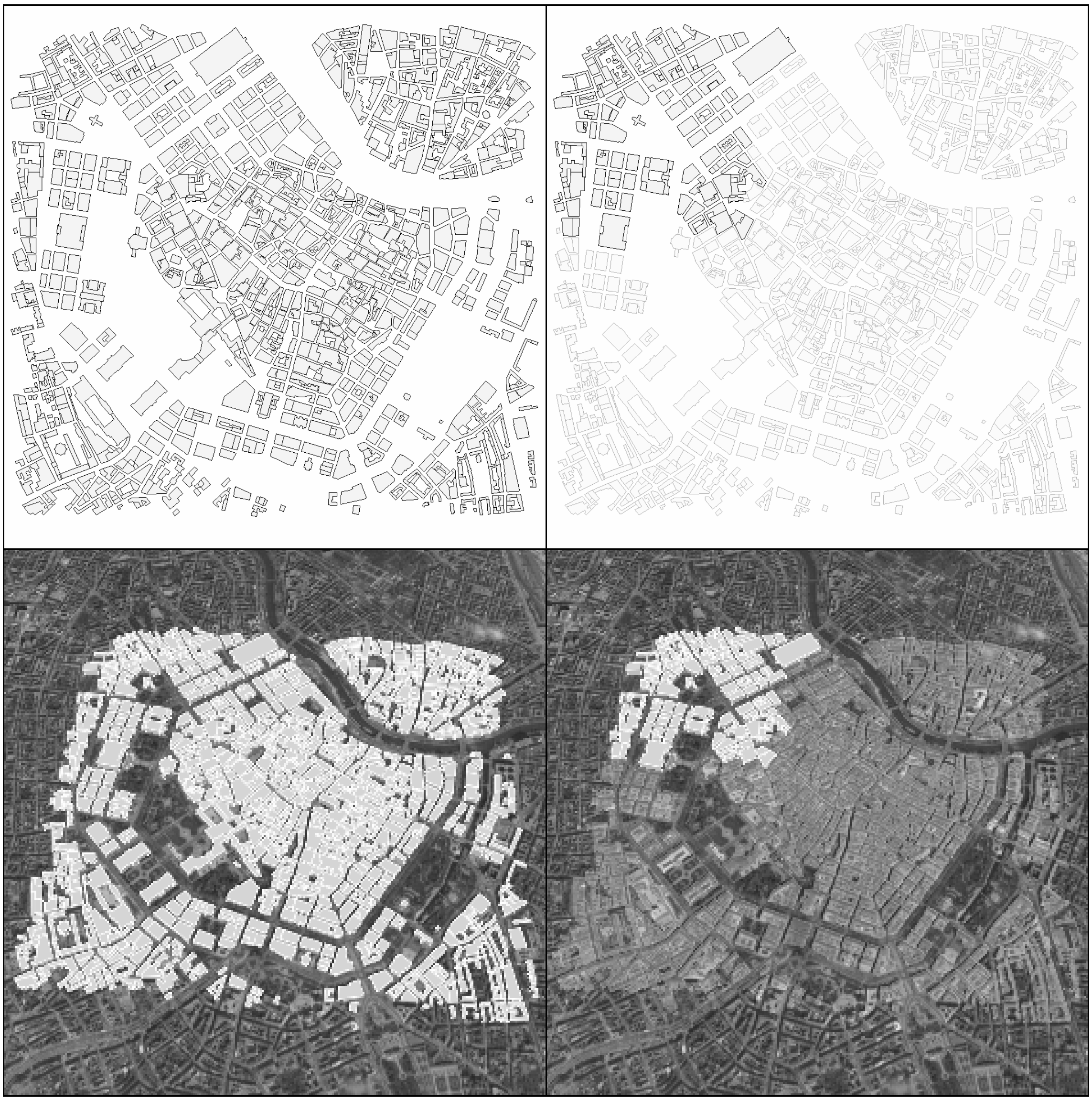


Figure 25. Step 1: pre-selection of the operating dataset (right) from the full dataset (left)

Step 1: Pre-selection. As explained, the first step of the query process is to pre-select an operating dataset from the database, using a bounding box query. Figure 25 shows part of the environment block model that was used with the implementation, stored in the *PostGIS* database: The total model depicted in Figure 25 consists of 1013 2.5D building blocks. After pre-selecting a region in the North-Western corner of the dataset using an 800 x 800 meter bounding box, the operating dataset is limited to 143 buildings, as shown in Figure 25, right.

Figure 26. Step 2: POI line of sight computation

Step 2: POI visibility. In the second step, the POI in the region of the operating data set are checked for clear line of sight, as explained in Section 7.2.1. Figure 26 shows the corresponding result, which was generated out of a POI data set of bars and restaurants in Vienna's first district (1002 POI in total), provided by a project partner of the *Point to Discover* research project consortium[120].

Figure 27. Step 3: Billboard computation

Step 3: Billboards. In the third step, the query engine computes the billboards for the visible buildings in the operating data set, as shown in Figure 27.

---

[120] Newspaper "Falter", see http://p2d.ftw.at/applications.html

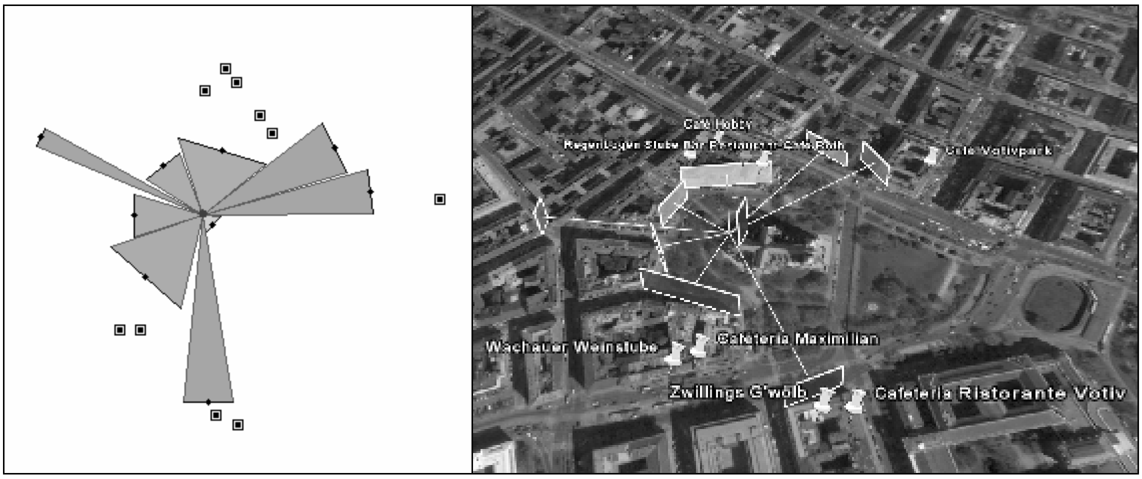


Figure 28. Completed LVis

The completed LVis, consisting of the visible POI and the billboards for visible buildings is shown in Figure 28. As explained, the LVis can optionally include hidden POI as well; the default behavior of the query component, however, is to omit hidden POI in order to keep the byte-size of the LVis small. (The XML code that corresponds to the example above is just below 4 kilobytes in uncompressed form and can be found in Appendix A for reference).

7.3 Presentation and Formatting Component

The *presentation and formatting component* implementation demonstrates the key concepts discussed in Section 5.2.3. It supports the translation of the LVis into a textual description, and the translation into a graphical map image. The component can thereby adapt the output to some extent, according to the properties of the client device.

7.3.1 WML/XHTML TRANSFORMATION MODULE

The WML/XHTML transformation sub-module of the *presentation and formatting component* implements the translation of the LVis into a mobile-friendly WAP- or Web page. The translation is performed by populating a generic WML or XHTML template with the textual information contained in the LVis, and adding a reference to an image that shows the POI contained in the LVis on a map graphic (only in the case of the XHTML page). The two Web-based output channels provided by the presentation and formatting component implementation are therefore:

- a basic textual presentation, compatible with low-end, WML compliant, WAP capable mobile phones
- a textual/graphical presentation, which resembles the map-and-hyperlink structure commonly found in desktop Web map applications (Tezuka *et al.* 2006); and which complies with the *XHTML Mobile Profile*, i.e. can be viewed in the XHTML browser of state of the art WAP 2.0-compliant mobile phones

7.3.2 MAP MODULE

The map sub-module of the *presentation and formatting component* dynamically generates the map graphic that is referenced in the XHTML document produced by the XHTML transformation module. In order to produce the base map, the implementation relies on the *Yahoo Map Image API*[121]: this free, HTTP-based API returns a graphical map image in the form of a PNG image file, according to parameters in the HTTP request. Though this procedure is proprietary in syntax, it resembles map image retrieval over a WMS interface.

After retrieving a map base image for the region depicted in the LVis over the API, the map module computes a local conformal projection for the map: This projection is used to translate the longitude and latitude of each POI to the respective East and North coordinate relative to the map center. (Note that the current map module implementation ignores the billboards contained in the LVis, and only considers the POI for the generation of the map graphic.) Based on the known ratio meters/pixel, which is fixed for each zoom level supported by the *Yahoo Map Image API* (and which was hand-measured by lack of an official documented map scale), the East and North values are translated to their respective X and Y coordinate on the graphic. Icons for each POI are drawn on the map graphic at the appropriate locations using the functions of the Java AWT library, and the final image is re-encoded as a PNG image file.

As a side remark, it has to be noted that there is no official documentation available on the type of projection that is actually used for the map images provided by the Yahoo Map Image API*: The framework implementation uses a Transverse Mercator projection (centered on the map) to translate POI WGS 84 coordinates to map graphic pixel coordinates. While this type of projection was chosen arbitrarily, it was found to provide very good correspondence between the map base image and the locations of the super-*

---

[121] http://developer.yahoo.com/maps/rest/V1/mapImage.html

imposed POI icons by the author. Some examples of maps produced with the framework implementation can be found in Section 8.3.2.

7.3.3 DEVICE CAPABILITIES REPOSITORY

The *device capabilities repository* is the knowledge base used by the *presentation and formatting component* to produce output that is optimized towards the specific characteristics of a particular target device. The *device capabilities repository* is essentially a large list of *device descriptions* – data structures that describe properties such as screen resolution, supported image formats or level of markup support – for a multitude of mobile phone models and browsers. While the motivation for utilizing device knowledge to dynamically adapt applications to behave in different ways for different devices is evident due to the device fragmentation problem, the efforts to develop a standardized description format are, however, still ongoing: a number of competing proprietary solutions exist, and there is no single agreed representation of device properties at the time of writing[122].

Within the scope of this thesis, the *device capabilities repository* was implemented based on the *Wireless Universal Resource File* WURFL[123], a collaborative, open source effort to continuously collect information about capabilities and features of mobile device models in the form of an XML document. Using the information provided by the WURFL, the *presentation and formatting component* can differentiate between WML- and XHTML-capable devices; furthermore it can determine the width of the device screen and size the map graphic accordingly. Further, more fine-grained adaptations (e.g. regarding font size or coloring) are possible in principle, but are, however, considered beyond the scope of this thesis.

7.4 Messaging Interworking Functions

As explained in Section 5.2.4, the task of the *messaging interworking functions* is to provide interfaces to various messaging channels, which users might choose to use for various reasons: e.g. for users with legacy devices, SMS might be the only available communication channel to access a mobile data service; travelling users might prefer to use SMS instead of browser access to avoid high data roaming charges; or users might simply prefer to access the service over a message-based channel for reasons of conven-

---

[122] http://www.w3.org/TR/dd-landscape/

[123] http://wurfl.sourceforge.net/

ience (for example, a map sent via MMS can easily be stored for later reference or forwarded to other people instantly).

7.4.1 SMS INTERWORKING FUNCTION

The *SMS interworking function* was implemented as the framework's primary mechanism for supporting legacy GSM phones. For commercial SMS services that automatically process and send large amounts of SMS, the use of HTTP-based interfaces that connect an operator's SMS center to the service provider's Web infrastructure has become customary. Access to these interfaces is either provided by the operator itself, or through 3rd party bulk SMS wholesalers, at varying pricing models.

Since the implementation produced as part of this thesis aims to serve as a proof of concept only, a free solution was preferred over paid access. As an alternative access method with a mode of operation that is comparable to commercially deployed bulk SMS solutions, the implementation therefore relies on the free, Web-based multichannel messaging service *twitter*[124]: the service features an open API to receive and send SMS messages from and to mobile phones. Incoming SMS are translated into e-mails (which carry the SMS text in the body); outgoing SMS can be sent directly using an HTTP based API. Based on the *twitter* service, the SMS interworking function was realized as a Java *Mailet*: similar to a Servlet, which handles HTTP requests instead of a Web server (compare Section 5) a *Mailet* is a software component that resides in a runtime environment of an e-mail server and handles incoming e-mail messages. As the mail server and *Mailet* host environment, the open source *Java Apache Mail Enterprise Server (JAMES)*[125] was utilized.

Since it is, in general, not possible to infer the location of the user automatically in case of SMS-based access, the framework implementation developed in this thesis requires the user to provide a location hint in the request SMS, i.e. a street or landmark name. The *Mailet* extracts the location hint and passes it to the positioning support component, which will attempt to geo-code it. The relevant source code of the *Mailet* (which also demonstrates the use of the *twitter* API to send an SMS in response to an incoming e-mail) can be found in Appendix B.

---

[124] http://twitter.com/

[125] http://james.apache.org/

7.4.2 MMS INTERWORKING FUNCTION

Since there was no free solution available for sending MMS multimedia messages at the time of writing, no dedicated *MMS interworking function was implemented*. It is noted, however, that since the structure of an MMS message resembles that of an e-mail (with multiple media elements combined to a composite message using the MIME multipart format, and support for the JPEG and PNG image formats[126]), it is possible in principle to re-use existing components of the implementation (such as the e-mail based SMS interworking solution or the *presentation and formatting component*'s map module) to process incoming MMS requests and/or compose MMS messages that carry a map graphic as reply.

7.5 Positioning Support Component

As explained in Section 5.2.5, the framework proposed in this thesis is designed so that mobile clients are primarily expected to feature their own, self-sustaining positioning technology such as GPS; and that they will autonomously deliver a location estimate with the query request. For requests that do not include a location estimate (e.g. requests issued via the *SMS interworking function*), the *positioning support component* may offer varying fallback mechanisms, through which it can attempt to infer the device's location using alternative information provided with the request. As a proof of concept, the implementation developed as part of this thesis includes an address geo-coding interface: in case a request issued to the framework contains a street address or landmark name, the positioning support component will attempt to resolve it and translate it into a geographic coordinate using Google's geocoding API[127] (a service that is available for free for non-commercial use). Examples of how the service functions in practice can be found in Section 8.3, where it is used in conjunction with case study 2 (legacy device case study).

7.6 Spatially Aware Browsers

Section 5.2.6 introduced the concept of spatially enhanced applications and browsers: Installed on future location- or orientation-aware mobile phones, these applications

---

[126] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, *Multimedia Messaging Service (MMS); Media formats and codecs*, (3GPP TS 26.140), http://www.3gpp.org/ftp/Specs/html-info/26140.htm

[127] http://www.google.com/apis/maps/documentation/services.html#Geocoding

promise to enable intuitive “real world browsing” of geo-referenced content available on the World Wide Web. Conceivable interaction metaphors for real world browsing have been discussed earlier in this thesis and include, among others, the Geo-Wand, the Smart Compass, or handheld augmented reality interfaces.

To illustrate how the LVis can serve as a common data exchange format for real world browsing interfaces, the author implemented a functional Geo-Wand, using a Java 2 Micro Edition (J2ME, JavaME) compatible Bluetooth-enabled, but otherwise mass market mobile phone, in conjunction with prototype orientation sensor hardware available to the author as part of the *Point to Discover* research project. Results from a series of functional trials conducted by the author are presented in Section 8.3. Furthermore, Section 8.2 discusses in detail the LVis’ suitability for the implementation of augmented reality interfaces. The use of the LVis to realize a Smart Compass GUI and the panorama-like skyline view presented in Section 6.1.2 was shown previously by Simon and Fröhlich (2007).

8 Validation

The previous section described a prototypical implementation of the framework proposed in Section 5. By implementing the key components, and testing the algorithms against real world data – such as commercial POI data, and a professionally surveyed 2.5D block model of the inner city of Vienna, Austria – a first step towards the functional validation of the framework has been achieved.

For a comprehensive verification of the system, the underlying interaction design principles it aims to support, and the engineering assumptions behind them, however, an implementation of the framework alone is not sufficient: For example, in Section 5 it was stated that one of the motivations for the framework's design evolved out of the recent availability of low-cost GPS receivers, which are becoming increasingly common on the market as a Bluetooth accessory or as a device feature integrated in state of the art mobile phones. Throughout the thesis it was implicitly assumed that GPS provides an accurate location estimate. It is known, however, that the accuracy of GPS can vary considerably due to various external influences. Also, the thesis claims to provide interoperability with legacy devices through fallback mechanisms such as manual address or landmark geo-coding: these may again introduce different types of deviation characteristics for the location estimate. Last but not least, it is argued in Section 6 that the Local Visibility Model is of particular value for the development of orientation aware interfaces like the Geo-Wand, the Smart Compass or augmented-reality-type interfaces: Again, the design of the framework has implicitly assumed error-free operation of the orientation sensors. For a legitimate validation of this work, it is therefore indispensable to discuss the potential errors the system will encounter under real-world conditions; the consequences these errors have on the query process; and the subsequent ramifications on different types of mobile user interfaces that are generated from the Local Visibility Model. Hence, this section presents a validation that is based on a combination of measures:

§ The first, fundamental question the validation addresses is: *What is the range of the GPS positioning error that can be expected under normal operation?* This question is answered with a series of GPS measurements. The measurements are thereby performed so that they reflect a typical usage scenario of the framework: they are conducted using a low-cost, handheld GPS receiver, under realistic conditions, i.e. operated by a pedestrian user, in different types of urban environment.

- After an assessment of the GPS error characteristics, the second question that consequently follows is: ***What are the effects of the error on the resulting Local Visibility Model***? This question is answered by computing sample queries from collected GPS data, and comparing the resulting LVis with the real world environment at that location. The comparison is thereby done with the aid of 360-degrees panorama photographs taken at the respective test locations.
- Once the previous two questions have been answered, it is finally necessary to look at the framework's functionality in its entirety: Three illustrative scenarios will be investigated to sketch possible user interface and interaction types, utilizing different mobile device configurations: As the nominal case, a standard mobile phone with GPS will be assumed in the first example; the query result is presented in the form of a map. As the legacy case, a low-end GSM phone (without GPS or the ability to access the Web) will be assumed in the second scenario. Communication happens via SMS, and the location is inferred from a street address or landmark name provided manually by the user in the request SMS. As the high-end case – and as a potential future interaction scenario – the framework will be tested with a functional Geo-Wand prototype that was manufactured as part of the *Point to Discover* research project.
- Since this thesis claims that the LVis is of particular value for the creation of orientation aware interfaces such as the Geo-Wand, the third usage scenario is examined in more detail: the potential accuracy achieved by the system under the combined negative influence of GPS error and sensor inaccuracy is investigated in a separate series of tests; and average values for the achievable performance of the Geo-Wand in different types of urban terrain are derived.

8.1 GPS Test

As explained above, the visibility based query method proposed in this thesis implicitly assumes a sufficiently accurate location estimate. Since the framework implementation relies on GPS as the primary means of automatic positioning, it is necessary to examine the positioning accuracy that can be achieved by state of the art GPS receivers in order to assess the framework's performance under real world conditions. The authors of (Aloi *et al.* 2007) discuss several challenges involved with testing the performance and accuracy of GPS receivers: By nature, GPS tests are never fully reproducible due to non-repeatable GPS satellite constellations and changes of the environment over time (e.g. the seasonally changing density of foliage or changes due removal or addition of buildings). Also, it is generally impractical for a test person (or vehicle) to repeatedly follow the same test path with total accuracy. Due to these influences, it is difficult to obtain a viable ground truth against which GPS measurements can be compared. Therefore, GPS tests are frequently carried out in a relative fashion in practice, by comparing the system under test against a qualified truth reference system (TRS) – for example a sophisticated positioning system that combines differential GPS with an inertial measurement unit (Aloi *et al.* 2007). Since, however, the goal of the validation presented in this thesis is to verify the practical applicability of the framework and the interaction metaphors it has been designed for, rather than to measure the performance and accuracy of a particular GPS receiver model; and since the cost of a TRS would generally exceed the budget available for this thesis, such an approach is inappropriate. Instead, the validation procedure follows the simplest possible method suggested by the authors of (Aloi *et al.* 2007): the GPS receiver's position estimates are plotted against a map. While this method does not provide quantifiable statistics for positioning accuracy, it does capture large outliers and provides a qualitative insight in the performance of the tested system.

8.1.1 TEST ENVIRONMENTS

In the automotive telematics industry, three types of terrain have evolved as commonly agreed test environment types for GPS tests: *open highway*, *foliage* and *urban canyon* (Aloi *et al.* 2007). Pedestrian use of GPS is, however, different from GPS use in car-navigation: users normally move on sidewalks close to the building line; they may move in and out of buildings frequently; and they cannot necessarily be expected to carry their GPS unit or GPS-enabled mobile phone in a place that ensures ideal reception. Rather they might carry the unit in a pocket of their shirt, trousers or jacket, or stowed away in a backpack most of the time. Due to these differences, this thesis suggests to refine the terrain definitions for the pedestrian case, towards a more fine grained classification scheme. Based on the automotive test environments, the following five environment types are defined: *open environment*, *low-density urban* (*suburban*) *environment*, *park environment*, *urban environment*, and *urban canyon environment*. Since no two outdoor environments are the same, it is difficult to specify hard, quantitative thresholds for the properties of each environment. The following paragraphs therefore attempt to describe each environment in a qualitative manner, through representative examples.

