Johannes Kepler Universität Linz

Institut für Bioinformatik

Mobile Tourist Guides

Evaluation of the State of the Art and Development of a Light-weight Framework for Location-based Services

Diplomarbeit zur Erlangung des akademischen Grades eines
Magisters der Sozial- und Wirtschaftswissenschaften (Mag.rer.soc.oec)
im Diplomstudium Wirtschaftsinformatik

eingereicht bei a.Univ.-Prof. Mag. Dr. Werner Retschitzegger

Christoph Grün

Linz, August 2005

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplom- bzw. Magisterarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Danksagung

An dieser Stelle möchte ich mich bei all jenen bedanken, die durch ihre fachliche und persönliche Unterstützung zur Entstehung dieser Diplomarbeit beigetragen haben.

Herrn Prof. Werner Retschitzegger und Herrn Dr. Wieland Schwinger möchte ich für die hervorragende Betreuung meiner Diplomarbeit danken. Ihr großartiger Einsatz und ihre wertvollen Ratschläge trugen wesentlich zur Entstehung dieser Arbeit bei.

Ich möchte mich auch bei Herrn Mag. Thomas Ebert und Herrn Markus Lackinger von der DORIS Gruppe des Landes Oberösterreich bedanken, die mir eine Schnittstelle zum Austausch von geografischen Informationen zur Verfügung gestellt haben.

Mein besonderer Dank richtet sich an meine Eltern, die mir das Studium ermöglichten und mir während der gesamten Studiendauer zur Seite standen. Schließlich möchte ich meinen Geschwistern und Freunden für ihre Unterstützung während des Studiums danken.

Abstract

Context-aware mobile systems aim at delivering information and services tailored to the current user's situation. One major application area of these systems is the tourism domain. The modern tourist seeks to get up-to-date information not only in the preparatory phase before going on vacation, but especially during the vacation itself through location-based services (LBS). Several mobile tourist guides have been developed to provide tourists with information on their tours. However, these guides are rather technology-driven and do not sufficiently take into consideration the full context of tourists, thus failing at adapting information to the tourists' individual demands. This thesis aims at analyzing the state of the art of mobile tourist guides and establishes, based on the results of this analysis, a framework for LBS that aims at eliminating some of the drawbacks of existing approaches.

The first part of the thesis evaluates nine mobile tourist guides based on an evaluation framework to identify their major strengths and weaknesses. The evaluation focuses on the two orthogonal dimensions of context and adaptation as well as on specific tourism-related characteristics, such as tourist life cycle or social activity. According to the evaluation results, current systems suffer from the following main limitations: First, existing approaches often use proprietary interfaces to other systems such as a Geographic Information System – GIS, and employ their own data repositories, thus falling short in portability and depending on time consuming content maintenance. Second, often thick clients are used, admittedly offering advantages like rich user functionality, but lacking out-of-the-box usage. Third, existing solutions are sometimes inflexible concerning configuration capabilities of the system.

To tackle these shortcomings, a light-weight framework for LBS is introduced in the second part of this thesis which aims at reducing some of the drawbacks of current systems. The framework integrates external data-sources such as a GIS over an open standard interface as well as existing Web content to provide users with the information they need. The thin client approach of the framework facilitates its employment since on the client side, only a graphical, ActiveX enabled browser, an Internet connection and a GPS sensor are required. A web-based interface allows to configure the inclusion of external Web content as well as the configuration of external GIS. To prove its applicability, a prototype of a mobile tourist guide is implemented for the city of Linz, which delivers geographic data in form of maps and information about points of interest (POIs) to mobile devices.

Kurzfassung

Mobile, kontextbezogene Systeme sind dadurch gekennzeichnet, dass sie den Benutzern jene Information zur Verfügung stellen, welche für sie in der jeweiligen Situation relevant ist. Ein wichtiges Anwendungsgebiet dieser Systeme ist der Tourismusbereich. Touristen möchten nicht nur vor der Reise Unterstützung für die Planung ihres Aufenthaltes bekommen, sondern vor allem auch während der Reise. Hierbei erhalten Touristen mittels mobiler Endgeräte aktuelle Informationen durch ortsbezogene Dienste, so genannte *location-based services* (LBS). In den letzten Jahren wurden bereits verschiedene Touristeninformationssysteme für mobile Endgeräte entwickelt, welche die Aufenthaltsplanung vor Ort erleichtern und touristisch relevante Informationen liefern. Die meisten Systeme sind jedoch eher technologiezentriert und berücksichtigen nicht ausreichend den Kontext von Touristen. Die Anpassung der Information an den Kontext erfolgt daher nur in eingeschränktem Ausmaß. Diese Arbeit verfolgt den Zweck, derzeitige, mobile Touristeninformationssysteme zu vergleichen und aufbauend auf den Ergebnissen dieses Vergleichs ein Framework für LBS aufzustellen, welches einige Mängel der existierenden Ansätze aufzuheben versucht.

Der erste Teil der Arbeit vergleicht neun mobile Touristeninformationssysteme mit Hilfe eines Evaluationsframeworks, um deren Stärken und Schwächen zu identifizieren. Die Evaluierung basiert auf den zwei orthogonalen Dimensionen Kontext und Adaption sowie auf tourismusspezifischen Eigenschaften wie der Reiseablauf und das soziale Umfeld von Touristen. Folgende Unzulänglichkeiten von bestehenden Systemen kamen zum Vorschein: Erstens verwenden bestehende Ansätze meist proprietäre Schnittstellen zum Austausch von Daten, zum Beispiel zu einem geografischen Informationssystem – GIS, und eigene Datenquellen. Dadurch sind sie in der Portabilität eingeschränkt und erfordern einen hohen Betreuungsaufwand. Zweitens verfolgen sie meistens das *thick-client* Konzept, welches zwar eine höhere Benutzerfunktionalität aufweist, jedoch nicht ohne Weiteres einsetzbar ist. Drittens sind einige Systeme inflexibel betreffend ihrer Konfigurationsmöglichkeiten.

Der zweite Teil stellt ein Framework zur Entwicklung von lokationsbasierten Webanwendungen für mobile Endgeräte vor, welches darauf abzielt, einige Mängel der bestehenden Systeme zu beseitigen. Ein Hauptaugenmerk des Frameworks liegt in der Integration von externen Datenquellen. Einerseits werden geografische Informationen über eine standardisierte Schnittstelle von einem externen GIS eingebunden. Andererseits verwendet das Framework bestehende Webseiten, welche

für den Benutzer relevante Informationen beinhalten. Um dem Benutzer einen einfachen Zugriff auf das System zu bieten, wird das *thin-client* Konzept eingesetzt. Der Benutzer benötigt lediglich einen grafischen Webbrowser mit ActiveX Unterstützung, eine Internetverbindung und einen GPS Empfänger. Eine webbasierte Schnittstelle ermöglicht die Konfiguration der Einbindung von bestehenden Webseiten als auch von einem externen GIS. Zu Evaluierungszwecken wurde dieses Framework im Tourismusbereich angewendet und ein Prototyp eines lokationsbasierten Touristeninformationssystems für die Altstadt von Linz geschaffen. Dieses stellt einerseits geografische Daten in Form von Karten und andererseits Informationen von Sehenswürdigkeiten, so genannter *Points of Interests* (POIs), am mobilen Endgerät dar.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The first chapter motivates this thesis by introducing context-aware systems and presents the objectives as well as the structure of the thesis.

1.1 Motivation

E-commerce and m-commerce have dramatically boosted the demand for services which enable ubiquitous access. Ubiquity was first stressed by Mark Weiser [Weis96], envisioning a scenario in which computers will be available throughout our physical environment while making them effectively invisible to the user. Almost every object in our everyday environment will be equipped with embedded processors and wireless communications to facilitate interaction with users and to perform and control a multitude of tasks and functions.