Open Environment. The *open highway* driving environment known in the automotive industry refers to terrain with no or minimum obstruction of the sky and only short, momentary outages, e.g. caused by driving underneath overpasses, in short tunnels, or passing by large trucks (Aloi *et al.* 2007). In accordance to this definition, this thesis defines *open environment* as an environment with analogous properties. Examples of this terrain type are rural areas, with no or few buildings and foliage, spaced far enough away from the user to not cause any relevant obstructions of the sky. A representative example is shown in Figure 29.

Figure 29. Open environment example

Low-Density Urban or Suburban Environment. This urban environment type is characterized by a low percentage of obstructed sky. Examples are suburban areas with low (2 to 3 floor) buildings; or open spaces in inner city areas, surrounded by medium-sized buildings (up to 6 floors) but spaced sufficiently far away from the user to obstruct only a small portion of the sky. Typical examples are suburban residential zones, or medium to large inner city squares, as shown in Figure 30.

Figure 30. Low-density urban environment example

Figure 31. Park environment example

Park Environment. In reference to the *foliage* driving environment, this thesis defines *park environment* as an environment that shares the properties of *low-density urban environment*, but with the addition of close-by trees and overhead foliage, as shown in the example in Figure 31. Wet foliage, in particular, is known to cause more severe attenuation effects on GPS satellite signals than dry foliage. Furthermore, since the density of the foliage affects the amount of signal attenuation that occurs, attenuation in this environment can be expected to be a seasonal effect.

Figure 32. Urban environment example

Urban Environment. This type of environment is more densely developed than low-*density urban environment*, with medium-sized buildings (up to 6 or 7 floors) spaced closely together, narrow streets, pedestrian zones and alleyways. Examples are downtown areas and historical city centers of Western- and Central-European cities like Paris, Vienna or Prague. The properties of urban environment are likely to vary more widely than those of the other environment types; a representative example for the *urban environment* that was encountered during the tests conducted for this thesis is shown in Figure 32.

Figure 33. Urban canyon environment example

Urban Canyon Environment. The *urban canyon* driving environment known in the automotive industry is modelled according to the densely developed downtown environments of North American cities such as Chicago, Los Angeles or New York. Characterized by high-rise buildings that obstruct a high percentage of the sky, this type of terrain is the most challenging environment for GPS receivers: multipath propagation effects and frequent loss of line of sight limits the number of satellites that can be tracked, leading to reduced accuracy of the location estimate and outages where the computation of a position is not possible at all. An example of urban canyon environment encountered during the tests presented in this thesis is shown in Figure 33.

8.1.2 GPS TEST PROCEDURE

The goal of the GPS tests is to provide qualitative insight into the average GPS error that can be expected under different environment conditions. The GPS tests were carried out with a customary Bluetooth GPS unit, based on a state of the art receiver chip-set[128]. The unit was used as is, without an external antenna, and carried in the front pocket of the author's shirt, resulting in a fairly typical depiction of 'casual' GPS use under uncontrolled conditions. A mobile phone was used to record position fixes to a log file at two second intervals, while the author walked along a predefined test path. In addition to longitude and latitude values, the logging application also recorded the number of satellites the receiver used to compute the position fix, as well as the horizontal dilution of precision (HDOP) value (a quality measure reported by the GPS unit that is derived from the geometry of the current satellite constellation, and which indicates whether the constellation is favourable for computing good quality position fixes).

The test was repeated three times, at four hour intervals. Since the duration of a GPS satellite orbit is slightly below twelve hours, three tests spaced at four hours can be expected to ensure different satellite constellations for each test.

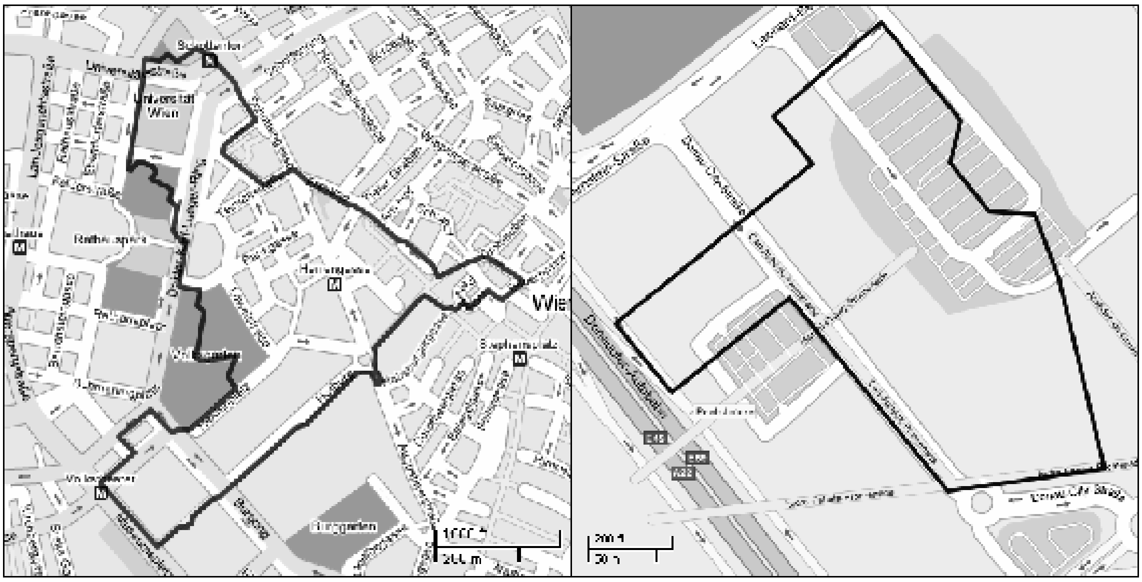


Figure 34. GPS test routes (left: inner city, right: urban canyon)

As a first test path, a circular route with a length of approx. 5.2 km was chosen in the inner city district of Vienna (see Figure 34 left). The route was selected such that it leads through the three environment types of *low density urban*, *park*, and *urban environment* at about equal shares. The route took the author between 47 and 52 minutes to complete (49 min, 47 min, and 52 min, respectively). Due to the absence of distinctive *urban canyon* environment in the inner city of Vienna, a second test path of about 1.4

---

[128] GlobalSat BT-338 SiRF Star III (http://www.globalsat.com.tw/eng/product_detail_00000039.htm)

km length was selected in a business district in the North of Vienna (see Figure 34 right). This route took the author 12 to 13 minutes to complete (12, 13 and 13 min respectively). No tests were conducted in *open environment*: since this type of terrain provides optimum conditions that do not stress a GPS receiver's tracking capabilities, evaluating GPS performance here is least insightful (Aloi *et al.* 2007). With regard to this thesis, it is expected that *open environment* is the least problematic environment and will yield results at least as good as in *low-density urban environment.*

The recorded GPS tracks were analyzed according to their approximate deviation from the reference path on the map. As a simple measure for the deviation, the normal distance to the reference path was computed for each position fix. Since this distance is not necessarily exactly the same as the distance between the test person's real location and the position fix, and since the reference path itself can not be assumed to be 100% accurate, the deviation measure can not be taken as an absolute statistic, but rather a qualitative indication of the GPS fix quality.

8.1.3 GPS TEST RESULTS: INNER CITY ROUTE

Figure 35 shows the results from the three GPS tests conducted along the inner city route, along with the HDOP value reported by the GPS receiver. A marker in the track image indicates the start location; the route was traversed in clockwise direction. Additional markers on the track image, which relate to the different types of environment, allow a visual mapping between track and deviation plot. Throughout all tests, 6 to 9 satellites were tracked by the receiver at all times (with the exception of an underpass where the number of tracked satellites would temporarily fall to 3 or less, as discussed below).

As can be seen in Figure 35, the three recordings show largely corresponding results: The route starts in densely developed *urban environment.* The narrowest alleyways are located on the right-most portion of the path (see Figure 35, marker 'URBAN'), with deviations in this area in the range of not below 20 meters. After traversing through more urban terrain for approx. 200 meters, the route passes through an underpass. This can be best observed in the third test, where a singular spike of approx. 300 meters was recorded due to temporary loss of GPS line of sight.

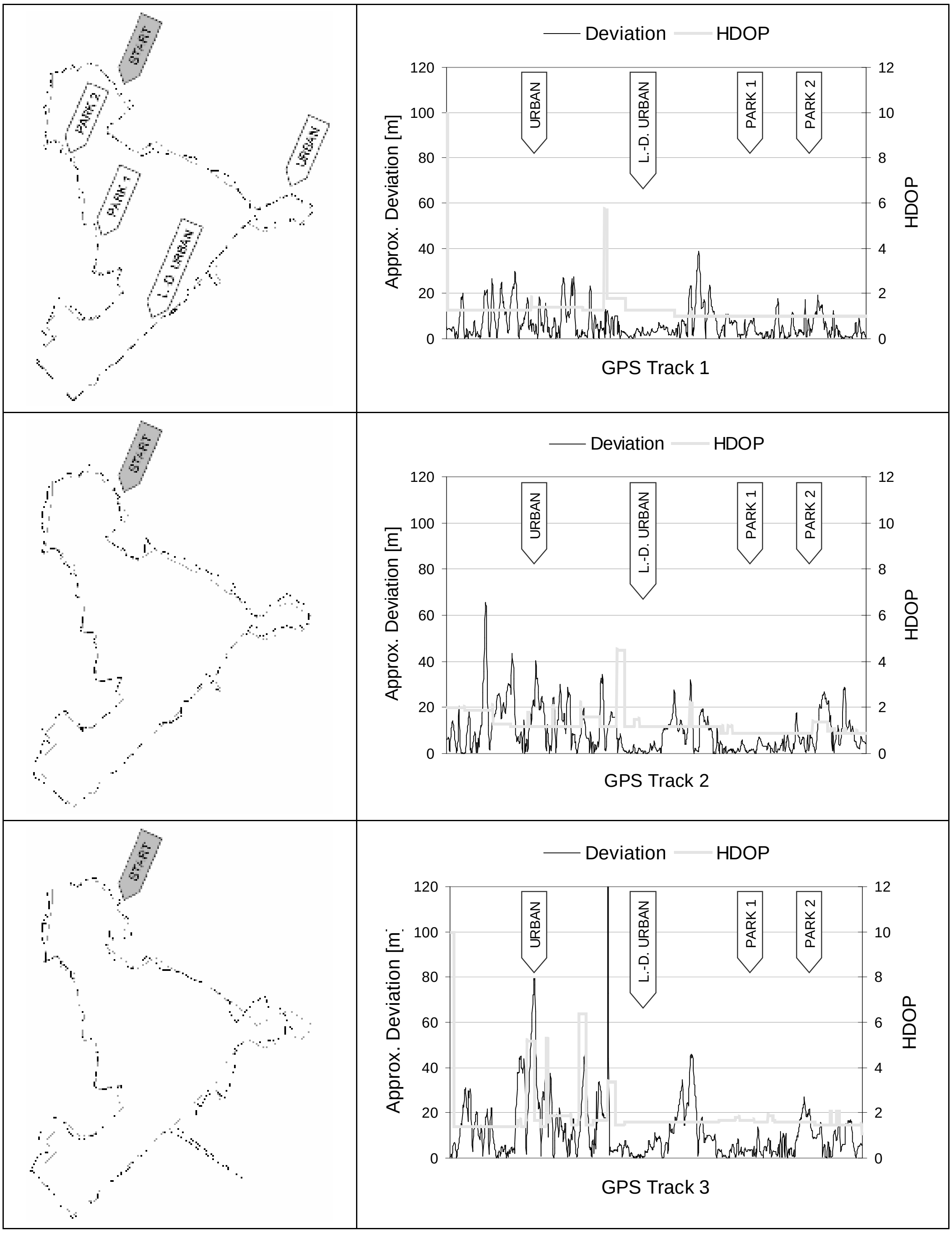


Figure 35. Approx. normal distance from map reference path and HDOP (inner city tests)

The route then enters a large square (i.e. *low-density urban environment*; see Figure 35, marker ‘L.-D. URBAN’). Deviation from the reference path drops considerably, below 10 meters or less. After a short portion through more densely developed terrain, the route enters *park environment*. Deviation from the path is in the range of 10 meters in the first park (see Figure 35, marker ‘PARK 1’), and slightly higher in the second park (around 20 meters, see Figure 35, marker ‘PARK 2’). Most likely, this difference can be attributed to the fact that in the second park, the path passes underneath denser foliage than in the first park.

Summarizing, it can be concluded that in *urban environment*, it is reasonable to expect an error in the range of at least 30 to 40 meters with a state of the art handheld GPS receiver. In *low-density urban environment*, the tests showed consistently good performance with maximum deviations from the path in the range of 10 meters or less in all tests. In the *park environment* portions, path deviation was shown to be similar or slightly above the *low-density urban environment* case. Another effect that can be observed in Figure 35 is that GPS inaccuracy was most pronounced in the third test: Most likely, this results from a less favourable GPS constellation at the time of the test. As can be seen in Figure 35, however, there is no noticeable correlation between the HDOP value and the approximate deviation during the tests; the HDOP is only slightly higher than in the first two tests. It can therefore be assumed that multipath effects were a primary source of error in the *urban environment* parts of the route, rather than loss of satellite signals. Hence, the HDOP alone can not necessarily be taken as a reliable quality indicator in this type of terrain.

8.1.4 GPS TEST RESULTS: URBAN CANYON ROUTE

Figure 36 shows the results from the three tests conducted along the *urban canyon environment* test path. As before, a marker in the track image marks the start of the route, which was traversed in counter-clockwise direction. 6 to 9 satellites were tracked by the receiver at all times.

Due to the more homogenous environment on the *urban canyon* route, the results recorded on this test path were more uniform: the approximate deviation from the reference path was measured to be in the range of at least 20 to 40 meters and more. As on the first test route, the HDOP value did not show any noteworthy correlation with the magnitude of the estimated GPS offset.

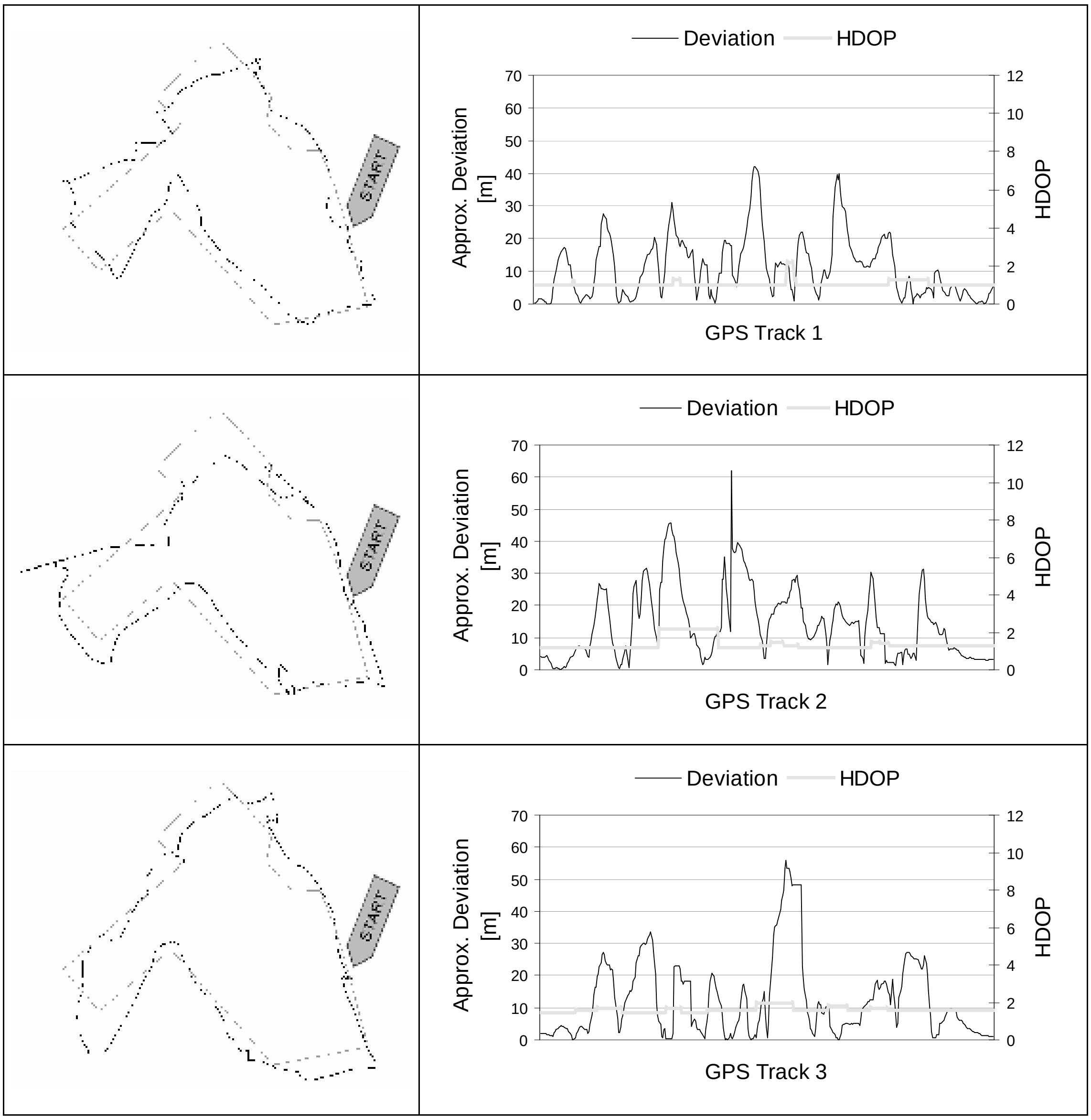


Figure 36. Approx. normal distance from map reference path and HDOP (urban canyon tests)

8.2 Effects of GPS Error on the Local Visibility Model

As a next step, it is investigated how GPS inaccuracy affects the framework's query process: The goal of this second validation stage is to illustrate the distortion that is introduced to the Local Visibility Model due to positioning errors, and to provide an assessment of the potential influence this distortion may have on the perceived quality of the user interfaces derived from it. Obviously, user interfaces that rely on a “physical world browsing” interaction metaphor – such as the Geo-Wand or potential future mobile phone augmented reality interfaces – will have stricter requirements on the positioning accuracy and, hence, be more susceptible to GPS error than simpler – e.g. text- or map-based – interfaces. The test conducted in this validation stage is therefore interested primarily in the LVis’ suitability for these high-end interaction styles.

Again, the presented test results shall provide qualitative insights into the distortion effects, rather than quantitative statistics: not only are these difficult to obtain without employing an expensive truth referencing system; they are also of limited value in practice, since they are highly dependent on the particular type of GPS receiver and client device used, and may be influenced by specific implementation details of the framework and the query engine (such as the quality of the block model, the availability of a vegetation model in addition to a building model; or the employment of map matching techniques to compensate for GPS errors, as will be discussed in Section 10.3). As the goal of this validation is not to test a specific implementation or device setup, but rather the concept of the Local Visibility Model as such, it is therefore more instructive to present specific examples, and to discuss concrete error situations that are likely to be encountered in the real world on a general level; and to draw qualitative, but generalizable conclusions from these examples.

The tests presented below therefore examine a number of sample LVis queries taken at representative locations. Their results are compared against the user’s real field of view (FOV), using 360-degrees panorama photographs. The GPS error is again estimated from a map. A more detailed evaluation of how the distortion effects manifest themselves on particular user interface types (e.g. by leading to an increased “pointing error rate” for the Geo-Wand) is presented in Section 8.3.

8.2.1 TEST PROCEDURE

In order to categorize the locations that were chosen for the LVis tests, the same environment types as defined for the GPS tests were used. A total of twelve locations, situated along the test routes from the GPS tests, were selected; with three locations in each type of environment, i.e. 9 of the test points are located along the inner city test route, and 3 test points are located along the urban canyon test route, as shown in Figure 37. For each location, a 360-degrees panorama photograph was produced by the author. This was done by combining several consecutive photographs taken with a digital SLR camera to a single image with a commercial photo stitching software.

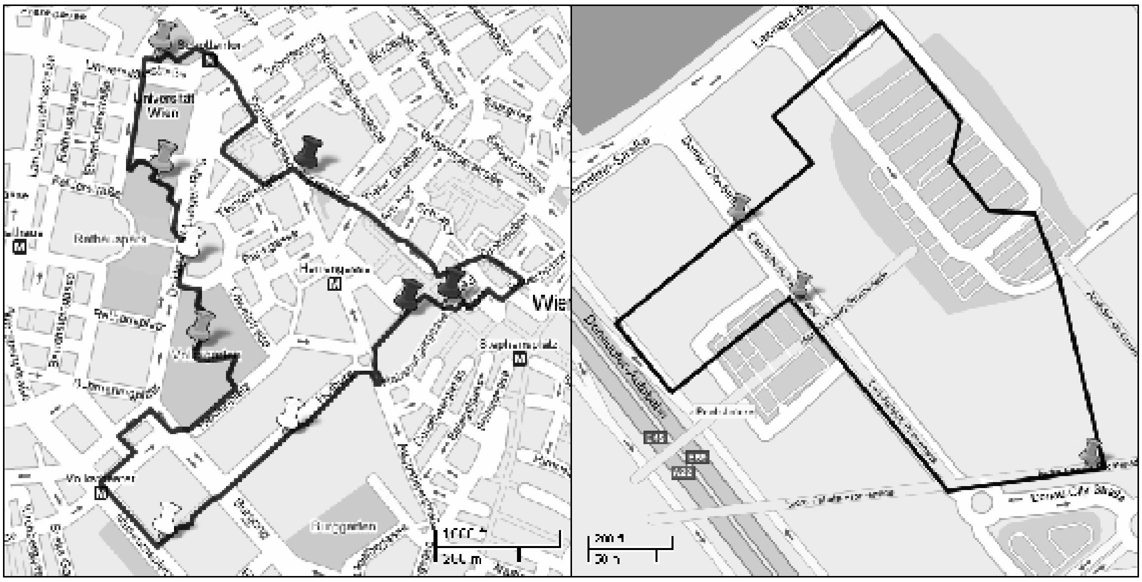


Figure 37. LVis sample locations on GPS test routes: low-density urban (white, left), park (gray, left), urban (black, left), and urban canyon environment (right) locations

To conduct the tests, the author approached each test location, remained stationary for at least 10 seconds, and then sampled a single GPS measurement out of the data stream delivered by the GPS. (The same Bluetooth GPS unit as in the GPS tests was used, again carried in the author's shirt pocket.) This sample value was then used as input to the query engine. From the resulting LVis, a top down map view, as well as a panoramic projection of the corresponding billboards was produced, using the visualization tools developed as part of this thesis (compare Section 7.2.3).