In the area of m-commerce, ubiquity is not seen in this highly pervasive sense but rather as a challenge for people to be anywhere at anytime and to use mobile applications. The rise of mobile devices and mobile communication networks pushed the idea of making any information available to anybody, anytime and anywhere. Traditional applications, which require a physical (wired) Internet connection, may not be suited to meet these demands. They are designed for a relatively stable office or home environment, with users sitting in front of stationary computers and being able to devote a relatively high amount of attention to the tasks on the computer. The circumstances under which m-commerce applications are used differ significantly from those for desktop applications. Mobile applications have to be designed for users acting in a highly mobile environment and using a diversity of mobile devices, e.g. mobile phones or handhelds, with limited capabilities such as small screen, low battery and CPU power and unstable network communications. It is not only the environment that can change rapidly from moment to moment but also the amount of attention a user can give to a mobile application can vary over time. The user's attention not only depends on environmental factors (e.g. street noise) but mostly on the activity, the user performs at a certain time and location. The activity can imply the interaction with other people and objects in the environment as well [Tara03].

Realising the 4A paradigm (anybody, anytime, anywhere, anything) in the way of broadcasting arbitrary information to users with mobile devices does not provide those value-added services m-commerce is looking for. The main task of mobile applications is to effectively support mobile users despite the restrictions of mobile devices and the physical environment, thus preserving semantic equivalence of services. Additionally, mobile applications need to provide value-added services, thus enabling semantic enhancement. This can be achieved by taking advantage from knowledge about the situation of use [KPRS03]. For example, by knowing that a specific user (e.g. with a taste for Italian food) is at a certain location (e.g. near an Italian Restaurant) at a certain time (e.g. lunch time), the system can infer that he/she might be hungry and provides him/her with the menu and the route to that restaurant. As this example shows, the 4A paradigm is not sufficient in m-commerce since the objective of m-commerce is not to provide arbitrary information to the user but to tailor the information to the usage context. The 4A paradigm therefore has to be replaced by the more adequate 4S paradigm (somebody, sometime, somewhere, something). This paradigm aims at reducing information overload by providing the right information at the right time in the right way. In case of the above example, the 4S are a specific user (somebody) who is near a restaurant (somewhere) at lunch time (sometime) and might be interested in the menu and the route of the restaurant (something) [SpAY05].

In order to provide relevant information to users acting in a dynamic environment, mobile applications have to follow the 4S paradigm by realising customisation [KPRS03], i.e. adapting the content to the current context. Context-awareness and adaptation are thus the main key factors for a successful mobile application. Before describing these factors in more detail, the objectives and the structure of the thesis is outlined.

1.2 Objectives of the Thesis

This thesis aims at gaining an understanding of the general concept of context-aware services, with special focus on the context factor location. One of the application domains particularly suited for providing context-aware services is the tourism domain, not least since in this way, tourists can be assisted not only before their journey but also during their trip every time and everywhere. The thesis therefore concentrates on tourism applications to provide a better understanding of the concepts of context-aware services and the context factor location. The main objectives of the thesis are the following.

First, various context-aware systems in the tourism field (so-called mobile tourist guides) are evaluated. To perform the evaluation an evaluation framework is developed consisting of several criteria (context, adaptation and tourism-related) which are relevant to context-aware tourist systems. Based on the evaluation relevant technologies as well as strengths and weaknesses of current systems are identified.

The second objective is to develop a framework for location-based services and to build a practical demonstrator for the framework based on the insights gained from the evaluation. For this purpose, a prototype of a mobile tourist guide for the city of Linz is implemented that delivers geographic data in form of maps and information about points of interest (POI) to mobile devices, based on the current user's location. The framework as well as the prototype take into consideration the problem of the localisation method as well as the constraints of mobile devices such as small screen, low CPU power, no keyboard, etc.

Parts of the thesis (cf. [SGPR05]) have already been accepted for publication at the OTM 2005 Workshop on Context-Aware Mobile Systems (CAMS'05).

1.3 Structure of the Thesis

Chapter 2 - *Background*. The first part describes the key factors of context-aware systems, namely context and adaptation. Context-aware systems are closely related to geographic information systems, since they provide (geospatial) information related to the locations of mobile users. The second part of this chapter describes open standards which can be used to query, analyse and visualize geospatial data.

Chapter 3 – *Context-aware Mobile Tourist Guides*. This chapter describes an evaluation framework for mobile tourist guides, evaluates nine mobile tourist guides on basis of this framework and presents the major findings and implications of the evaluation results.

Chapter 4 - *A Light-weight Framework for Location-based Services*. This chapter describes a framework for location-based services as well as its prototypical implementation LiMoG (short for Linzer Mobile Tourist Guide) which enables tourists to explore the city of Linz.