The billboard panorama projection was overlaid on the photograph in order to allow detailed visual judgement of the correspondence between the user's real FOV and the LVis. Please note that the flat projection used by the billboard visualization tool does generally not correspond well to the projection produced by a photographic camera: for the sake of simplicity, the tool maps the vertical angles used to describe the billboards' height to a linear scale; the vertical placement and absolute height proportions of the billboards on the panorama photographs is therefore not to be taken as an exact measure. The horizontal placement, as well as the relative proportions between individual billboards, however, is exact.

8.2.2 LVIS EXAMPLES: LOW-DENSITY URBAN ENVIRONMENT

The three test locations chosen in *low-density urban environment* are represented on Figure 37 (left image) as white push-pins: the first and the second location are both situated at large, inner city squares (the *Heldenplatz* and the *Museumsplatz*, respectively); with a mostly unobstructed view of the sky. The third location (*Burgring*) is situated next to a broad street; the area is delimited by a large building to the East direction, the other directions face towards mostly open terrain.

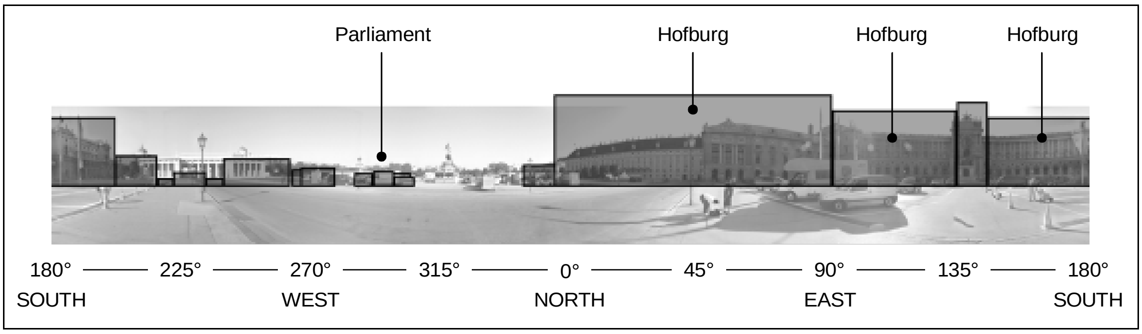


Figure 38. Low-density urban example 1 (Heldenplatz), panorama view

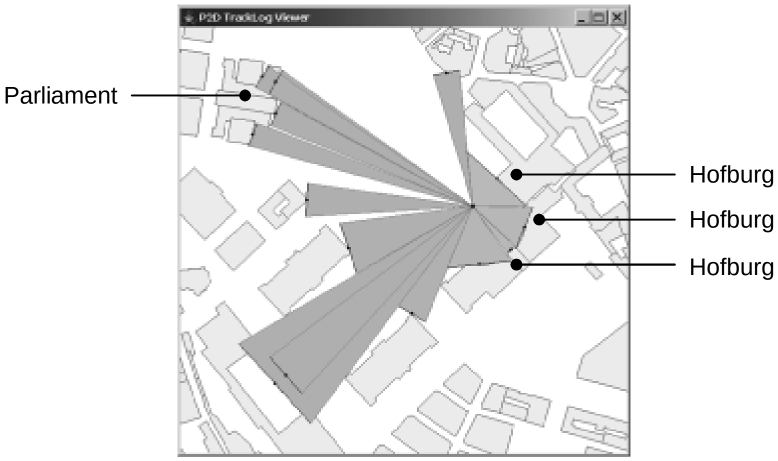


Figure 39. Low-density urban example 1 (Heldenplatz), map view

Example 1: Heldenplatz. The *Heldenplatz* represents the largest open space that is contained in the environment block model used for this implementation. The GPS tests in this area yielded an estimated GPS error of about 10 meters or less. As expected, an excellent correspondence can be observed between the LVis (compare Figure 39) and the panorama photo, shown in Figure 38: The most prominent buildings in the FOV (i.e. different parts of the *Hofburg* to the North, East and South are closely matched by their billboard representations; as are buildings further away (such as the Parliament building in the North-West). The small gate that can be seen in the photograph in the South-West area is missing in the block model. Hence, the billboards represent the buildings behind the gate, rather than the gate itself.

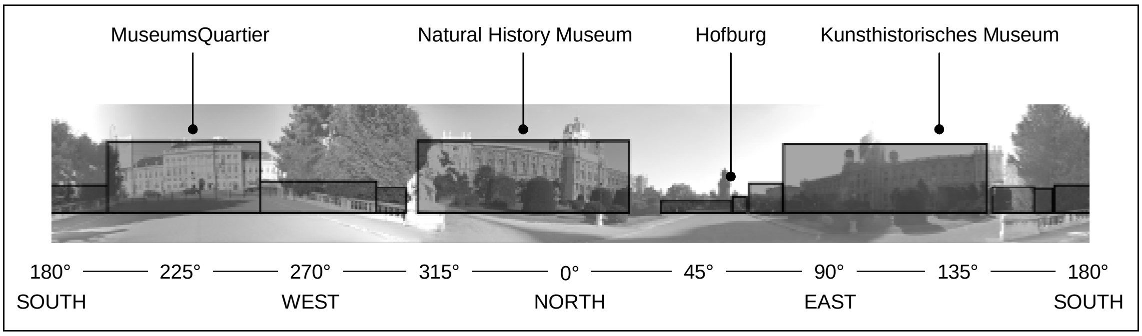


Figure 40. Low-density urban example 2 (Museumsplatz), panorama view

Figure 41. Low-density urban example 2 (Museumsplatz), map view

Example 2: Museumsplatz. As on the previous test location, the estimated GPS error on the second test location (*Museumsplatz*) is minimal. Accordingly, the panorama photo again shows excellent correspondence with the billboards contained in the LVis, as can be observed in Figure 40 and Figure 41. The billboards for the three dominant buildings in the FOV (the Natural History Museum to the North, the *Kunsthistorisches Museum Vienna* to the East, and the *MuseumsQuartier Wien* to the South-West) are accurately positioned and suitably proportioned. The three distant billboards to the North-East, which represent different sections of the *Hofburg*, also show excellent correspondence with the *Hofburg's* real location.

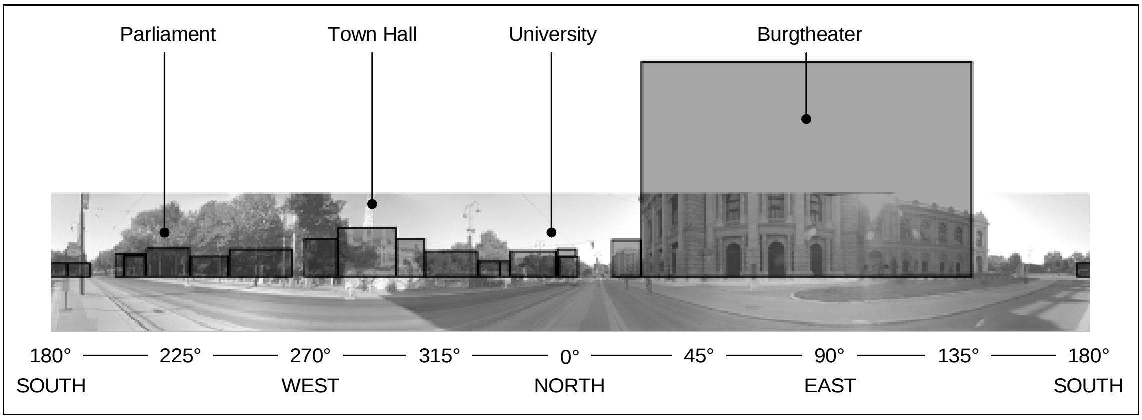


Figure 42. Low-density urban example 3 (Burgring), panorama view

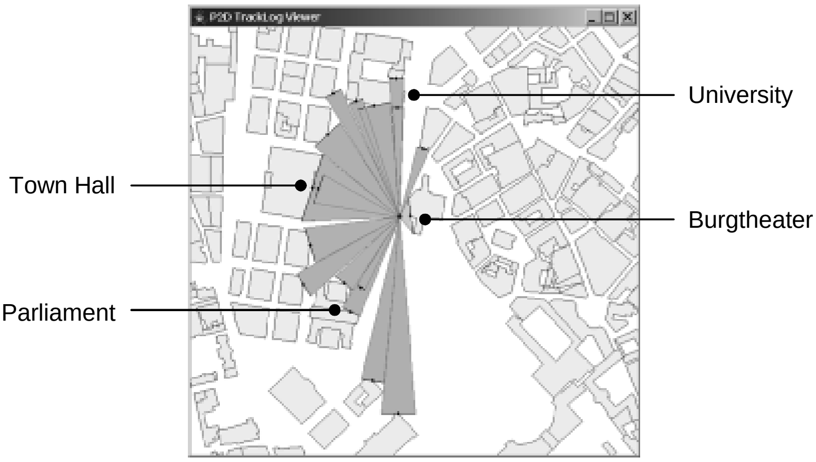


Figure 43. Low-density urban example 3 (Burgring), map view

Example 3: Burgring. The third test location (*Burgring*) was chosen so that the FOV is mostly open; except for one large, nearby building (*Burgtheater*), which blocks a considerable amount of the clear view of the sky. Despite the fact that the GPS tests have yielded results slightly below the quality of the first two locations, the correspondence between the panorama photo and the sample LVis that was observed for this particular sample is excellent (see Figure 42 and Figure 43): The *Burgtheater* billboard overlaps with the photograph almost exactly, as do the billboards for other prominent buildings in the vicinity, such as the Town Hall to the West or the University building to the North. (The accurate location and width of the “gap” between billboards in the North, where the street passes the University can be observed particularly well.) Several other billboards can be seen in Figure 42 which represent buildings that are actually not visible from the user's point of view due to obstruction by trees (which are not captured in this implementation's environment model) such as the Parliament building to the South.

8.2.3 LVIS EXAMPLES: PARK ENVIRONMENT

The GPS test route defined in Section 8.1 traverses through three different parks. One sample LVis query was performed in each one. The first park (***Volksgarten***) is characterized by relatively wide, open spaces and walkways with little overhead foliage. The other two parks (***Rathauspark*** and ***Votivpark***) are smaller, with generally more foliage near the walkways. Overhead foliage was also present at both test locations. The ***park environment*** test locations are shown in Figure 37 (left image) as gray push-pins.

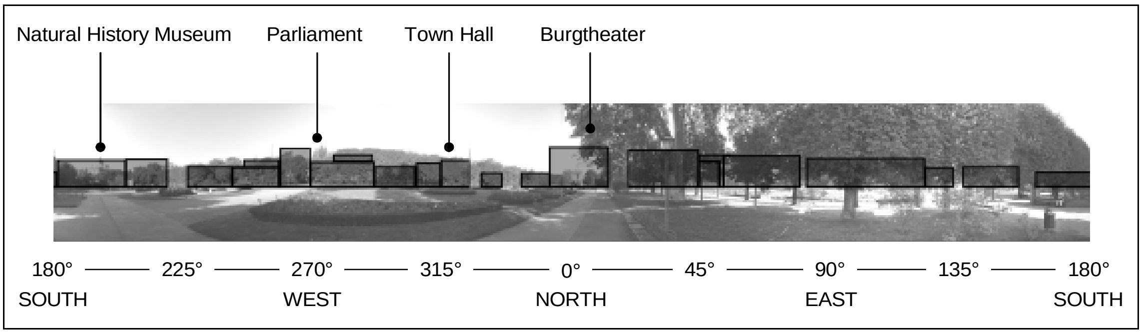


Figure 44. Park example 1 (Volksgarten), panorama view

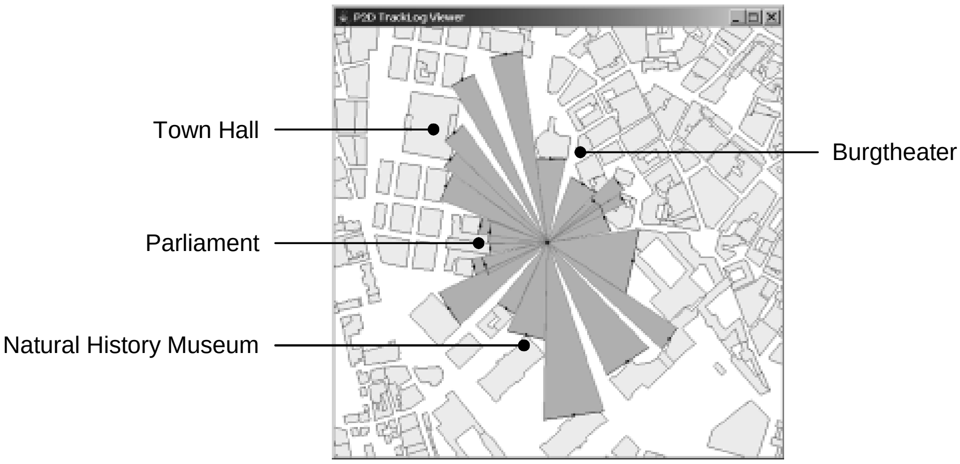


Figure 45. Park example 1 (Volksgarten), map view

Example 1: Volksgarten. The GPS tests in the first park area (*Volksgarten*) yielded a high positioning accuracy. The offset was estimated to be in the range of about 10 meters, which is similar to the test areas in low-density urban environment. As expected, there is again good correspondence between the LVis and the FOV. The most prominent visible buildings – such as the Natural History Museum, the Parliament building, the Town Hall or the *Burgtheater* – are accurately represented. As can be seen in Figure 44 and Figure 45, the LVis contains several billboards that represent buildings which are actually obstructed by vegetation in the real FOV: i.e. there is again a mismatch between what is really visible to the user, and the visibility information captured in the LVis due to the fact that the environment model does not consider vegetation. Depending on the type of application, this circumstance might either be problematic (since the LVis is practically erroneous in areas with foliage) or desirable (because of the ability to "look behind" vegetation).

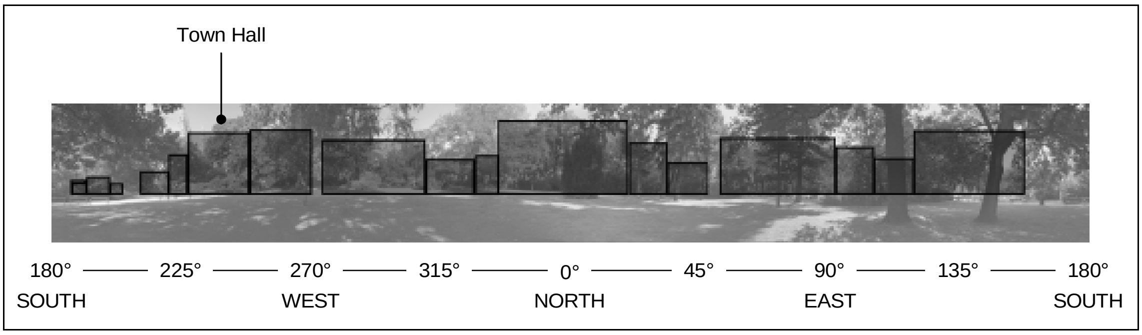


Figure 46. Park example 2 (Rathauspark), panorama view

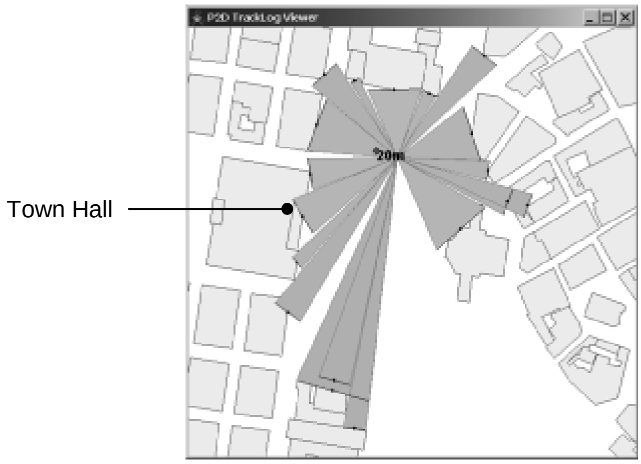


Figure 47. Park example 2 (Rathauspark), map view

Example 2: Rathauspark. Due to the higher amount of overhead foliage, positioning accuracy was measured to be slightly lower in the area of the second *park environment* test location (*Rathauspark*). In accordance with the GPS test results, the offset estimated for the sample query is in the range of about 20 meters (see Figure 47). As can be seen in the panorama image in Figure 46, there are noticeable offsets: e.g. about 10° in the case of the Town Hall. Since the Town Hall covers roughly 40° of the FOV horizontally, there is still reasonable overlap between the building and the billboards. However, the most noteworthy effect is again that the billboards contained in the LVis primarily represent buildings that are actually obstructed by vegetation in the real FOV.

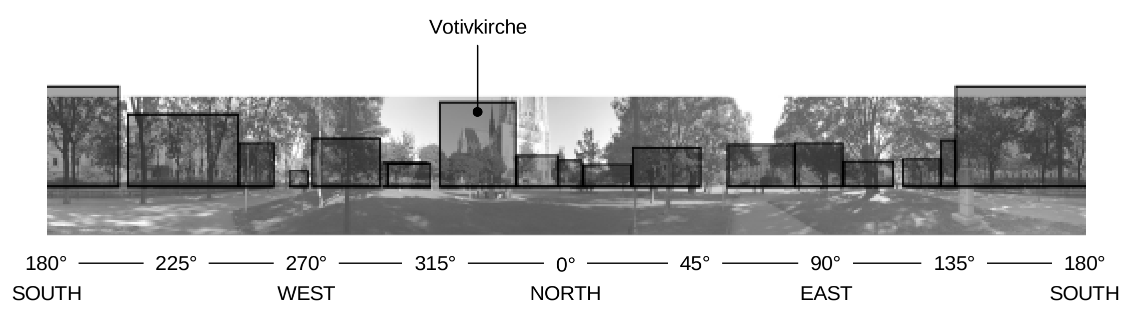


Figure 48. Park example 3 (Votivpark), panorama view

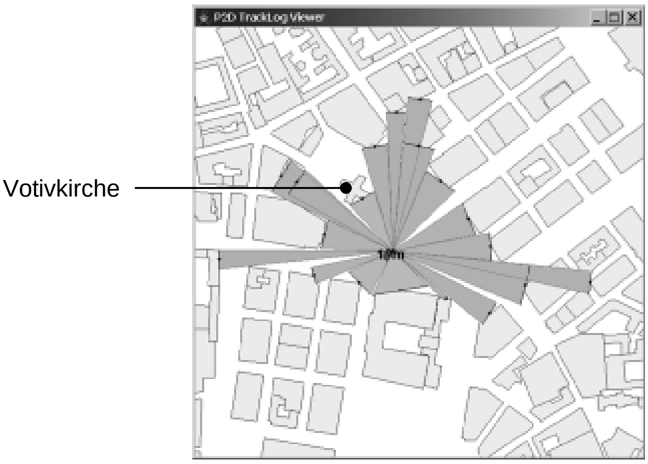


Figure 49. Park example 3 (Votivpark), map view

Example 3: Votivpark. The third test location is characterized by surrounding and overhead foliage that is less dense than in the previous test location, but denser than in the first *park environment* test location. The GPS offset for the LVis sample was estimated to be in the range of about 10 meters (compare Figure 49). The billboard for the most prominent building – the *Votivkirche* to the North – is offset from its true location by about 18° (at a width of about 27°, i.e. there is still more than 50% overlap). Again, most other billboards represent buildings that are only partially visible through the foliage.

8.2.4 LVIS EXAMPLES: URBAN ENVIRONMENT

As the GPS tests confirm, *urban environment* is among the most challenging environments for GPS receivers. Consequently, it can be expected that the correspondence between the LVis and the real FOV will degrade notably in this type of environment. The three chosen test locations share comparable properties: each is situated in a densely developed inner city area, surrounded by medium-sized, multi-story buildings; the first location on a small square (*Freyung*); the second location (*Graben*) along an inner city pedestrian zone; the third location (*Kohlmarkt*) in a narrow alleyway. The locations are shown as black push-pins in the left image of Figure 37.

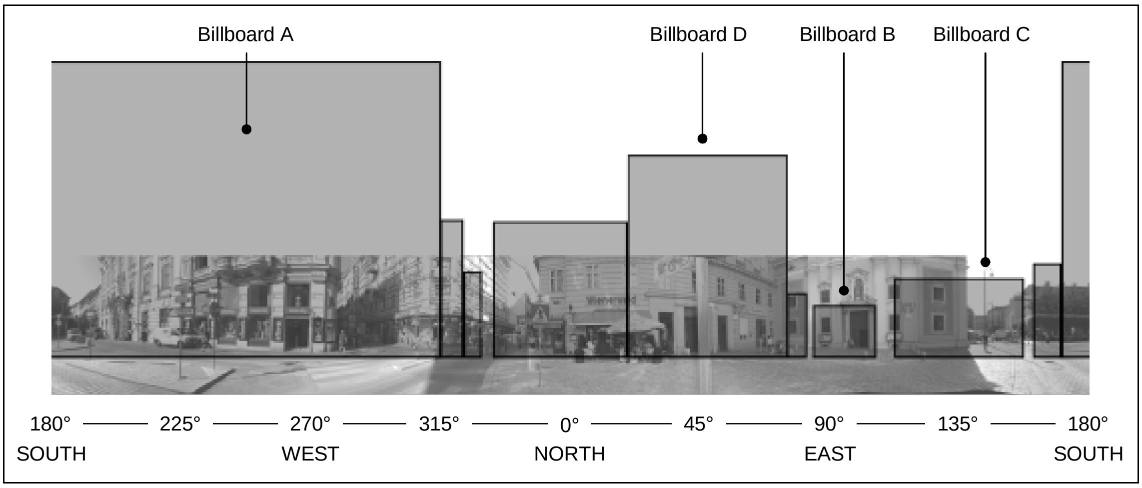


Figure 50. Urban example 1 (Freyung), panorama view

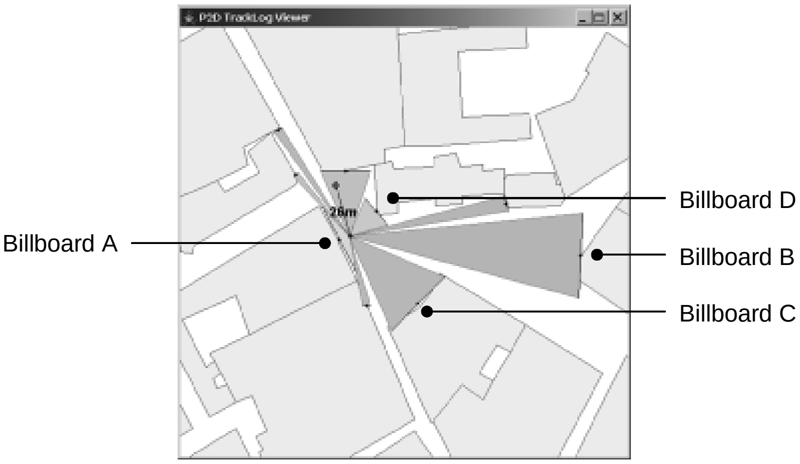


Figure 51. Urban example 1 (Freyung), map view

Example 1: Freyung. The first test location is closely surrounded by buildings that obstruct a large portion of the clear view of the sky. While the estimated GPS offset for the sample is only about 26 meters – which is relatively low in the given environment type – the consequences are considerable, as can be seen in Figure 50 and Figure 51: because the distance between the user and the surrounding buildings is much lower than in the previous environment types, the effects of a shift of 26 meters on the geometric arrangement of the billboards are much more severe than would be the case e.g. in *low-density urban environment*: For example, the user's real world position is located opposite a street leading to the South-West. Due to the GPS offset, the query engine assumes a position where the view down this street is fully blocked by the building South of the street ('Billboard A'). Other instructive examples for errors in the LVis include the following:

- § The building to the East is hidden, as seen from the user's real point of view. Due to the GPS error, however, the query engine assumes this building to be visible; and its billboard ('Billboard B') erroneously covers almost 22° of the panorama.
- § The building to the South-East – which is quite dominant in the FOV, covering roughly 35° of the panorama – corresponds to a billboard which is about 44° wide (due to the fact that the GPS offset shifts the query location closer to the building), and shifted Eastward by about 35° ('Billboard C'). As a consequence, there is practically no overlap between billboard and building in the FOV.
- § A similar problem can be observed with the building East of the user's real location: due to the GPS offset, the 55°-wide billboard is shifted Northward by about

55°, leading to a situation where there is practically no overlap between the billboard and the building it represents (‘Billboard D’).

- The building to the South-West is represented too big in the LVis due to the GPS shift: while in the real FOV, it covers about 83°, the billboard representation (‘Billboard A’) occupies 150° degrees of the panorama. Whereas on the one hand, this means that there is full overlap between the building and the billboard, it also means that the billboard obstructs other buildings or areas it is not meant to cover.

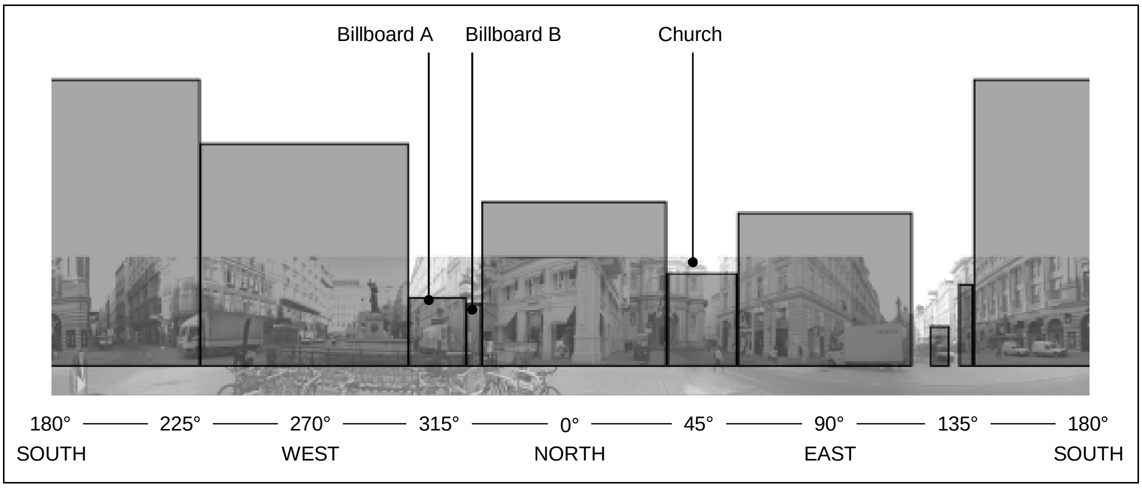


Figure 52. Urban example 2 (Graben), panorama view

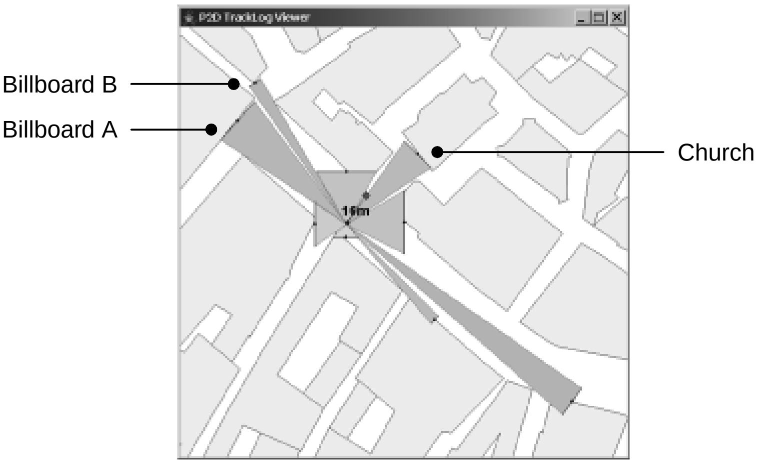


Figure 53. Urban example 2 (Graben), map view

Example 2: Graben. The conditions on the second *urban environment* test location (*Graben*) are comparable to those on the previous location: GPS error is estimated to be in the range of about 18 meters, which, in practice, corresponds to a displacement of the user's position from the true location to the other side of the street. The four closest (and, hence, most dominant) buildings directly to the North, East, South, and West, respectively (compare Figure 53) are modeled relatively accurate, i.e. there is a high degree of overlap (more than 50%) between billboards and buildings in the FOV. Other instructive cases that can bee observed in Figure 52 include the following:

- As explained in the previous example (*Freyung*), GPS errors that place the user's location away from a street crossing (i.e. a location where the user faces directly down an alleyway) can lead to distinct errors, where buildings down the street which are visible in the real FOV will become obstructed in the LVis. In the particular case of the *Graben* test location this effect is less severe, since the GPS offset happens to be along the direction of the crossway. For example, the area of overlap between the church (compare Figure 52) and its billboard representation is still in the range of 50%. This would generally not be the case if the GPS offset would have happened in the perpendicular direction.
- The two buildings to the North-West, on the other hand, are most affected by the offset in the South-West direction: While the width of both billboards ('Billboard A' and 'Billboard B') sufficiently corresponds to the properties of the real FOV (since the distance to the buildings has not changed much), there is a distinct offset of around 25°. Since the billboards are only about 20° and 5° in width, respectively, this amounts to a situation where is no more overlap between billboards and buildings in the FOV.

Example 3: Kohlmarkt. The third test location (*Kohlmarkt*) is situated on a three-way street crossing on a narrow shopping alley. The GPS error for the sample was estimated to about 12 meters; again a relatively low value for this type of environment. Unlike in the previous example, however, the direction of the offset is less favorable this time, as can be observed in Figure 55: the error shifts the query location away from the street crossing, into the crossway, thus blocking the line of sight to most of what is otherwise visible in the user's real FOV. The billboard that represents the building to the North is about twice the width it should be, obstructing the entire North-East half of the shopping alley and large parts of the building to the East and South. Similarly, the billboard that represents the building to the West blocks the entire South-West half of the alley, thus obstructing all objects which are actually visible down the street.

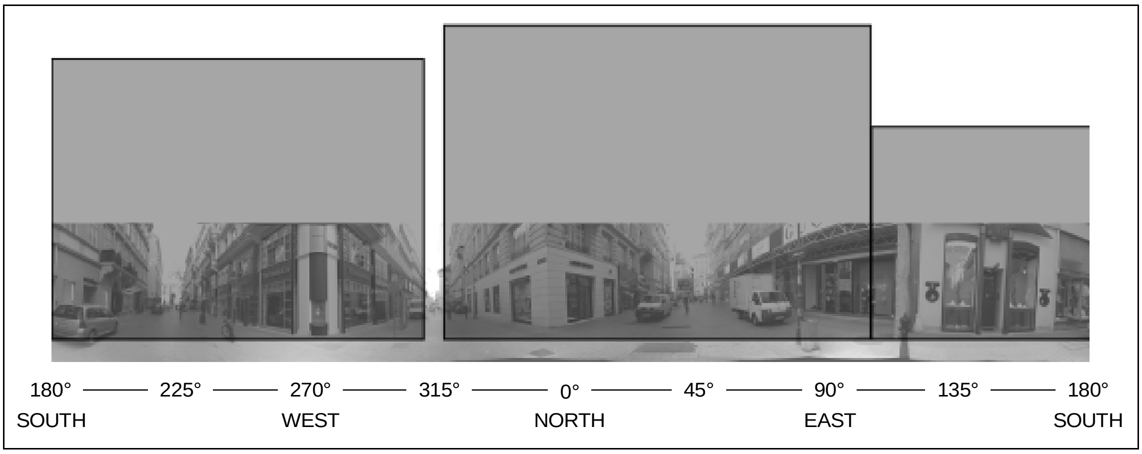


Figure 54. Urban example 3 (Kohlmarkt), panorama view

Figure 55. Urban example 3 (Kohlmarkt), map view

8.2.5 LVIS EXAMPLES: URBAN CANYON ENVIRONMENT

The three test locations that were chosen in *urban canyon* environment are represented as gray push-pins in the right image of Figure 37. Due to the unavailability of professionally surveyed data, the block model for the area was hand modeled by the author as explained in Section 7.2, using satellite images from *Google Maps* as a visual guide. As a result, the proportions and heights of the buildings in the model do not always correspond perfectly with the real world environment (which acts as an additional source of error in the LVis). Also, some buildings that exist in the real world are missing in the model, as will be explained in the examples below.

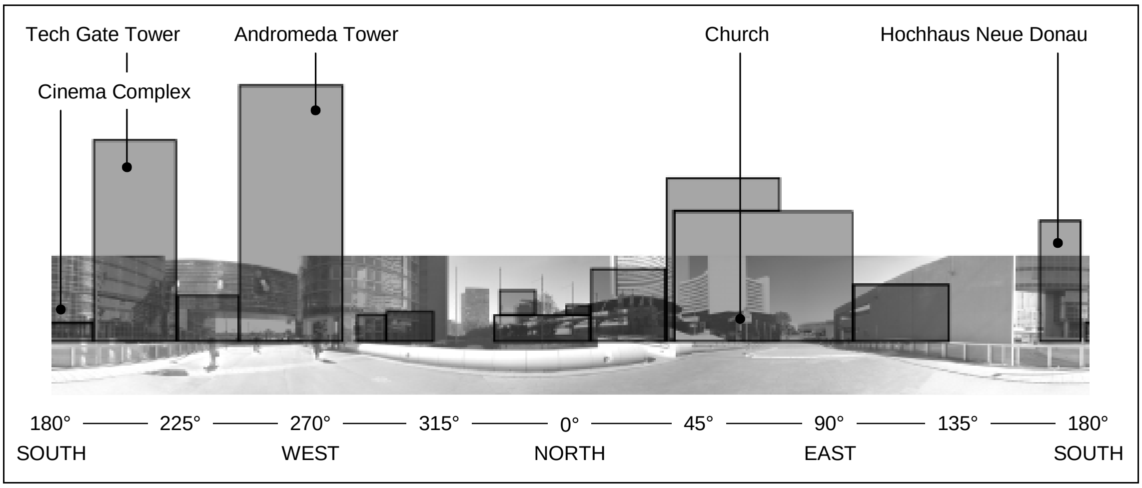


Figure 56. Urban canyon example 1 (Tech Gate Tower), panorama view

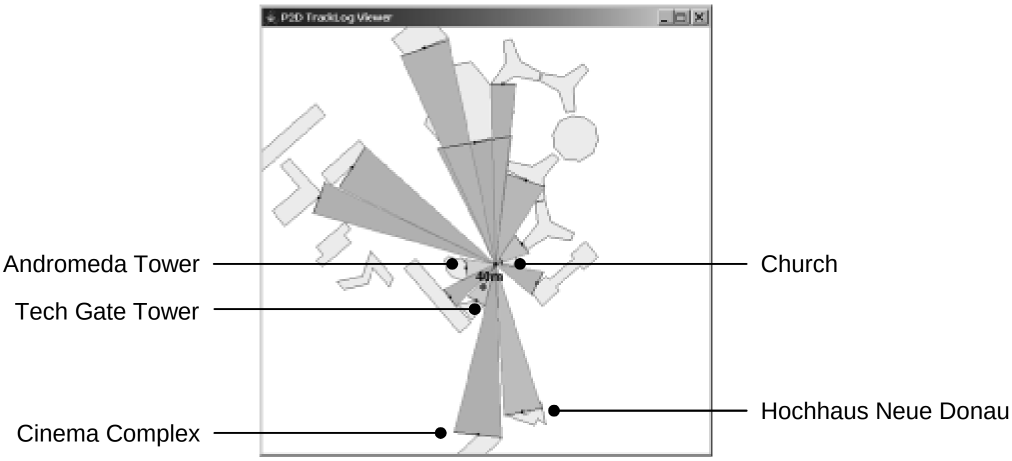


Figure 57. Urban canyon example 1 (Tech Gate Tower), map view

Example 1: Tech Gate Tower. The first test location is situated next to two tall office towers, the *Tech Gate Tower* and the *Andromeda Tower*, located to the South and North-West, respectively. To the North and East, the FOV is relatively open. As shown in Figure 57, the GPS position is offset from its true value by an estimated 40 meters. This offset leads to considerably displaced billboards for the visible buildings, and worse, to the inclusion of billboards for buildings which are in fact hidden from the user real world FOV: For example, the high rise building far South (*Hochhaus Neue Donau*) and the adjacent cinema complex (compare Figure 57) depicted in the LVis are actually hidden behind the Tech Gate Tower. Similarly, the housing complexes in the North-

West are hidden behind the Andromeda Tower in reality. Other error examples include the distortion of the billboards representing the *Tech Gate Tower* and the *Andromeda Tower*, respectively (see Figure 56), and the massive size distortion of the billboard that represents the small, cube-shaped church North-East of the users real location.

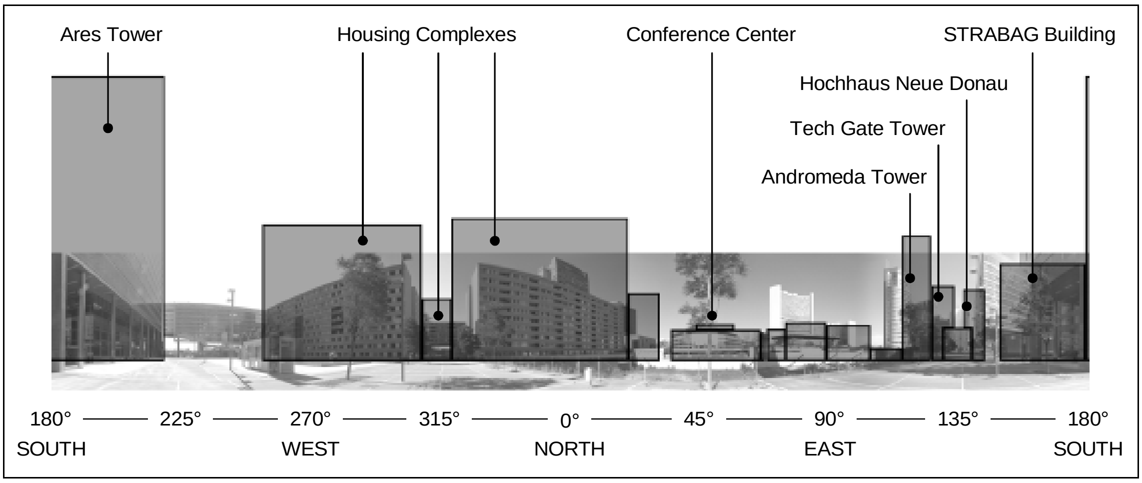


Figure 58. Urban canyon example 2 (Carl-Auböck-Promenade), panorama view

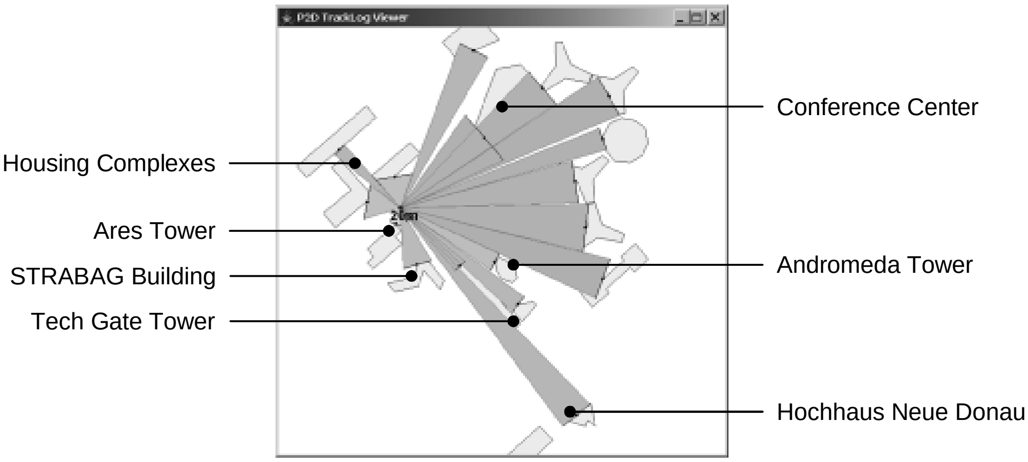


Figure 59. Urban canyon example 2 (Carl-Auböck-Promenade), map view

Example 2: Carl-Auböck-Promenade. The second test location (*Carl-Auböck Promenade*) is delimited by tall office towers to the South (the *Ares Tower* and the *STRABAG Building*, respectively). The remaining FOV is mostly unobstructed, leading to a relatively open view of the sky (compare Figure 59). The GPS position was estimated to be offset by about 20 meters. Consequently, there is good correspondence between the

LVis and the FOV. The most prominent buildings – such as the housing complexes to the North-West and North, the conference center to the East and the adjacent office towers – are placed accurately with little errors and good overlap. A slight offset of about 9° can be observed for the group of office towers to the South-East: Due to their narrow width, which is about in the same order of magnitude (9° for the *Andromeda Tower*, 7° for the *Tech Gate Tower*, and 8° for the *Hochhaus Neue Donau*, see Figure 58), the area of overlap between billboard and building is seriously diminished. However, their distinct shape makes it easy to visually relate between buildings and their billboard representations. This may be an advantage for some types of user interfaces that rely on visual browsing metaphors, such as augmented reality interfaces.

Example 3: Wohnpark Donau City. The third test location represents the most distinct case of *urban canyon environment* that was encountered in this validation. The location is situated in between several blocks of flats that are part of a large housing complex (*Wohnpark Donau City*). As shown in Figure 61, the GPS estimate was offset by approx. 15 meters, which, in fact, represents a remarkably low error given the surrounding environment. Despite the low offset, however, the consequences on the LVis are severe: The query location is shifted behind one of the building blocks, leading to a situation similar to the third *urban environment* example (Kohlmarkt): Most of the buildings that are visible from the user's real point of view (i.e. the characteristic skyline consisting of the *Andromeda Tower*, the *Tech Gate Tower* and the *Hochhaus Neue Donau*, as seen in the previous example) are obstructed by the large billboard that represents the building to the East ('Billboard A'), and which is about twice the width it should be. The billboard that represents the *Ares Tower* to the South is offset by about 18°, resulting in a situation where there is no noticeable overlap between billboard and building. The low structure directly to the North is not contained in the hand-crafted environment model; hence, the billboard in this direction actually represents the building behind the structure.