Chapter 5 - *Outlook*. This chapter suggests some improvements and possible extensions of the framework.

Chapter 2

Background

This chapter defines the theoretical and technical background of context-aware systems. The theoretical background details the key factors context and adaptation. Since this thesis addresses context-aware systems in the field of tourism where geospatial data in form of maps are an important component, the technical background describes open standards for querying geographical information.

2.1 Theoretical Background

The adaptation of mobile services to the context of the user provides a great potential for developing value-added and more user friendly systems. Subsection 2.1.1 provides a definition of context and lists various context factors. Subsection 2.1.2 defines adaptation and presents dimensions which can be adapted with respect to the context.

2.1.1 Context

Being aware of the user's context is a pre-requisite for realising value-added services. One must understand what the context is to determine how it can be exploited and to take most advantage of it for supporting the user's goals.

Dey and Abowd [DeAb99] defined the context as "*any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves*". This general concept of context can be divided into several more specific context elements. Tarasewich [Tara03] considers three broad categories of context (cf. Figure 1), comprising environment, participants and activities. The environment category is concerned with the properties of objects in the physical environment. The category participants includes personal properties as well as user expectation and location. The category activities describes the task and goals of the users. The different categories can have inter-relations, e.g. participant/environment relationships (user at home, user at work). Chronology of time is regarded as well, allowing for a context history and a prediction of future context events.

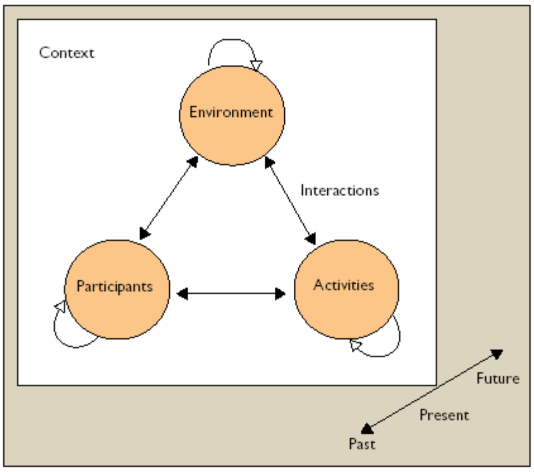

Figure 1. Graphical Representation of Context Model [Tara03]

The three categories can be further divided into different context factors.

Nivala and Sarjakoski [NiSa03] regard the following context factors in the area of mobile map services as important:

- Context *Location*: Location is the key context factor. Location information is related to a certain position and several technologies (cf. Section 3.1.3) are used to provide location information with different accuracies. The Open Geospatial Consortium has announced several open standards to enable easy exchange of location information (cf. Section 2.2).
- Context *Time*: The context time can include the time of day or time of year. Context-awareness with respect to the time of day allows for presenting e.g. the opening hours of a building to users. The time of year can be used to distinguish between different seasons, e.g. winter and summer. Reichenbacher [Reic04] combines the two context factors location and time to the context *situation*, which is a function of *location* and *time*. Reichenbacher understands the context *situation* in the original sense (lat. situs) of being situated, i.e. placed in a spatio-temporal reference system and should not be mixed up with the meaning of "situation" in the definition of Dey and Abowd, where it characterizes the whole environment.
- Context *Physical Surroundings*: The user's physical surroundings can be divided into background noise, illumination, temperature and weather. A high

level of background noise prohibits the usage of audio as a modality for input or output. The screen of mobile devices is difficult to read during the day, so the colours of the user interface and map have to be adapted to the brightness of light. Temperature and weather conditions can affect the user's subjective meaning of "near". When it is cold or raining, the area, which is defined as near to the user should be closer.