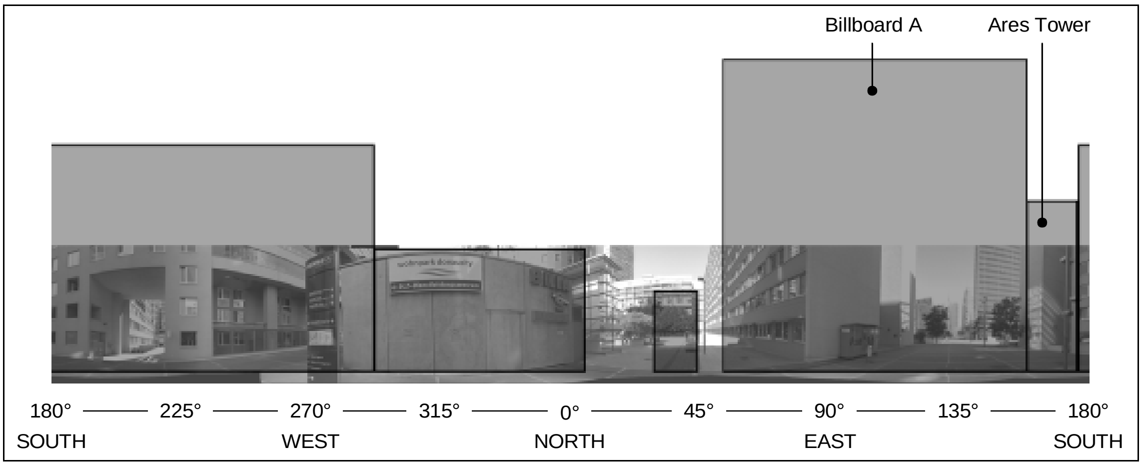


Figure 60. Urban canyon example 3 (Wohnpark Donau City), panorama view

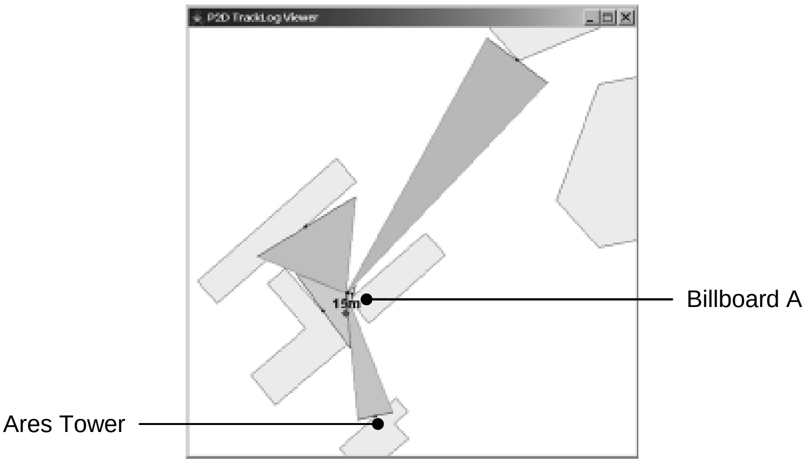


Figure 61. Urban canyon example 3 (Wohnpark Donau City), map view

8.2.6 CONCLUSIONS

A first, unsurprising conclusion that can be drawn from the examples presented above is that, naturally, GPS error plays a crucial role for the quality of the query result. Far more importantly, however, the results clearly show that knowledge of GPS error alone is insufficient to judge how well or how bad real-world browsing metaphors can perform in practice. The constructional properties of the environment have an at least equally strong influence: they are the deciding factors that determine whether an offset of 15 meters has no noticeable effect on the quality of the query result, or a disastrous one. As a noteworthy side result, the presented examples already provide a first indication of which factors may be suitable to assess the quality and performance of future real-world browser implementations: For example, the tests suggest that *area of overlap* – i.e. the fraction of a building in the real FOV that is covered by its corresponding virtual representation (the billboard) – and *area of obstruction* – i.e. the fraction of the virtual representation that covers regions other than the corresponding building – may be viable metrics for the quality of the query result, and the user interface derived from it. While these metrics can be captured in statistics in principle, not all factors that influence the perceived quality of a particular system implementation will be quantifiable; or will apply to any form of interaction metaphor that can be derived from the query result: For example, the LVis' property to preserve the shape of the skyline may be a beneficial feature for certain types of graphical interfaces, since they allow the user to visually map the virtual representation with the objects in the real FOV, even if there is no or little area of overlap (as explained in Section 8.2.5). On the other hand, this property might be irrelevant for non-graphical interfaces such as the Geo-Wand.

It is difficult to predict which forms of user interfaces will prevail as an effective and usable means to "browse the physical world". More HCI research is needed to test and validate different types of user interfaces and their suitability for different tasks and contexts (Fröhlich *et al.* 2007). Even though it is beyond the scope of this thesis to address these HCI issues, the presented framework can potentially serve as a prototyping tool for future research into this direction; and the LVis can be flexibly used to experiment with various types of geospatial interfaces on a wide range of mobile devices. The following sub-section discusses three concrete examples for such mobile geospatial interfaces that can be realized with the framework implementation in detail.

8.3 Example Test Scenarios

The previous stage of the validation has established a general understanding of the types of errors that occur under real world conditions, the situations that cause them, and the magnitude of their effect on the LVis. It was concluded that in order to judge how real world browsing metaphors can perform in practice, it is necessary to consider the quality of the position estimate, as well as the constructional situation in the environment where browsing takes place in a balanced way. It was furthermore argued that different types of user interfaces are likely to be more or less susceptible to certain error characteristics that appear in the LVis. The goal of this third and final stage of the validation is therefore to apply the LVis to three concrete geospatial interaction scenarios; and to examine how they reflect the varying quality of the query result differently:

- The first scenario illustrates the use of the framework with a client device that can most likely be considered a nominal setup for mobile geospatial Web interaction for the near future: a state of the art mobile phone is assumed as the target device, with large color screen (240 x 320 pixel resolution), UMTS connectivity, the ability to install 3rd party Java applications, and GPS receiver (either as external Bluetooth accessory or built-in). As the most common and important format for presenting geographical data, a map is used to visualize the LVis query result on the phone.
- As an example for the legacy support the framework provides, a no-frills GSM phone is used assumed as client device in the second scenario: The phone does neither feature Bluetooth, nor UMTS connectivity, nor the ability to access the Web or to install 3rd party applications in any form. On this device, SMS is the only communication channel available besides voice. As explained previously, the location is inferred from a street address or landmark name provided by the user in this case.
- Finally, the third scenario examines the practical suitability of the framework for an orientation aware interaction method: a functional implementation of a Geo-Wand is used as a test device. Since real-world browsing metaphors such as the Geo-Wand represent a primary motivation behind the framework's architecture and conceptual design, this scenario is investigated in more depth: a series of tests is carried out to provide detailed insights into the achievable performance of the Geo-Wand, and the approximate error characteristics that can be expected with a pointing-based system under real world conditions.

8.3.1 TEST SCENARIO 1 AND 2: PROCEDURE

The goal of the first two test scenarios is to illustrate how the framework interoperates with contemporary mobile phones, as they are readily available on the market today: this allows a qualitative assessment of the impact of the varying quality of the LVis on conventional types of user interfaces, i.e. text- or map-based GUIs. As explained, mass-market mobile phones were assumed as target devices in these scenarios; namely a high-end device that uses GPS positioning, and presents the query result in the form of a map image; and a low-end mobile phone that accesses the framework via the SMS interworking function, and where the location is inferred from a street address provided manually by the user in the request SMS.

Out of the twelve test locations that were previously defined for the LVis tests (compare Section 8.2), four test locations were selected for the test scenarios; with one location in each type of environment (*low-density urban*, *park*, *urban* and *urban canyon environment*). For each test location, a single sample query was produced for each of the two client device setups. As the content base for the scenarios, POI data of restaurants in the inner city of Vienna was used (compare LVis examples in Section 7.2.3). Furthermore, since no restaurant data was available in the *urban canyon* test environment, a dummy POI data set was produced manually by the author, with one POI placed in each building in the area.

8.3.2 TEST SCENARIO 1 AND 2: RESULTS

This sub-section presents the results that were produced in response to the sample queries: the original LVis delivered by the query engine is visualized in the same top-down view format used previously in Section 7.2.3 and Section 8.2, and presented alongside the resulting client user interface.

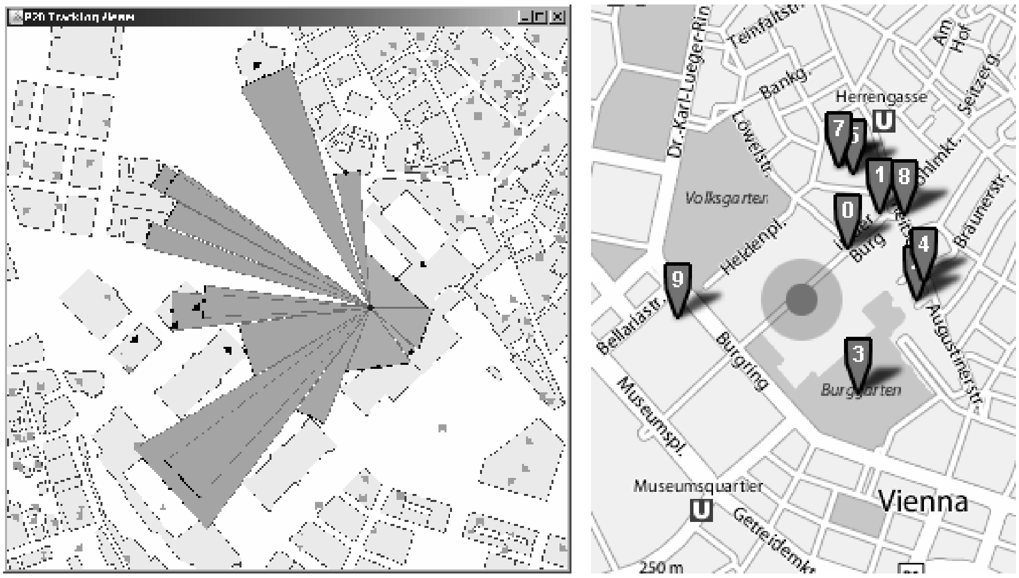
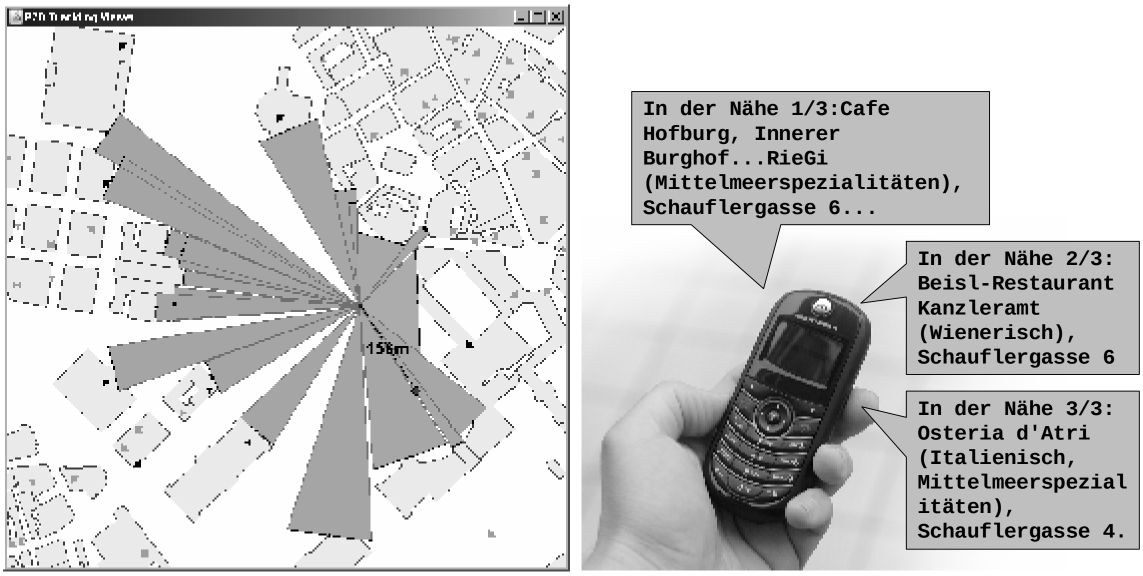


Figure 62. LVis query result and corresponding map image (Heldenplatz)

Example 1: Heldenplatz. The first example was produced on *Heldenplatz*, one of the test locations situated in *low-density urban environment.* Figure 62 depicts the LVis query result together with the corresponding 240 x 320 pixel map graphic: A dot in the center of the map represents the user's approximate location as it was estimated by the GPS. The ten nearest POI are indicated on the map with markers numbered from 0 to 9.

In a practical implementation, the client application could therefore allow the user to switch from the map to a detail information screen (e.g. presenting name, description, street address and other information contained in the LVis) by pressing the corresponding digit key on the phone keypad. As can be observed in Figure 62, the map discards the visibility information contained in the LVis: there is no visual distinction between visible and hidden POI. This design decision has been taken for the current implementation of the map module. However, a possible differentiation (e.g. through a different color or type of marker) is easily possible. In fact, it may make sense for future implementations, as HCI research suggests that hints on the visibility of geographic features may increase the usability of mobile maps (Fröhlich *et al.* 2006).



In der Nähe 1/3:Cafe Hofburg, Innerer Burghof...RieGi (Mittelmeerspezialitäten), Schauflergasse 6...

In der Nähe 2/3: Beisl-Restaurant Kanzleramt (Wienerisch), Schauflergasse 6

In der Nähe 3/3: Osteria d'Atri (Italienisch, Mittelmeerspezialitäten), Schauflergasse 4.

Figure 63. LVis query result and corresponding SMS response (Heldenplatz)

Figure 63 (left) shows the LVis that was produced in response to the SMS query issued from the low-end phone. The text “Heldenplatz, Vienna, Austria” was sent in the request SMS as location hint. As can be seen, the 3rd party geo-coding service used by the positioning support component delivers a viable result, which is about 150m offset from the users true location. The SMS response shown in Figure 63 (right) illustrates the rules by which the framework translates the LVis into short, meaningful text messages: The text starts with a generic introduction (“In der Nähe”, meaning “nearby”) and an indication of the sequence number of the message, in case the response consists of multiple SMS (e.g. three messages in the example). In each SMS, POI are listed with their name, the description (if any) in brackets, and the street address (if contained in the LVis). All remaining LVis attributes are ignored. Individual POI are separated by an ellipsis. The text for one SMS is limited to 140 characters; multiple SMS will be generated if necessary, so that the response will contain at least four POI, or so that it consists of a maximum of 3 SMS messages (whichever applies first). As in the map scenario, the SMS interworking function discards the visibility information that is contained in the LVis: a comparison of Figure 63 (left) with Figure 62 (left) clearly shows that the offset caused by the address-based positioning introduces considerable error to the LVis. While the basic set of POI included with the LVis remains similar, the visibility information is, in general, no longer valid. Unlike in the previous (map) case, an inclusion of visibility information in the SMS user interface is therefore not reasonable.

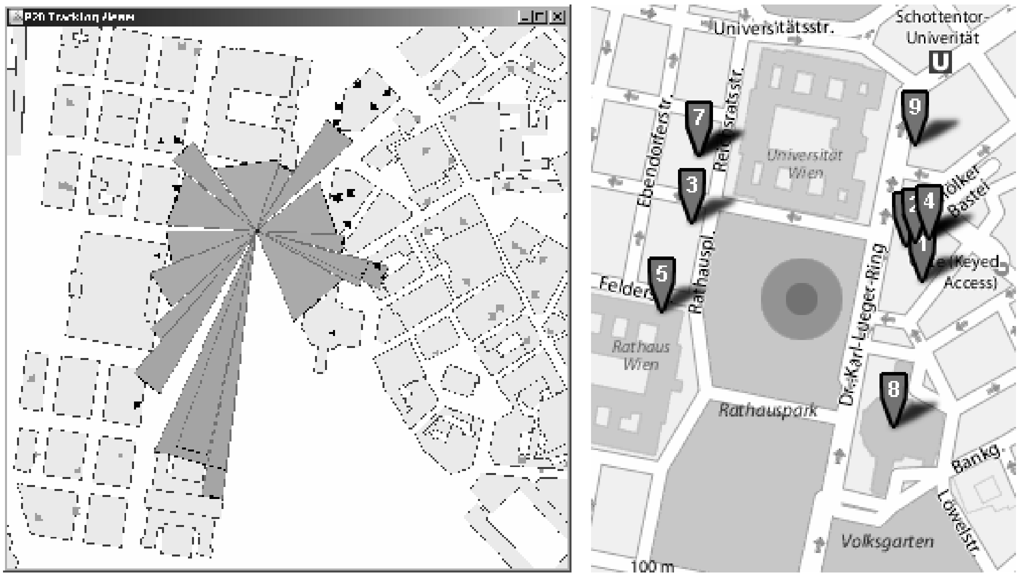


Figure 64. LVis query result and corresponding map image (Rathauspark)

Example 2: Rathauspark. The second sample query was produced with data for the *Rathauspark* location. Figure 64 shows the LVis (left) and the corresponding map image (right). The map image contains a particularly distinct example of a negative effect that is frequently observed on popular Web map applications today: symbol clutter. Due to the close proximity of some POI, their markers overlap, and their numbering is rendered (partially) unreadable. As discussed in Section 4.2.3, HCI research on how effects such as this can be avoided on small-screen mobile maps is a topic of ongoing research which is beyond the scope of this thesis. The example can, however, be seen as a confirmation that experimentation with a functional system will quickly reveal such limitations; and that the author's initial approach of designing the framework so that it can serve as a prototyping toolkit for a range of geospatial interfaces appears justified.

In der Nähe 1/2: Wiener Rathauskeller, Rathausplatz 1... Restaurant Vestibül, Dr.-Karl-Lueger-Ring 2... Sluka, Rathausplatz 8

In der Nähe 2/2: Einstein (Wienerisch), Rathausplatz 4... Cafe Landtmann (Wienerisch), Dr.-Karl-Lueger-Ring 4.

Figure 65. LVis query result and corresponding SMS response (Rathauspark)

Figure 65 shows the LVis and corresponding SMS response that was produced for the legacy scenario, using the phrase “Rathausplatz, Vienna, Austria” as location hint in the request. As in the previous example, the geo-coding result is viable; the LVis, however, is distorted. As a consequence, the order of the POI, when sorted by distance, deviates slightly from the true situation. In general, however, the response remains acceptable and meaningful nonetheless: three of the five closest venues (*Café Landtmann*, *Einstein* and *Wiener Rathauskeller*) and are all contained in the SMS response. The remaining two venues are less than 150m further away than other, more close-by venues.

Example 3: Graben. The third example was produced for the *Graben* location in the *urban environment* test area. A noteworthy effect that can be observed in Figure 66 is a slight offset between the framework’s environment model and the map graphic: This effect is also present in the other map images. However, it is most pronounced in the *urban environment* example, since this is the only one that uses the highest possible zoom level supported by the *Yahoo Map Image API* (which is used to generate the base map). The reason for the offset can not be clearly attributed to a single error source. It is actually an accumulation of various unavoidable errors: such as minimal measurement errors in the environment model used by the framework; minor distortion errors introduced by the projection used in the environment model; distortions introduced by the mismatch between the projection used to translate POI coordinates into pixel coordinates and the map’s (unknown) native projection and map scale; and, finally, minor measurement inaccuracies in the map itself.

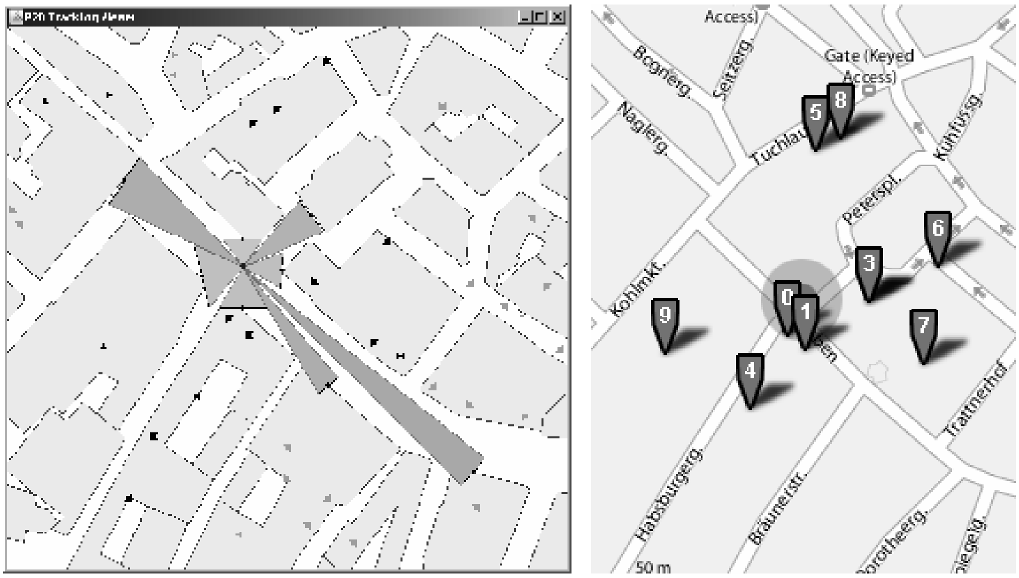


Figure 66. LVis query result and corresponding map image (Graben)

Figure 67 shows the corresponding LVis and SMS response, produced from the query “Graben, Vienna, Austria”. As expected, the positioning offset is smaller in densely developed urban terrain, where roads and squares are smaller, and location hints can be resolved with higher granularity. With regard to the set and order of contained POI, the distortion of the LVis is lower; the visibility information is, however, is still to erroneous to be included with the response.

```
In der Nähe 1/2: Eissalon
Iori, Trattnerhof 1...
Ensemble Theater Bar,
Petersplatz 1...
Weisshappel Stüberl
(Wienerisch), Petersplatz 1
```

```
In der Nähe 2/2:
Yohm Cafe, Graben
26...Chattanooga,
Graben 29...
Segafredo
(Italienisch),
Graben 29...Yohm
(Asiatisch),
Petersplatz 3.
```

Figure 67. LVis query result and corresponding SMS response (Graben)

Example 4. Carl-Auböck-Promenade. The final example shows the *Carl-Auböck-Promenade*, located in the *urban canyon* environment. The result of the GPS query is presented in Figure 68, the response to the SMS query (using "Carl-Auboeck-Promenade, Vienna, Austria" as query text) is shown in Figure 69.

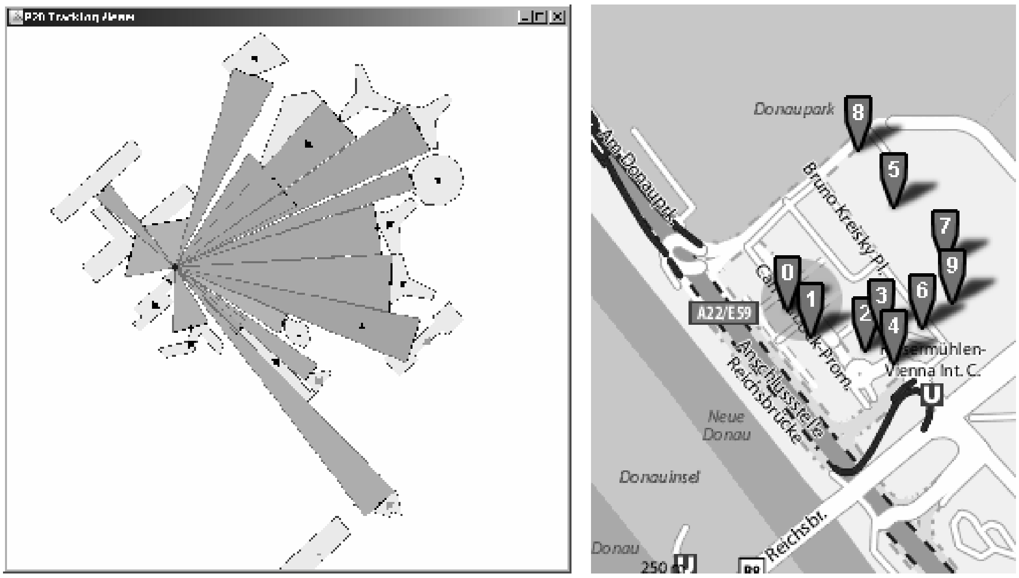
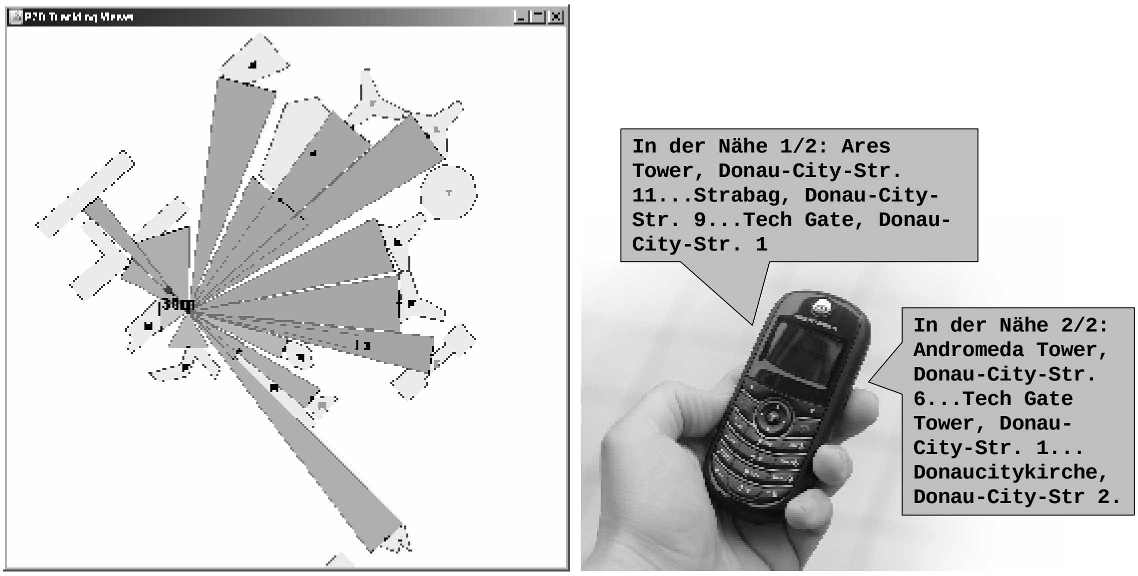


Figure 68. LVis query result and corresponding map image
(Carl-Auböck-Promenade)

As can be observed in Figure 69 (left), the offset introduced due to the address-based positioning is relatively low (approx. 35 meters). (This coincidence is due to the fact that the test location is in the middle of the *Carl-Auböck-Promenade*; and the 3rd party geo-coding service used by the positioning support component will return the middle of the road in case no house number is provided with the geo-coding request.) Because of the low offset, the SMS response corresponds almost exactly with the map: the only exception is the sixth POI (*Donaucitykirche*), which is listed as the seventh POI (i.e. POI number 6) on the map.



In der Nähe 1/2: Ares Tower, Donau-City-Str. 11...Strabag, Donau-City-Str. 9...Tech Gate, Donau-City-Str. 1

In der Nähe 2/2: Andromeda Tower, Donau-City-Str. 6...Tech Gate Tower, Donau-City-Str. 1... Donaucitykirche, Donau-City-Str 2.

Figure 69. LVis query result and corresponding SMS response (Carl-Auböck-Promenade)

In conclusion, the author claims that the above examples confirm that the LVis can serve as a viable intermediate data exchange format for conventional map- and text-based LBS user interfaces. While two of the LVis' primary properties – information about the geometry of the surrounding environment and explicit designation of the visibility of geographic features – are ignored in the presented examples, the LVis still serves as a compact encoding format for the relevant attributes of POI (such as longitude, latitude, name, description or street address) and adds a convenient polar coordinate location, which is relevant and useful for the generation of all user interfaces that include information about the distance of POI – either explicitly or implicitly, e.g. by sorting POI by increasing distance.

The above examples also provide insight into some of the limits of the LVis' suitability for conventional LBS user interfaces: For example, the `adr` attribute which provides a street address for a billboard or POI in the LVis is optional; and the current framework implementation will include an address in the LVis only if it was provided in the original content. Clearly, the SMS responses would be practically useless if no street address was available for the POI. Hence, the quality of the original content is crucial in order for the framework to produce usable output. Secondly, the usefulness of SMS based interaction would be greatly enhanced if the service was able to provide instructions on how to get to the locations listed in the response, and not just their name and address.

This confirms the need for optionally embeddable navigation information in the LVis, as outlined in Section 6.3.2.

8.3.3 TEST SCENARIO 3: PROCEDURE

The third test scenario illustrates how the framework is able to support orientation aware interaction, based on mobile devices equipped with GPS and orientation sensors. Besides exemplifying the principle of an orientation based interaction method, the test scenario also has a second important objective: As explained, the design of the framework's query component implicitly assumed error-free location estimates. The subsequent negative effects of GPS error, as well as the effects of inaccuracy introduced by legacy mechanisms such as address geo-coding, has been discussed in detail above. Error-free positioning, however, is not the only implicit assumption that has been made in the design of the framework: so far, it was also implicitly assumed that orientation detection is free of errors. The examples presented in 8.2 compared the LVis with the real FOV by superimposing it on a panorama photograph. It was ignored that an orientation aware device will, in general, experience certain heading and tilt errors as well. In addition to the errors contained in the LVis due to GPS offsets, these orientation errors may further decrease the quality of the correspondence between the user interface and the real FOV.

In order to provide insight into how the combination of GPS and orientation error affects the perceived performance of an orientation based real world browsing application, this test scenario therefore examines a functional implementation of a Geo-Wand as a concrete example. ‘Pointing samples’ were collected at each of the twelve test locations that were previously used for the LVis tests (compare Section 8.2): At each location, four to nine buildings were designated as ‘pointing targets’. During the test, the author stood stationary at the test location, and pointed the device at the designated target buildings in a pre-defined order. Each pointing gesture was confirmed by a button click, causing the system to log the current geo-coordinates measured by the GPS and the heading and pitch values measured by the orientation sensors to a file for later analysis and visualization.

The goal of this test was to get an approximate value for the ‘pointing success rate’ in different environments, i.e. the percentage of cases where the system would accurately select the building the user physically pointed at in the real world, versus the cases where the system selected a false target due to GPS and/or sensor errors. 80 pointing targets were defined in total: 17 on the *low-density urban* environment test locations, 17 on the *park environment* test locations, 23 on the *urban environment* test locations, and

23 on the *urban canyon environment* test locations. As in the case of the GPS tests (Section 8.1), the pointing tests were repeated three times, at four hour intervals.

Figure 70. Orientation aware device prototype used for the test

Figure 70 shows the prototype device that was used for the test: As explained, the prototype was produced as part of the *Point to Discover* research project (with hardware contributed by other project members). It consists of a custom-made plastic shell that snaps onto the back of a standard, mass market mobile phone. Mounted inside the shell is a digital compass and 2-axis tilt sensor module purchased from a commercial vendor. The module is connected to a standalone power supply and a Bluetooth transceiver chip that transmits the sensor measurements to the phone wirelessly using a simple, text-based protocol. The prototype does not include an integrated GPS receiver; the same Bluetooth GPS unit as in the GPS tests (again carried in the author’s shirt pocket) was therefore used in conjunction with the setup shown in Figure 70.

8.3.4 TEST SCENARIO 3: RESULTS

As explained, the results of the pointing tests were analyzed according to their ‘pointing success rate’, i.e. the cases where the Geo-Wand was able to select the correct target building (or billboard, respectively), based on the GPS position estimate and the orientation sensor measurements vs. the situations where GPS and sensor errors lead the Geo-Wand to select a wrong building as the pointing target, or no target at all. In a second analysis, *partial hits* were allowed as well: pointing queries that missed the target by not more than 5 degrees were also counted as successful hits (in this case, the result returned by the Geo-Wand is a set of targets, rather than a single target). Figure 71 shows examples for cases rated as a pointing success (Figure 71, left) and a *partial* pointing success (Figure 71, right) as encountered during the tests.

Figure 71. Geo-Wand pointing examples: successful (left) and partially successful query (right)

In total, the test yielded an overall pointing success rate of 71.25% (3 tests with 80 targets each, and a total of 171 successful pointing queries). In the case where *partial hits* were also counted as hits, the overall pointing success rate increased to 83.75% (201 successful pointing queries). Table 1 shows the detailed results for each of the three tests, with results subdivided by environment type. As can be seen, the measured pointing success rate approaches 100% in *low-density urban environment*, and is slightly decreased in the *park environment*.

Two more results are worth pointing out: First, there is a high variation between the results from three tests in *urban environment* (60.87% in test 1 vs. 30.43% in test 3). The increase of the pointing error rate with each test coincides with the decrease of GPS accuracy that was measured in the three GPS tests: since the tests were conducted approximately at the same time of day, it is reasonable to expect that satellite constellations experienced during both test series was comparable. Moreover, the LVis test results presented in Section 8.2 provide evidence that the consequences of GPS error on the LVis (and hence the distortion of the virtual FOV vs. the real-world FOV) are substantial in densely developed environment. The author therefore argues that GPS performance is more limiting to the overall performance of the Geo-Wand than compass or tilt sensor performance, which can also be observed in the error examples discussed in detail below.

Table 1. Pointing success rate by test repetition and environment categories

	Test 1		Test 2		Test 3		TOTAL	
		with partials		with partials		with partials		With partials
Low-Density Urban	88.24%	94.12%	100%	100%	100%	100%	96.08%	98.04%
Park	88.24%	100%	82.35%	94.12%	76.47%	88.24%	82.35%	94.12%
Urban	60.87%	65.22%	56.52%	91.30%	30.43%	34.78%	49.28%	63.77%
Urban Canyon	69.57%	86.96%	65.22%	82.61%	65.22%	86.96%	66.67%	85.51%

The second noteworthy effect is that the Geo-Wand performed notably better in the *urban canyon environment* than in the – seemingly less problematic – *urban environment.* The author attributes this result to two possible causes: First, the area chosen for the test – despite being the only suitable area available to the author – was still relatively open, with wide spaces in between high rise buildings (compare examples 1 and 2 in Section 8.2.5). Therefore, the area might not fully qualify as a typical *urban canyon* environment. In fact, the GPS tests confirm that GPS accuracy was not much below the *urban environment* case. As the second cause for the good performance, a closer inspection of the results reveals the following reason: Buildings in the *urban canyon environment* are considerably larger than those in the *urban environment* (e.g. office towers vs. smaller inner city buildings); i.e. the size of the pointing targets (i.e. the billboards) is increased, while GPS and compass conditions remain largely the identical.

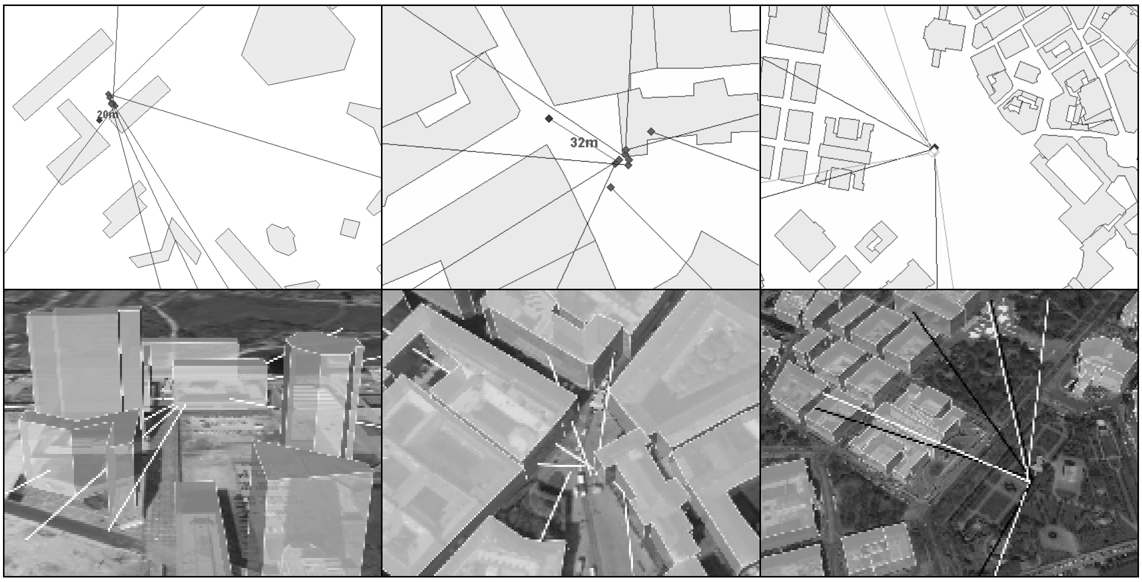


Figure 72. Geo-Wand error examples, from left to right: (a) user placed behind object due to GPS error, (b) missed alleyway due to GPS error, (c) temporary compass offset

In order to gain a more detailed insight into the types of errors that lead the Geo-Wand to decide incorrectly, a visual inspection of the pointing results was conducted, using two tools: First, the results were visualized in the framework's proprietary GUI component. Secondly, the result data was converted to 3D geometry and imported to *Google Earth* for interactive inspection. Figure 72 exemplifies three commonly observed error characteristics. Top and bottom images thereby represent the same situation, as displayed in the custom viewing tool (top) and in *Google Earth* (bottom), respectively. (Note that for the sake of discussion and easier visualization, the individual LVis results are not shown in the images. Instead, the images directly show the pointing vectors, as they were logged by the Geo-Wand.)

Figure 72a shows a situation which was recorded at the same location as the third example presented in Section 8.2.5 (*Wohnpark Donau City*): a GPS error places the user wrongly behind a building by about 20-30m. This leads to a blocked line of sight to most of the pointing targets, even though there is clear line of sight in reality. Figure 72b shows a comparable case in *urban environment* (recorded at *Freyung*, compare with the first example in Section 8.2.4): a GPS error places the user away from a street crossing by approx. 30 meters. This leads to an error when the user points towards a building down the street (see pointing line parallel to the street in S-W direction): consequently the building South of the alley is wrongly assumed to be the target, rather than a building down the alley. These frequently occurring examples reconfirm that in densely developed terrain, a comparatively low GPS error can have severe consequences on the pointing success rate and, hence, on the perceived system performance.

Figure 72c shows the case of errors caused by a temporary compass offset: samples from two tests are compared against each other, with samples from test 2 shown in white and samples from test 3 in black. While the samples from test 2 are accurate, the two Northern samples from test 3 are offset by their true direction by about 20 degrees. In the tests, such effects either occurred locally (due to nearby ferromagnetic material) or temporarily, as in the example above. The origin of the temporary effects is not yet fully clarified; more research will be needed in the future to identify their sources (such as electrical equipment operating nearby, for example) and their range reliably. The example shown above, however, was the only one the author encountered during the tests where compass error was observed with such distinctiveness; in all other samples, GPS error was by far the most dominant cause of error.

9 Discussion

This section presents related work that is relevant with regard to the results of this thesis. The related projects are briefly outlined, and it is discussed where their scope or outcome differs, agrees or complements this thesis.

9.1 Works of Faisal

Faisal (2003) investigated the use of pointing as an alternative query modality for mobile GIS. When comparing the pointing behaviour of subjects holding the pointer close to the eyes (*“telescope pointing”*) vs. with their arms outstretched, Faisal found that considerable pointing errors can occur due to the deviation between the line of sight of the user’s eye and the pointer direction. He proposed a correction factor that reduces the angular difference between the line of sight of the eyes over the tip of the pointer and the pointer direction. The factor is computed based on distances measured from the geometry of the user’s body (e.g. distances between neck, eyes, shoulder joint and pointer).

Faisal acknowledges the fact that GPS inaccuracy also forms one possible source of error for pointing-based selection. A discussion on the type or magnitude of the effect of that error on the performance of the overall system, however, is not provided. Given the results of this thesis – which clearly identify GPS inaccuracy as the dominant error source – it seems questionable whether there is need for the correction factor proposed by Faisal, at least in a mass audience system that operates with low-cost hardware. Pointing-based mobile GIS tools for professional use, on the other hand, are likely to feature higher quality positioning equipment (e.g. differential GPS). Therefore, these may benefit from algorithmic angular error correction; in particular if they are used to identify small, far away targets.

An additional observation reported by Faisal may, in fact, be more relevant with regard to this thesis: In his studies, he found that subjects consistently overshot nearby targets. An investigation whether this behaviour can reproducibly be observed with users of the Geo-Wand prototype used in this thesis (and a consequential incorporation of this circumstance into the query algorithm) may be worthwhile as future work.

9.2 Works of Gardiner and Carswell

Gardiner and Carswell (2003) introduced a directional query method in a spatial database system. Their method uses a comparable “field of view” metaphor as the spatial query engine proposed in this thesis. The thesis fully agrees with the fundamental statement of Gardiner and Carswell: that determination of the line of sight of a user is an important functionality that spatial databases need to provide in the future, in order to support location and orientation aware mobile systems.

In their system, however, Gardiner and Carswell implicitly assume error-free positioning: Though they mention spatially enabled PDAs with embedded 3D compasses and tilt sensors as potential target platform, a discussion on the impact of GPS and orientation sensor error is beyond the scope of their work. Furthermore, their query algorithm is limited to operating in two dimensions. The introduction of a 3D data model is listed as future work.

9.3 Project MARA

With its concept of modelling the geometry of the visible environment in the query result; and the consequential idea that this information can be used on the mobile phone GUI, e.g. to provide a “real world browsing” layer superimposed on the phone’s camera image (compare Section 6.1.2), this thesis touches upon the field of augmented reality research.

Augmented reality (AR) systems are mobile computing systems that supplement the real world with additional information by means of a “see-through” user interface metaphor. Based on a concept originally pioneered by Sutherland (1968), early mobile AR systems like the *Touring Machine* (Feiner *et al.* 1997) or the *Tinmith* system (Piekarski and Thomas 2001) were comprised of a tracked, head-mounted display, and computing hardware carried by the user in a backpack. Early efforts to create more lightweight and agile mobile AR systems for handheld use include *NaviCam* (Rekimoto and Nagao 1995) and the PDA-based AR system presented by Wagner and Schmalstieg (2003): Unlike the *Touring Machine* or *Tinmith*, which rely on semi-transparent displays, both of these systems superimpose the AR imagery on the live image of a camera attached to the device (“video see-through”).

A project which experimented with video see-through AR on a commercial smartphone is the *Mobile Augmented Reality Applications* (*MARA*) project conducted at Nokia Research Center (Kähäri and Murphy 2006). The specifications of the prototype sensor hardware used in *MARA* are almost identical to those of the *Point to Discover* prototype device: a GPS receiver is used for positioning; accelerometer-based tilt sensors and a tilt-compensated compass are used to detect the 3D orientation of the mobile phone; both are connected wirelessly via Bluetooth. Video demonstrations found at the project Website[129] show that the application's user interface represents nearby objects of interest as rectangular markers: This suggests that a billboard-like approach may have been followed in the *MARA* data model as well.

Furthermore, a remark on a possible future improvement of the application's accuracy through relative positioning technologies suggests that *MARA* faces similar problems with regard to GPS offset. To the best of the author's knowledge, however, information on the project details and results has never been published or made otherwise publicly available. A reliable judgment of how the *MARA* system and application infrastructure relates to the approach followed in this thesis is therefore not possible.

---

[129] http://research.nokia.com/research/projects/mara/index.html

10 Conclusion and Outlook

This thesis investigated the engineering issues that content authors, software developers and interaction designers are facing today when developing geospatial applications – i.e. applications that access, present and manipulate geo-referenced content – for mobile devices. Motivated by the recent popularity of end-user generated mapping applications on the World Wide Web, the thesis reflected on how a similar model of end-user enablement can be achieved in the domain of location based services. This section summarizes the key conclusions and results of the thesis in concise form, relates them to the three research questions defined in Section 1.2, and highlights possible directions for future research.

10.1 The Framework

The key contribution of this thesis is a generic application framework for mobile geospatial applications based on Web infrastructure. The framework simplifies the creation of these applications by addressing key engineering challenges. At the same time, it allows for varying degrees of developer involvement:

Content authors on the one hand, who are only interested in ensuring that their content is suitable for location based mobile access, will not need to be aware of the framework's internal functionality at all. The only requirements are that they provide their content in one of the formats defined in the *GeoStack*, and that it adheres to certain quality levels: e.g. the POI geo-referencing must be sufficiently accurate; POI names, labels and textural descriptions must be expressive and concise; and POI meta-information should not lack certain data without which a satisfactory user experience would not be possible (e.g. street addresses without which SMS-based services would be unusable, as discussed in Section 8.3.2).