- Context *Navigation History*: Keeping track on the user's previous navigational targets may be useful to offer routes that have the most likely POIs and suitable routes for the user or to visualize the path of the user on a map.
- Context *Orientation*: Information on the user's orientation would allow for presenting the map in the right position with respect to the user's direction of movement or to present major POIs or landmarks within the user's visual angle. Studies showed that users prefer egocentric oriented maps, which are aligned to the movement direction. An electronic compass on the client device is pre-requisite. Ego-centric maps make manual reorientation redundant and thus reduce the cognitive effort of users to match between reality and map view. But if more than one POI is displayed on the map an orientation independent depiction (e.g. a north oriented map) is advantageous [ZiJo04].
- *Cultural and Social* Context: Social context includes reasoning whether the user is alone or acts in a group of other people. Cultural context provides information on the cultural situation of the user. The colours used can have different meanings in various cultural areas and the information can be presented in different languages.
- Context *System*: The characteristics of the system in use constitute another important context factor. The diversity of mobile devices is growing. Users expect to use the same application on various devices, although they have different capabilities and limitations concerning screen size, type of display, CPU power, etc. Since mobile devices depend on wireless, unstable network communications, the quality of network connectivity is not always the same. In this way, device and network characteristics have a great influence on the way how information is transmitted and visualised. A standard to describe the device capabilities is the Composite Capabilities/Preference Profile (CC/PP). CC/PP[1] is based on RDF, the Resource Description Framework, which was designed by the World Wide Web Consortium (W3C)[2] as a general purpose metadata description language.

[1] CC/PP. http://www.w3.org/Mobile/CCPP

[2] World Wide Web Consortium (W3C). http://www.w3.org

- Context *User*: User context investigates demographic attributes such as gender, or age of a person as well as personality variables, e.g. preferences, experiences and skills or learning history.
- Context *Purpose of Use*: One of the most important context factors is the user's purpose for using the context-aware application. It might be difficult to identify the tasks of a user as well as his/her activities which should be supported by the mobile application. According to the activity theory, an activity is a sequence of actions conducted by a human being aimed at achieving a goal. The goal can be a problem or a task of the user. Elementary activities are localising, way finding (e.g. planning a route to a destination) or searching.

Since it is difficult to consider all context elements, the mobile application has to pay attention for those which have the most relevance for the information usage. Location is a key context factor, but its sole use does not always lead to value-added services. An approach that considers *more context factor*s such as time, device, user activity, etc. has a better chance to provide more useful information. The more the system considers different context factors, the merrier it is able to provide the "right" information to the user. A context-aware system requires less user attention and enables the user to concentrate on his actual activity, thus improving the general usability.

In addition to identifying the relevant context factors it is also important to model the *relationships* between them and to consider the *validity* of context. For example, the context time (e.g. the season) can have an influence on possible activities (e.g. in winter users are able to ski on lakes whereas information on swimming facilities may not be as useful for people during the winter term). Schmidt and Gellersen [GeSB02] describe the validity of context as being dependent on space and time. With increasing spatial or temporal distance from the existence of a context instance, the validity is decreasing.

Context information can be either derived *from different sensors* or accessed from *external sources*. Considering the location context, for example, location information can be provided automatically by a GPS sensor. User context is more likely to be *provided by the user* in a manual way, but may be updated over time based on the user's interaction with the system. Some context information can be accessed from other sources, e.g. the weather conditions for a given location can be queried from a meteorological web service [Reich04].

2.1.2 Adaptation

Adaptation is the process to fit the system to the current usage situation. Prerequisite for realizing adaptation is that the system has knowledge of the usage context. The main focus of adaptation is on providing the user with more relevant and accurate and thus adequate information that meets his/her needs better and allows a fast capturing of the information essence. In the mobile environment, this is of great importance since the user's main focus is on his/her current activity as well as on the dynamic environment and since the screen of a mobile device is small and does not allow for much information presentation [Reic04].

2.1.2.1 Adaptable vs. Adaptive

There are two forms of adaptation. "A system is called *adaptable* if it provides the end user with tools that make it possible to change the system characteristics" [Oper94]. The other form of adaptation is adaptivity. "A system is called *adaptive* if it is able to change its own characteristics automatically according to the user's needs" [Oper94]. The difficulty lies in defining a good balance between a system which adapts automatically to a new context and one that does not adapt but supplies a full range of features that can be accessed by the user [Reic03]. If a system is completely adaptive, the user might get a feeling of losing control over the system. But a completely adaptable system seems not ideal either because it requires too much interaction of the user with the system [Reic04]. A compromise would be a system that provides both types of adaptation: automatic adaptation solely by the system and adaptation that implies some user interaction. For example, the system initiates adaptation and proposes some changes/alternatives, the user then decides upon which action to be taken and the system executes the user's choice.