Application developers, on the other hand, who use the framework to create a service – such as a location based treasure hunt game or community application – will likely wish to customize some of the components of the framework: e.g. configure a different visual style for the map module, or modify the rules used by the SMS interworking function to generate SMS from the LVis. To this group of developers, the framework represents a software library that shields them from some of the engineering intricacies common to mobile geospatial applications, e.g. the variations in device capabilities, as discussed in Section 4.2.1 to Section 4.2.5. They will need to be aware of the framework's overall architecture and the roles of the individual framework modules. They will, however, in general not need an understanding of the query process and the Local Visibility Model.

Finally, to developers who aim to create custom spatially enhanced client applications; or interaction designers who wish to experiment with sensor driven real-world browsing metaphors, the framework represents a prototyping aid: They will need to be aware of the framework's internal architecture, components, and in particular the concepts and elements of the Local Visibility Model. It provides them with a flexible, yet compact egocentric perspective on content and geometry in the user's immediate physical environment and can, hence, act as a common data model behind various types of geospatial user interfaces, as has been discussed in Section 6.1.2, Section 7.6 and Section 8.3.2.

10.1.1 RESEARCH QUESTION 1

The primary level of developer involvement, as discussed above, results out of the first design approach demanded in Section 4.3: harmonization with established practices and data formats on the World Wide Web. This requirement directly relates to the first research question stated in Section 1.2:

> *How can existing Web mapping tools and standards benefit the development process of mobile geospatial applications and location based services?*

The answer that the framework offers to this question is, in fact, an indirect one: Content authors can transparently re-use their data, while the framework's ego-centric query method will provide a location-based view on it. Consequently, since the framework serves as the link between location aware devices and existing Web mapping tools and standards, they indirectly benefit the development process insofar, as content authors can continue to use them, and no dedicated modifications are required for mobilizing their content.

10.1.2 RESEARCH QUESTION 2

The secondary level of developer involvement – namely the use of the framework as a server-side software library – relates to the second design approach demanded in Section 4.3: abstraction of device features and capabilities towards the application. Again, there is a direct correspondence between this requirement and one of the research question stated in Section 1.2:

> *How can the application programming interface be de-coupled from the varying technological characteristics of different mobile devices, as well as the implementation details of different positioning methods?*

In order to provide an answer to this question, the thesis proposes several measures, which are realized within the framework architecture. The central premise is a clear separation between *data model* and *presentation* of the spatial query result: as detailed in Section 6, the query result represents a geometrical abstraction of the environment around the user. Hence, it is largely independent of any particular type of user interface or interaction metaphor. As such, it can therefore be adapted towards a wide range of mobile devices with varying capabilities.

In order to achieve the required de-coupling between device-specific adaptation and the programming interface, the adaptation process must be made transparent to the application developer: for this purpose, the framework introduces the concept of a *presentation and formatting component*. The component translates the abstract query result into a device-specific representation at runtime, based on device profile information stored in a device profile database (compare Section 5.2.3). While the separation between egocentric spatial data model and presentation that is introduced by the framework is a unique contribution of this thesis, the realization of a presentation and formatting component can be achieved using state of the art tools and libraries, as has been shown in Section 7.3.

The second requirement stated in research question 2 relates to a similar de-coupling of the programming interface from the technology (or method) used to locate the user. Due to its low cost, global coverage, and independence from network infrastructure, GPS has been assumed as the preferred means of positioning for the framework presented in this thesis. A ubiquitous proliferation of GPS-enabled mobile phones, however, is unrealistic not just for the near future. Also, GPS is limited in terms of accuracy and availability, as detailed in Section 4.2.4. Therefore, the framework introduces the concept of a server-side *positioning support component*: This system module provides a single interface to multiple alternative positioning technologies beyond GPS, so that the process of positioning and the details of the underlying implementation remain transparent to the application developer. As discussed in Section 5.2.5, various such alternative options for determining the user's location exist (e.g. by matching WLAN SSIDs or GSM cell tower IDs against a geo-location database) and can be included in the *positioning support component* in principle. As a proof of concept, the *positioning support component* developed in this thesis has been implemented with one support module: an address geo-coding module that uses a third party Web service to translate from a street or landmark name provided manually by the user to a geographical coordinate. Thus, the module can act as an ultimate fall-back solution that allows inferring the location of even the most basic legacy device, as discussed in Section 7.5 and exemplified in Section 8.3.2

10.1.3 RESEARCH QUESTION 3

As explained in detail, the recent efforts of handset manufacturers to equip mobile phones with integrated orientation sensors has been a major motivating factor for the topic and scope of this thesis. It is the expectation of the author that these features will enable innovative “real-world browsing” applications for a mass audience in the near future; and that experimentation with new application scenarios and interaction styles is crucial in order to develop a better understanding of the possibilities and limitations of this next generation location based service technology. Consequently, the third research question stated in Section 1.2 is:

> *What infrastructure and tool support can be given to foster innovative interaction concepts enabled by orientation aware mobile phones, without adding to the complexity and implementation overhead of the development process.*

As with the previous two research questions, there is a direct relationship between the question and one of the design approaches stated in Section 4.3: application and user interface prototyping support (which, in turn, directly relates to the deepest level of developer involvement, as described above).

The author argues for the following answer to the third research question: *The Local Visibility Model*.

- Since the LVis provides information about nearby geographic features in three dimensions, rather than only in two, it inherently facilitates spatial interaction concepts that go beyond traditional 2D maps.
- Its ego-centric data model and the relative, polar coordinate notation considerably reduces the development effort for orientation aware user interfaces such as the Geo-Wand or the Smart Compass, which would normally require transformations from Cartesian to polar coordinate space.
- Explicit information about the visibility of geographic features reduces the preprocessing required to realize visibility-driven interaction concepts such as augmented reality user interfaces.

- The use of simple, easy to understand metaphors to model the environment (i.e. points of interest and billboards) and standard units and measurements (i.e. meters and decimal degrees) minimizes the entry barrier for application developers previously not involved with geographic applications, or who have little expertise with graphical vector formats like SVG or 3D geometry formats like VRML.
- Due to the fact that the LVis separates the data model from the presentation, it is agnostic of the type of interaction and user interface implemented on the mobile device. Introducing orientation aware browsing capabilities into an existing service or application built with the framework can therefore happen without the need for any modifications in the server domain. An explicit migration step on behalf of the content author or service developer is not required.

What the framework does not simplify is the user interface design and development process itself: Developers must still implement the client-side code that turns the *information* that is contained in the LVis into *interaction* that is facilitated by the mobile device's GPS, sensors and other interface elements like buttons, stylus or joystick. As this code will always remain specific to the type of interaction; the form and design of the user interface; and the implementation details and APIs of the device and software platform, the framework cannot provide any assistance here.

Nonetheless, the author argues that it is exactly in this respect that the fundamental philosophy behind the LVis – namely to provide one common data exchange format for all types and forms of mobile geospatial browsers – becomes most valuable: If the vision of a spatially aware browser installed on any mass market mobile phone is to become a reality eventually, mobile device manufacturers and browser vendors need to be able to rely on such an agreed data interchange standard. Without such a format, there is the danger of uncontrolled growth of proprietary geospatial browsing solutions, and the emergence of a new form of device fragmentation, where client devices equipped with different sensor sets (e.g. location and heading only vs. full 3D orientation) mandate different treatment on the server side. As a result, adoption among the Web developer community would reduce; and innovation in the field of location based services would run the risk of being stalled.

10.2 Validation Results

The validation results have highlighted the accuracy limitations of GPS-based positioning under real world conditions. Their influence on the query process was discussed, and the distortion that occurs between the LVis and the user's real field of view was illustrated. Furthermore, it was exemplified how the LVis can facilitate different types of user interaction; and in what way the distortion effects induced by positioning error re-surface in the user interface.

A key conclusion that can be drawn from the validation results is the following: orientation aware interaction is becoming feasible; albeit with a few restrictions imposed by the accuracy limits of today's technology: Contemporary low-cost GPS receivers and orientation sensors, as they are increasingly found in mass market mobile phones, deliver a level of accuracy that is quite adequate – provided that the surrounding environment is sufficiently open, and construction density is low. In typical urban environment, however, the accuracy that was observed in the validation is insufficient for commercial deployment. Some of the restrictions will become less severe as technology matures (e.g. as magnetometer compasses get replaced by gyro-compensated compasses that are less susceptible to electromagnetic interference). Other limitations, however, will remain for several years to come: as the tests have shown, seemingly small positioning errors can have large consequences in densely developed terrain, e.g. when they place the user behind a corner, or away from a street crossing. This will most likely remain a challenge for future systems using differential techniques and next generation satellite positioning systems, such as the European *Galileo* system.

10.3 Directions for Future Research

As a next step, it is crucial to develop a deeper understanding of how positioning error and the constructional situation of the environment interrelate, so that future mobile geospatial systems can be engineered to provide increased perceivable performance and accuracy to the user. As highlighted briefly in Section 8.2.6, the Local Visibility Model may well be a simple but suitable tool to study the general distortion effects that occur between the user's real field of view and the system's virtual representation of it; and to guide the development of quantitative methods and metrics that allow an a priori assessment of the potential uncertainty that is to be expected in a particular environment.

Such a priori knowledge may allow future systems to compensate for errors algorithmically: For example, in GPS-based car navigation systems and personal navigation devices it has become common to employ *map matching* techniques to reduce the negative effect of GPS inaccuracy (Scott and Drane 1994). Map matching for car navigation exploits the fact that cars are restricted to traveling on roads. Therefore, a priori knowledge of the road network geometry can be used to correct inaccurate position estimates in a relatively straight-forward manner. In a similar fashion, future error correction techniques that take into account the properties of orientation aware interaction may be able exploit a priori knowledge of the environment block model and the potential distortion that occurs between the user's field of view and the system's virtual representation of it, to identify more plausible pointing targets than those indicated by the query parameters.

Conversely, a priori knowledge on the expectable magnitude and form of error on a specific query location may not just be relevant for algorithmic error correction approaches. It may also be a suitable aid to interaction designers for designing better user interfaces which are less susceptible to error: e.g. by communicating the potential amount of uncertainty to the user rather than concealing it (for example through representations of uncertainty intervals or error probabilities); or by switching to less error critical representations in situations of high uncertainty (for example from a Geo-Wand interface to a graphical map).

The inclusion of collaborative techniques into the spatial query algorithm proposed by this thesis may represent a further interesting path for future research: Based on anonymized feedback on how often content associated with a particular billboard or point of interest has been accessed by users, the query engine may enrich the LVis with meta-information on the "popularity" of geographical features. Again, this information can in turn be used by the application to favour frequently selected targets over rarely selected ones (e.g. in a Geo-Wand interface); or allow interaction designers to include clues on the inferred relevance of certain targets in the user interface.

With regard to the overall architectural design, the framework proposed in this thesis certainly represents only one among many possible solutions. Nevertheless, the functional modules point at some of the key issues that developers of mobile geospatial applications are facing today. Future tools, frameworks and standards will need to address those in an adequate manner if the complexity gap that exists between end-user driven Web map development and LBS creation is to be bridged eventually. Also, the Local Visibility Model as it is described herein is undoubtedly too basic in terms of elements, syntax and scope to serve as the universal "HTML for geospatial browsers", as has been implied above. Yet, the fundamental need that the LVis aims to emphasize and address does exist: the need for a universally accepted and supported interchange format for local geospatial data; a format that encodes all the information that is needed to provide users with intuitive browsing interfaces on various types of location-, heading- or 3D orientation aware mobile phones. The author therefore argues that, above anything else, it is necessary to develop the conceptual model and thinking behind the LVis further in the future: in order to promote and advance the vision of a ubiquitous, Geospatial Web; to evolve the means by which mobile technologies can help people to experience and interact with their physical environment; and to turn geo-information into a useful, integrated and intuitive addition to daily life.

Bibliography

M. Agrawala and C. Stolte (2000) A Design and Implementation for Effective Computer-Generated Route Maps, *AAAI Symposium on Smart Graphics*, Stanford, CA, USA, AAAI Press.

S. Ahern, D. Eckles, N. Good, S. King, M. Naaman, and R. Nair (2007) Over-exposed? Privacy Patterns and Considerations in Online and Mobile Photo Sharing, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*, San Jose, California, USA, pp. 357-366, ACM Press.

D. N. Aloi, M. Alsliety, and D. M. Akos (2007) A Methodology for the Evaluation of a GPS Receiver Performance in Telematics Applications, *IEEE Transactions on Instrumentations and Measurement*, vol. 56, issue 1, IEEE.

J. de la Beaujardiere (Ed.) (2004) *OGC Web Map Service Interface*, Open GIS Consortium Inc.

M. Bell, M. Chalmers, L. Barkhuus, M. Hall, S. Sherwood, P. Tennent, B. Brown, D. Rowland, S. Benford, A. Hampshire, and M. Capra (2006) Interweaving Mobile Games with Everyday Life, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*, Montréal, Québec, Canada, ACM Press.

S. Benford, R. Anastasi, M. Flintham, A. Drozd, A. Crabtree, C. Greenhalgh, N. Tandavanitj, M. Adams, and J. Row-Farr (2003) Coping with Uncertainty in a Location-Based Game, *IEEE Pervasive Computing*, vol. 2 (3), pp. 34-41, IEEE.

A. R. Beresford and F. Stajano (2003) Location Privacy in Pervasive Computing, *IEEE Pervasive Computing*, vol. 2 (1), pp. 46-55, IEEE.

G. Borriello, M. Chalmers, A. LaMarca, and P. Nixon (2005) Delivering Real-World Ubiquitous Location Systems, *Communications of the ACM*, 48 (3), pp. 36-41, ACM Press.

S. Brewster (2002) Overcoming the Lack of Screen Space on Mobile Computers, *Personal and Ubiquitous Computing*, 6 (3), pp 188-205.

A. J. Brimicombe (2002) GIS – Where are the frontiers now? *Proceedings GIS 2002*, Bahrain, pp. 33-45.

S. Burigat and L. Chittaro (2005) Location-aware Visualization of VRML Models in GPS-based Mobile Guides, 10th International Conference on 3D Web Technology (Web3D '05), Bangor, United Kingdom, pp. 57-64, ACM Press.

S. Burigat and L. Chittaro (2007) Geographic Data Visualization on Mobile Devices for User's Navigation and Decision Support Activities, *Spatial Data on the Web – Modeling and Management*, A. Belussi, B. Catania, E. Clementini, and E. Ferrari (Eds), pp. 261-284, Springer.

S. Burigat, L. Chittaro, and S. Gabrielli (2006) Visualizing Locations of Off-Screen Objects on Mobile Devices: A Comparative Evaluation of Three Approaches, *8th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2006)*, Espoo, Finland, pp. 239-246, ACM Press.

J. Burrell and G. K. Gay (2001) Collectively Defining Context in a Mobile, Networked Computing Environment, *CHI '01 Extended Abstracts on Human Factors in Computing Systems (CHI '01)*, Seattle, Washington, USA, pp. 231-232, ACM Press.

M. Y. Chen, T. Sohn, D. Chmelev, D. Haehnel, J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. Smith, and A. Varshavsky (2006) Practical Metropolitan-Scale Positioning for GSM Phones, *8th International Confernce on Ubiquitous Computing (Ubicomp 2006)*, Orange County, California, USA, pp. 225-242, Springer LNCS.

A. K. L. Cheung (2006) Representational Issues in Interactive Wayfinding Systems: Navigating the Auckland University, *Web and Wireless Geographical Information Systems – 6th International Symposium (W2GIS 2006)*, Hong Kong, China, pp. 90-101, Springer LNCS.

J. Conallen (2002) *Building Web Applications with UML, 2nd Edition*, Addison-Wesley.

S. Consolvo, I. E. Smith, T. Matthews, A. LaMarca, J. Tabert, and P. Powledge (2005) Location Disclosure to Social Relations: Why, When, & What People Want to Share, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2005)*, Portland, Oregon, USA, pp. 81-90, ACM Press.

N. Davies, K. Cheverst, K. Mitchell, and A. Efrat (2001) Using and Determining Location in a Context-Sensitive Tour Guide, *Computer*, vol. 34, no. 8, pp. 35-41, IEEE.

J. Dillemuth (2005) Map Design Evaluation for Mobile Display, *Cartography and Geographic Information Science,* 32 (4), pp. 285-301.

G. M. Djuknic and R. E. Richton (2001) Geolocation and Assisted GPS, *Computer*, vol. 34, no. 2, pp. 123-125, IEEE.

J. E. Dobson and P. F. Fisher (2003) Geoslavery, *Technology and Society Magazine*, vol.22, no.1, pp. 47-52, IEEE.

C. Drane, M. Macnaughtan, and C. Scott (1998) Positioning GSM Telephones, *IEEE Communications Magazine*, vol. 36, issue 4, pp. 46-54, 59, IEEE.

M. J. Egenhofer (1999) Spatial Information Appliances: A Next Generation of Geographic Information Systems, *1st Brazilian Workshop on GeoInformatics*, Caminas, Brazil.

M. J. Egenhofer and W. Kuhn (1998) Beyond Desktop GIS, *Proceedings of GIS Planet 98*, Lisbon, Portugal.

J. Elson, J. Howell, and J. R. Douceur (2007) MapCruncher: Integrating the World's Geographic Information, *Operating Systems Review* 41 (2), pp. 50-59, ACM Press.

F. Espinoza, P. Persson, A. Sandin, H. Nyström, E. Cacciatore, and M. Bylund (2001) GeoNotes: Social and Navigational Aspects of Location-Based Information Systems, *3rd International Confernce on Ubiquitous Computing (Ubicomp 2001)*, Atlanta, Georgia, USA, pp. 2-17, Springer.

D. Fairbairn, G. Andrienko, N. Andrienko, G. Buziek, and J. Dykes (2001) Representation and its Relationship with Cartographic Visualization, *Cartography and Geographic Information Science*, vol. 28, no. 1, pp. 13-28.

F. Faisal (2003) *Query-by-Pointing: Algorithms and Pointing Error Compensation*, M.S. thesis in Spatial Information Science and Engineering, University of Maine.

S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster (1997) A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment, *1st International Symposium on Wearable Computers (ISWC 1997)*, Cambridge, Massachusetts, USA, pp. 74-81, IEEE Computer Society.

M. Florins and J. Vanderdonckt (2004) Graceful Degradation of User Interfaces as a Design Method for Multiplatform Systems, Proceedings of the 9th International Conference on Intelligent User Interfaces (IUI '04), Funchal, Madeira, Portugal, pp. 140-147, ACM Press.

C. E. Frank (2003) *An Egocentric Spatial Data Model for Intelligent Mobile Geographic Information Systems*, M.S. thesis in Spatial Information Science and Engineering, University of Maine.

P. Fröhlich, R. Simon, L. Baillie, and H. Anegg (2006) Comparing Conceptual Designs for Mobile Access to Geo-Spatial Information, *8th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2006)*, Espoo, Finland, ACM Press.

P. Fröhlich, R. Simon, L. Baillie, J. Roberts, R. Murray-Smith, M. Jones, and R. Nair (2007) Mobile Spatial Interaction Whitepaper, 1st Workshop on Mobile Spatial Interaction (http://msi.ftw.at).

K. Gardiner and J. D. Carswell (2003) Viewer-Based Directional Querying for Mobile Applications, *4th International Conference on Web Information Systems Engineering Workshops (WISEW'03)*, pp. 83-91, IEEE.

G. Ghinita, P. Kalnis, and S. Skiadopoulos (2007) PRIVÉ: Anonymous Location-Based Queries in Distributed Mobile Systems, *16th International Conference on World Wide Web (WWW 2007)*, Banff, Alberta, Canada, pp. 371-380, ACM Press.

R. H. Güting (1994) An Introduction to Spatial Database Systems, *The International Journal on Very Large Databases*, vol. 3, issue 4, pp. 357-399, Springer.

C. Gutwin and C. Fedak (2004) Interacting with Big Interfaces on Small Screens: a Comparison of Fisheye, Zoom, and Panning Techniques, *Proceedings of Graphics Interface 2004*, London, Ontario, Canada, pp. 145-152, ACM International Conference Proceeding Series, vol. 62, Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada.

S. Holland, D. R. Morse, and H. Gedenryd (2002) AudioGPS: Spatial audio navigation with a minimal attention interface, *Personal and Ubiquitous Computing*, 6 (4), pp. 253-259.

J. I. Hong and J. A. Landay (2004) An Architecture for Privacy-Sensitive Ubiquitous Computing*, Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys '04)*, Boston, Massachusetts, USA, pp. 177-189, ACM Press.

J. Hu, S. You, U. Neumann, and K. K. Park (2004) Building Modeling from LIDAR and Aerial Imagery, ASPRS'04, Denver, Colorado, USA.

E. Kaasinen (2003) User needs for location-aware mobile services, *Personal and Ubiquitous Computing*, vol. 7, issue 1, pp. 70-79, ACM Press.

M. Kähäri and D. J. Murphy (2006) MARA – Sensor Based Augmented Reality System for Mobile Imaging Device, Demo at the *5th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 06)*, Santa Barbara, California, USA.

R. Khare (2006) Microformats: The Next (Small) Thing on the Semantic Web? *IEEE Internet Computing*, vol. 10, issue 1, pp. 68-75, IEEE.

J. Kjeldskov and C. Graham (2003) A Review of Mobile HCI Research Methods, *5th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2003),* Udine, Italy, pp. 317-335, Springer LNCS.

S. Kristoffersen and F. Ljungberg (1999) "Making Place" to Make IT Work: Empirical Explorations of HCI for Mobile CSCW, *International ACM SIGGROUP Conference on Supporting Group Work (GROUP'99)*, Phoenix, Arizona, USA, ACM Press.

A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit (2005) Place Lab: Device Positioning Using Radio Beacons in the Wild, *3rd International Conference on Pervasive Computing (Pervasive 2005)*, Munich Germany, Springer Lecture Notes in Computer Science.

A. Leonhardi, U. Kubach, K. Rothermel, and A. Fritz (1999) Virtual Information Towers – A Metaphor for Intuitive, Location-Aware Information Access in a Mobile Environment, *3rd International Symposium on Wearable Computers (ISWC'99)*, San Francisco, California, USA, pp. 15-20, IEEE.

S. Long, R. Kooper, G. D. Abowd, and C. G. Atkeson (1996) Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study, *Proceedings of the 2nd Annual International Conference on Mobile Computing and Networking (MobiCom '96)*, pp. 97-107, Rye, New York, ACM Press.

B. MacKay, D. Dearman, K. Inkpen, and C. Watters (2005) Walk 'n Scroll: A Comparison of Software-based Navigation Techniques for Different Levels of Mobility, *7th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '05)*, Salzburg, Austria, pp. 183-190, ACM Press.

S. Maierhofer, R. Simon, and R. F. Tobler (2007) Simplified Guided Visibility Sampling for Location Based Services, *REAL CORP 2007*, Vienna, Austria.

M. Matosevic, Z. Salcic, and S. Berber (2006) A Comparison of Accuracy Using a GPS and a Low-Cost DGPS, *IEEE Transactions on Instrumentation and Measurement*, vol. 55, issue 5, pp. 1677-1683, IEEE.

T. Mitchell (2005) *Web Mapping Illustrated*, O'Reilly Media, Inc., Sebastopol, USA.

K. Mitchell, D. McCaffery, G. Metaxas, J. Finney, S. Schmid, and A. Scott (2003) Six in the City: Introducing Real Tournament – A Mobile IPv6 Based Context-Aware Multiplayer Game, *Proceedings of the 2nd Workshop on Network and System Support for Games (NetGames '03)*, Redwood City, California, USA, pp. 91-100, ACM Press.

Y. Miyazaki and T. Kamiya (2006) Pedestrian Navigation System for Mobile Phones Using Panoramic Landscape Images, *International Symposium on Applications and the Internet*, Phoenix, AZ, USA, pp. 102-108, IEEE Computer Society.

S. Mizobuchi, M. Chignell, and D. Newton (2005) Mobile Text Entry: Relationship Between Walking Speed and Text Input Task Difficulty, *7th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '05)*, Salzburg, Austria, pp. 122-128, ACM Press.

A. Mueller, T. Mundt, and W. Lindner (2001) Using XML to Semi-automatically Derive User Interfaces, *Proceedings of the 2nd International Workshop on User Interfaces to Data Intensive Systems (UIDIS'01)*, pp. 91-95, IEEE Computer Society.

B. Myers, S. E. Hudson, and R. Pausch (2000) Past, Present and Future of User Interface Software Tools, *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 7, issue 1, pp. 3-28, ACM Press.

S. Narayanaswamy, S. Seshan, E. Amir, E. Brewer, R. W. Brodersen, F. Burghardt, A. Burstein, Yuan-Chi Chang, A. Fox, J. M. Gilbert, R. Han, R. H. Katz, A. C. Long, D. G. Messerschmitt, and J. M. Rabaey (1996) Application and Network Support for InfoPad, *IEEE Personal Communications*, vol. 3, issue 2, pp 4-17, IEEE.

R. Nevatia and K. Price (2002) Automatic and Interactive Modeling of Buildings in Urban Environments from Aerial Images, *Proceedings of the 2002 International Conference on Image Processing (ICIP'02)*, Rochester, USA, vol. 3, pp. 525-528, IEEE.

J. Pascoe (1997) The Stick-e Note Architecture: Extending the Interface Beyond the User, *2nd International Conference on Intelligent User Interfaces (IUI '97)*, Orlando, Florida, United States, pp. 261-264, ACM Press.

J. Pascoe, N. Ryan, and D. Morse (2000) Using While Moving: HCI Issues in Fieldwork Environments, *ACM Transactions on Computer Human Interaction*, vol. 7, pp. 417-437, ACM Press.

G. Percivall (Ed.) (2003) *OGC Reference Model*, Open Geospatial Consortium Inc.

P. Persson, F. Espinoza, P. Fagerberg, A. Sandin, and R. Cöster (2002) GeoNotes: A Location-based Information System for Public Spaces, *Designing Information Spaces: The Social Navigation Approach*, K. Höök, D. Benyon, and A. J. Munro (Eds.), pp. 151-173, Springer.

W. Piekarski and B. H. Thomas (2001) Tinmith-Metro: New Outdoor Techniques for Creating City Models with an Augmented Reality Wearable Computer, *5th International Symposium on Wearable Computers (ISWC 2001)*, Zurich, Switzerland, pp. 31-38, IEEE Computer Society.

C. Portele (Ed.) (2007) *OpenGIS Geography Markup Language (GML) Encoding Standard*, Open Geospatial Consortium Inc.

G. Pospischil, M. Umlauft, and E. Michlmayr (2002) Designing LoL@, a Mobile Tourist Guide for UMTS, *4th International Symposium on Human Computer Interaction with Mobile Devices (Mobile HCI 2002),* Pisa, Italy, pp. 140-154, Springer LNCS.

N. B. Priyantha, A. Chakraborty, and H. Balakrishnan (2000) The Cricket Location-Support System, Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom '00), Boston, Massachusetts, USA, pp. 32-43, ACM Press

A. R. Puerta, H. Eriksson, J. H. Gennari, and M. A. Musen (1994) Model-Based Automated Generation of User Interfaces, *Proceedings of the 12th National Conference on Artificial Intelligence*, Seattle, Washington, USA, pp. 471-477, American Association for Artificial Intelligence.

I. Rakkolainen and T. Vainio (2001) A 3D City Info for Mobile Users, Computers & Graphics, vol. 25, no. 4, pp. 619-625, Elsevier.

C. Randell, C. Djiallis, and H. Muller (2003) Personal Position Measurement Using Dead Reckoning, *7th IEEE International Symposium on Wearable Computers (ISWC 2003)*, New York, USA, pp. 166-173, IEEE.

M. Rantanen, A. Oulasvirta, J. Blom, S. Tiitta, and M. Mäntylä (2004) InfoRadar: Group and Public Messaging in the Mobile Context, *3rd Nordic Conference on Human-Computer Interaction (NordiCHI 2004)*, Tampere, Finland, pp. 131-140, ACM Press.

B. Rao and L. Minakakis (2003) Evolution of Mobile Location-based Services, *Communications of the ACM*, vol. 46, issue 12, pp. 61-65, ACM Press.

J. Raper (2007a) Design Constraints on Operational LBS, *Location Based Services and TeleCartography*, G. Gartner, W. Cartwright, and M. P. Peterson (Eds.), pp. 13-25, Springer Lecture Notes in Geoinformation and Cartography.

J. Raper (2007b) Geographic Relevance, *Journal of Documentation*, vol. 63, issue 6, pp. 836-852, Emerald Group Publishing Ltd.

J. Raper, G. Gartner, H. Karimi, and C. Rizos (2007) A Critical Evaluation of Location Based Services and Their Potential, *Journal of Location Based Services*, vol. 1, issue 1, pp. 5-45, Taylor & Francis.

O. Rashid, I. Mullins, P. Coulton, and R. Edwards (2006) Extending Cyberspace: Location Based Games Using Cellular Phones, Computers and Entertainment, vol. 4, no. 1, ACM Press.

T. Reichenbacher (2007) The Concept of Relevance in Mobile Maps, *Location Based Services and TeleCartography*, G. Gartner, W. Cartwright, and M. P. Peterson (Eds.), pp. 231-246, Springer Lecture Notes in Geoinformation and Cartography.

J. Rekimoto and K. Nagao (1995) The World through the Computer: Computer Augmented Interaction with Real World Environments, *Proceedings of the 8th Annual ACM Symposium on User interface and Software Technology (UIST '95)*, Pittsburgh, Pennsylvania, USA, pp. 29-36, ACM Press.

L. J. Rouse, S. J. Bergeron, and T. M. Harris (2007) Participating in the Geospatial Web: Collaborative Mapping, Social Networks and Participatory GIS, *The Geospatial Web: How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society*, A. Scharl and K. Tochtermann (Eds.), Advanced Information and Knowledge Processing Series, Springer, London.

K. Schreiner (2007) Where We At? Mobile Phones Bring GPS to the Masses, *IEEE Computer Graphics and Applications*, vol. 27, issue 3, pp. 6-11, IEEE.

A. Scott and C. R. Drane (1994) Increased Accuracy of Motor Vehicle Position Estimation by Utilising Map Data, Vehicle Dynamics, and other Information Sources, *Proceedings of the 1994 Vehicle Navigation and Information Systems Conference*, Yokohama, Japan, pp. 585-590, IEEE.

R. Simon, F. Wegscheider, and K. Tolar (2005) Tool-Supported Single Authoring for Device Independence and Multimodality, *7th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '05)*, Salzburg, Austria, pp. 91-98, ACM Press.

R. Simon (2006) The Creative Histories Mobile Explorer – Implementing a 3D Multimedia Tourist Guide for Mass-Market Mobile Phones, *Proceedings of the 1st EVA Vienna Conference*, Vienna, Austria, pp. 59-66.

R. Simon, P. Fröhlich, and H. Anegg (2006) Beyond Location Based – The Spatially Aware Mobile Phone, *Proceedings of the 6th International Symposium on Web and Wireless Geographical Information Systems (W2GIS 2006)*, Hong Kong, China, Springer LNCS, vol. 4295/2006, pp 12-21.

R. Simon and P. Fröhlich (2007) A Mobile Application Framework for the Geospatial Web, *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*, Banff, Alberta, Canada, pp. 381-390, ACM Press.

R. Simon, H. Kunczier, and H. Anegg (2007a) Towards Orientation-Aware Location Based Mobile Services, *Location Based Services and TeleCartography*, G. Gartner, W. Cartwright, and M. P. Peterson (Eds.), pp. 279-290, Springer Lecture Notes in Geoinformation and Cartography.

R. Simon, P. Fröhlich, and H. Anegg (2007b) Enabling Spatially Aware Mobile Applications, *Transactions in GIS*, vol. 11, issue 5, pp. 783-794, Blackwell Publishing.

Souchon and J. Vanderdonckt (2003) A Review of XML-Compliant User Interface Description Languages, *10th International Conference on Design, Specification, and Verification of Interactive Systems (DSV-IS 2003)*, Madeira, Springer LNCS, vol. 2844, pp. 377-391, Springer.

S. Strachan, P. Eslambolchilar, R. Murray-Smith, S. Hughes, and S. O'Modhrain (2005) gpsTunes: Controlling Navigation via Audio Feedback, *7th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '05)*, Salzburg, Austria, pp. 275-278, ACM Press.

I. E. Sutherland (1968) A Head-Mounted Three-Dimensional Display, *Proceedings of the AFIPS Fall Joint Computer Conference*, Washington, D.C., USA, pp. 757-764, Thompson Books.

I. E. Sutherland, R. F. Sproull, and R. A. Schumacker (1974) A Characterization of Ten Hidden-Surface Algorithms, *ACM Computing Surveys*, vol. 6, issue 1, pp. 1-55, ACM Press.

T. Tezuka, T. Kurashima, and K. Tanaka (2006) Toward Tighter Integration of Web Search with a Geographic Information System, *15th International World Wide Web Conference (WWW 2006)*, Edinburgh, Scotland, pp. 277-286, ACM Press.

C. Traynor and M. G. Williams (1997) A Study of End-User Programming for Geographic Information Systems, *Papers Presented At the Seventh Workshop on Empirical Studies of Programmers*, Alexandria, Virginia, USA, ACM Press.

H. Trætteberg, P. J. Molina, and N. J. Nunes (2004) Making Model-Based UI Design Practical: Usable and Open Methods and Tools, *Proceedings of the 9th International Conference on Intelligent User Interfaces (IUI '04)*, Funchal, Madeira, Portugal, pp. 376-377, ACM Press.

S. Trewin, G. Zimmermann, and G. Vanderheiden (2003) Abstract User Interface Representations: How Well do they Support Universal Access? *Proceedings of the 2003 Conference on Universal Usability (CUU '03)*, Vancouver, British Columbia, Canada, pp. 77-84, ACM Press.

A. Turner (2006) *Introduction to Neogeography*, O'Reilly Short Cuts, O'Reilly.

M. Uhlirz (2007) A Market and User View on LBS, *Location Based Services and Tele-Cartography*, G. Gartner, W. Cartwright, and M. P. Peterson (Eds.), pp. 47-58, Springer Lecture Notes in Geoinformation and Cartography.

K. Virrantaus, J. Markkula, A. Garmash, V. Terziyan, J. Veijalainen, A. Katanosov, and H. Tirri (2001) Developing GIS-Supported Location-Based Services, *Second International Conference on Web Information Systems Engineering (WISE 2001)*, Kyoto, Japan, vol. 2, pp. 66-75, IEEE.

P. A. Vretanos (Ed.) (2005) *Web Feature Service Implementation Specification*, Open Geospatial Consortium Inc.

D. Wagner and D. Schmalstieg (2003) First Steps Towards Handheld Augmented Reality, *7th International Symposium on Wearable Computers (ISWC 2003)*, White Plains, New York, USA, pp. 127-135, IEEE Computer Society.

R. Want, B. N. Schilit, N. I. Adams, R. Gold, K. Petersen, D. Goldberg, J. R. Ellis, and M. Weiser (1995) An Overview of the PARCTAB Ubiquitous Computing Experiment, *IEEE Personal Communications*, vol. 2, issue 6, pp 28-43, IEEE.

R. Wasinger, C. Stahl, and A. Krüger (2003) M3I in a Pedestrian Navigation & Exploration System, *5th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2003)*, Udine, Italy, pp. 481-485, Springer LNCS.

M. Weiser (1991) The Computer for the 21st Century, *Scientific American*, vol. 265, no. 3, pp. 94-104.

J. Williamson, S. Strachan, and R. Murray-Smith (2006) It's a Long Way to Monte Carlo: Probabilistic Display in GPS Navigation, *8th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2006)*, Espoo, Finland, pp. 89-96, ACM Press.

T. A. Wikle (1991) Geographic Information Systems: Digital Technology with a Future? *SIGCAS Computers and Society*, vol. 21, issue 2-4, pp. 28-32, ACM Press.

D. Zhang and B. Adipat (2005) Challenges, Methodologies, and Issues in the Usability Testing of Mobile Applications, International Journal of Human-Computer Interaction, vol. 18, no. 3, pp. 293-308, Lawrence Erlbaum Associates.

Appendix A: Lvis XML Code Sample

```
<?xml version="1.0" encoding="UTF-8"?>

<lvis lng="16.35869286203369" lat="48.215725663680175">
  <billboard id="object2d.20146" lng="16.358830543740375"
       lat="48.21563204453073" alt="0.0" d="13" hdg="130"
       elev="36" hwidth="80" vwidth="72"/>

  <billboard id="object2d.20666" lng="16.358181337178753"
       lat="48.216166081454446" alt="0.0" d="62" hdg="320"
       elev="10" hwidth="38" vwidth="21"/>

  <billboard id="object2d.20424" lng="16.358987918380407"
       lat="48.21628069881812" alt="0.0" d="66" hdg="17"
       elev="11" hwidth="68" vwidth="22">
   <poi name="Bar Restaurant-Cafe Roth" descr="Wienerisch"
      href="http://cobra.ftw.tuwien.ac.at:11371/01110_1.jpg"
      lng="16.359613590704612" lat="48.21642777810011"
      alt="0.0" d="103" hdg="38" elev="0"/>
   <poi name="Charlie Ps"
      href="http://cobra.ftw.tuwien.ac.at:11371/03570_1.jpg"
      lng="16.359468770960454" lat="48.21659870327145"
      alt="0.0" d="113" hdg="27" elev="0"/>
   <poi name="Regenbogen Stube" descr="Vegetarisch"
      lng="16.358757535144168" lat="48.216832719336274"
      alt="0.0" d="124" hdg="0" elev="0"/>
   <poi name="Cafe Hobby"
      lng="16.359179106843705" lat="48.216895581736104"
      alt="0.0" d="135" hdg="13" elev="0"/>
   <poi name="Samolis" descr="Vegetarisch"
      lng="16.35903425837238" lat="48.21701254077252"
      alt="0.0" d="145" hdg="7" elev="0"/>
  </billboard>

  <billboard id="object2d.20836" lng="16.357788681684763"
       lat="48.21574216541468" alt="0.0" d="67" hdg="270"
       elev="9" hwidth="37" vwidth="18"/>

  <billboard id="object2d.20431" lng="16.35790675651319"
       lat="48.215182002777176" alt="0.0" d="83" hdg="222"
       elev="9" hwidth="58" vwidth="18">
   <poi name="Cafeteria Maximilian" descr="Italienisch"
      lng="16.35778176278234" lat="48.21474626990427"
      alt="0.0" d="128" hdg="209" elev="0"/>
   <poi name="Wachauer Weinstube"
      href="http://cobra.ftw.tuwien.ac.at:11371/00572_1.jpg"
```

```
        lng="16.357505141343808" lat="48.214755324790126"
        alt="0.0" d="139" hdg="217" elev="0"/>
    </billboard>

    <billboard id="object2d.20994" lng="16.360471575061514"
         lat="48.216267101747135" alt="0.0" d="145" hdg="62"
         elev="3" hwidth="21" vwidth="7"/>

    <billboard id="object2d.20467" lng="16.36091833126294"
         lat="48.21587140876519" alt="0.0" d="166" hdg="81"
         elev="4" hwidth="16" vwidth="9">
     <poi name="Cafe Votivpark" descr="Wienerisch"
        href="http://cobra.ftw.tuwien.ac.at:11371/03145_1.jpg"
        lng="16.361799981883504" lat="48.21579764026694"
        alt="0.0" d="230" hdg="85" elev="0"/>
    </billboard>

    <billboard id="object2d.20426" lng="16.3565811789584"
         lat="48.21646347561436" alt="0.0" d="177" hdg="295"
         elev="4" hwidth="6" vwidth="9"/>

    <billboard id="object2d.20435" lng="16.358668700769293"
         lat="48.21406880956057" alt="0.0" d="183" hdg="178"
         elev="4" hwidth="15" vwidth="8">
     <poi name="Zwillings Gw&ouml;lb" descr="Wienerisch"
        lng="16.35872982837104" lat="48.213981544440244"
        alt="0.0" d="193" hdg="177" elev="0"/>
     <poi name="Cafeteria Ristorante Votiv"
        href="http://cobra.ftw.tuwien.ac.at:11371/03144_1.jpg"
        lng="16.359019574628558" lat="48.2138825408113"
        alt="0.0" d="205" hdg="171" elev="0"/>
    </billboard>
  </lvis>
```

Appendix B: SMS Interworking Function Mailet

```
public class HWMailet extends GenericMailet {

 // The URL to the twitter message sending API
 private static final String DIRECT_MESSAGE_URL =
            "http://twitter.com/direct_messages/new.xml";

 public void init(MailetConfig config) throws MessagingException {
  super.init(config);
 }

 /**
 /* Main Mailet service method.
 /* @param mail the e-mail message passed from the mail server
  */
 public void service(Mail mail) {
  try {
   if (mail.getMessage().getSubject().indexOf("Direct") > -1) {

    // Identified as a direct message (e.g. SMS) from twitter
    Object content = mail.getMessage().getContent();
    if (content instanceof String) {

     // Extract the user name of the sending user
     String msg = (String) content;
     int idx = msg.indexOf("http://twitter.com/");
     if (idx > -1) {
      idx += 19;
      String usrNme = msg.substring(idx, msg.indexOf("\n",idx));

      // 1. Extract street address/landmark name
      // 2. Geocoding (Positioning Support Component)
      // 3. Compute LVis (Spatial Query Component)
      // 4. Translate to text
      // (not shown in code sample)
      String result = computeResult(msg);

      // Compile twitter API call HTTP request
      HttpClient h = new HttpClient();
      h.getState().setCredentials(
          new AuthScope(null, 80, null),
          new UsernamePasswordCredentials("username", "password")
                              );
```

```
      PostMethod method = new PostMethod(DIRECT_MESSAGE_URL);
      method.setDoAuthentication(true);
      method.addParameter("user", usrNme);
      method.addParameter("text", result);

      try {
       h.executeMethod(method);
      } catch (Exception e) {
       // HTTP error when calling twitter API (log error)
      }
     } else {
      // Error extracting name of sending user – malformed
      // twitter message? (log error)
     }
    } else {
     // E-mail content not a String – malformed twitter
     // message? (log error)
    }
   }
  } catch (Exception e) {
   // Log error
  }
 }
}
```

Biography

Rainer Simon was born in Linz, Austria, in 1976. He received a degree in electrical engineering (with distinction) from the Vienna University of Technology in 2002. From 1998 to 2000 he worked as a freelance multimedia designer for the university's media centre. Between 2003 and 2007, he held the position of a researcher and project manager at the *Telecommunications Research Center Vienna*, where he participated in and headed several application-oriented research projects in the fields of mobile multimedia services and mobile geospatial applications. His research interests include Ubiquitous and Pervasive Computing, Web engineering, geoinformatics, geo-visualization and mobile multimedia communication technologies.

Rainer has published and spoken at major international conferences such as MobileHCI 2005 and WWW 2007. He was awarded the Best Paper award at the 6th Symposium on Web and Wireless Geographical Information Systems (W2GIS 2006) and the Best Poster Award at the WWW 2004 conference. He co-organized a workshop and special journal issue on 'Mobile Spatial Interaction'. Presently, Rainer is employed at the Austrian mobile network operator ***mobilkom Austria*** as a solutions architect and systems designer for call center automation and mobile voice and video service infrastructure.