Automatic adaptation should be used if only minor changes in the context are necessary. Good examples for self-adaptation by the system are "updating the location information on the map as the user moves" or "zooming the map dependent on the current speed of the user". In these cases the user would feel disturbed by interaction dialogs. This could be observed during evaluations of the EU-Project TellMaris [ScCL05]. In the first evaluation phase, GPS for sensing the location was not available. The users had to interact with the model all the time in order to keep the view in the screen up-to-date and were therefore rather annoyed [ScCL05].

A *user-controlled adaptation* is for example appropriate when making tour proposals. If not enough time is available to finish a tour, the system should not calculate a new one without notifying the user. Instead, it should make some suggestions and let the user ultimately decide which POI to include in the new tour.

2.1.2.2 *Adaptation Dimensions*

Content information itself is not the sole target of adaptation. Not only the information content must be adequate but also the presentation form has to be adapted to the usage situation. For example, audio might not be the adequate means to present information to a user when he/she is in a church or in an area with high background noise. According to Reichenbacher [Reic04], the following *adaptation dimensions* can be distinguished:

- *Information domain*: the information content can be adapted to the current context, e.g. by filtering those POIs which are in the vicinity of the current user's location.
- *User Interface domain*: the user interface can be adapted to the current context, e.g. by enabling/disabling certain functions or changing the interaction mode, from pen-stylus input to voice interaction.
- *Presentation domain*: the presentation of information can be adapted to the current context, e.g. from visualisation presentation to an audio output.
- *Technology domain*: the information encoding can be adapted to specific devices with different characteristics or to the transmission media (e.g. network bandwidth). When there is a low network bandwidth, colourful images can be replaced by black/white ones.

2.2 Technical Background

The goal of this section is to describe the technical background of this work and its related fields. Context-aware systems strongly depend on geospatial data and are therefore closely related to GIS (Geographic Information Systems), not least since location is a major context factor. The main focus of this section is therefore on shedding some light on open GIS standards and services. According to Huxhold & Levinsohn [HuLe95], a geographic information system is "*a collection of information technology, data, and procedures for collecting, storing, manipulating, analyzing and presenting maps and descriptive information about features that can be represented on maps*".

The Open Geospatial Consortium[3] (OGC) is concerned with defining open software interfaces for geographic information which facilitate the exchange of geospatial data over the web. A huge amount of different geographic data formats has been developed during the last decades. Most of them are proprietary to the vendors of GIS systems. As a result, the exchange of geospatial data as well as the change from one system to another was difficult. Until recently, GIS systems have been mainly used by experts, who had to be trained for their usage. With the emergence of context-aware systems the demand for geo-services that can be used not just by experts but by anyone has increased and asks for standards to exchange geospatial data and provide new geo-services.

2.2.1 Geography Mark-up Language - GML

GML [LaBT04] is an XML based encoding standard to model, encode, transport and store geographic information. Since GML is based on XML, it is easy to exchange GML data over the web. Besides, it inherits all advantages of XML such as that it is human readable, easy to understand, etc. GML schemas, which are based on XML schemas, describe the structure of GML data and define elements and attributes that are used in data instances. GML is used to describe geographic objects of the world. In GML, real-world objects are called features which are categorized into particular types. GML features can be concrete and tangible (e.g. buildings, streets) or abstract and conceptual (e.g. political boundaries). A feature is described in terms of its properties. Properties can be geometric (e.g. location, form) or non-geometric (e.g. colour, height). For example, the feature “school” can be described by the geometric property location and the non-geometric properties name or level of education taught.

GML can be easily converted in a different language using XSL (Extensible Stylesheet Language) transformations. In this way, GML data can be translated into SVG (Scalable Vector Graphics) format in order to display the spatial data in form of maps.

GML is only one part of the work of the Open Geospatial Consortium. It also produces standards for geospatial web services such as the Web Map Service (WMS) or Web Feature Service (WFS). In general, web services are applications that respond to requests sent to them from remote applications over the Internet.

[3] Open Geospatial Consortium. http://www.opengeospatial.org

Geospatial web services are web services which provide access to geographic data and perform some data processing.

2.2.2 Web Map Service - WMS

A WMS [Kolo03] is a non-propietary geospatial service, developed by the Open Geospatial Consortium. A WMS allows spatial data to be shared via HTTP in the form of map images. Map images can be requested from different WMS servers and be overlaid on the fly. The map returned by the WMS server is a raster image. The image format can be JPEG, GIF, PNG or SVG. In this way, a WMS server controls access to sensitive geospatial data because it only allows a view on the geospatial data. A WMS server must at least support two operations, namely a *GetCapabilities* request and a *GetMap* request. The *GetCapabilities* request retrieves metadata about a WMS server. As an answer to a *GetCapabilities* request, a WMS server returns an XML document to the client that contains information about the capabilities of the WMS server, including the geographic area covered, the thematic layers available, etc. This metadata information allows the client to specify a *GetMap* request. A *GetMap* request consists of a set of parameters, such as the geographic extent of the requested image, the image format, the coordinate reference system, the size, etc. When the WMS server receives such a *GetMap* request, it returns geographic information in form of a map image.

2.2.3 Web Feature Service - WFS

A WFS [SURA] is similar to a Web Map Service (WMS), but with the difference that it returns vector data in XML format. Vector data are points, lines, polygons in contrary to raster data which are regular grids. Like the Web Map Service, the WFS also has a *GetCapabilities* request. The *GetMap* request is replaced by a *GetFeature* request. The *GetFeature* request allows the client application to access, query, create, update and delete data elements from GIS database servers over the Web. When a client queries spatial data, a WFS server extracts the data, transforms it into GML and transports this GML document to the client.

2.2.4 Open Location Services - OpenLS

The OGC specification Open Location Services (OpenLS) [Harr03] defines access to Core Services and Abstract Data Types (ADT). An ADT is the basic information construct, such as POIs, positions or routes. The main important Core Services are

Directory, Route and Presentation Service. The access is defined by request-reponse pairs. The Directory Service provides access to a directory to find the nearest or a specific place, product or service. The client sends a request with a position, a category or keywords and gets as response an instance of a point of interest ADT. The Route Service provides the client with a route containing a list of geographic positions along the route. The presentation service displays geographic and service data (instances of ADT) as a map in PNG, JPEG, GIF or SVG format.

2.2.5 Scalable Vector Graphics - SVG

SVG [Reich02] is a vector format which is well suited for displaying GIS information. SVG vector graphics are smaller in file size than raster images and can be displayed on any device with any size and any resolution without changing clarity. So there is no loss in detail while zooming which allows for better readability. For mobile devices, SVG offers two profiles, SVG Basic and SVG Tiny. SVG Basic is aimed at PDAs and SVG Tiny at cellular phones. Since a SVG file is an XML file, it is extremely portable and can work directly with XML-based technology. As an XML application SVG offers hyperlinks to other files as well as vector and raster graphics. By using XSLT or CSS stylesheets, SVG allows for grouping, styling and transforming of 2D graphic elements such as rectangles, circles, ellipses, lines, polygons, symbols and path elements. SVG drawings can be dynamic and interactive. SVG documents are scriptable by using the Document Object Model (DOM) for SVG. SVG is therefore useful for building dynamic web pages with 2D graphics, whose visualisation can be adapted to the context on-the-fly.

Chapter 3

Context-Aware Mobile Tourist Guides

One of the application domains particularly suited for providing ubiquituous access on basis of customization is the *tourism domain*, not least since in this way, tourists can be assisted not only in the preparatory phase of a vacation but especially during the vacation itself (cf. e.g., [GPSP04], [PrRe00], allowing access with any media, at anytime, from anywhere (cf., e.g., [BeLL039]. Such applications supporting the tourist on the move by means of location-based services are often called *mobile tourist guides.* They provide the tourist, for example, with personalized on-site tourism information about POIs (e.g., environmental and landscape attractions or gastronomy), or assist the tourist in organizing an individual tour. A series of such mobile tourist guides have recently been proposed, offering a wide range of functionalities with respect to context-awareness and adaptation (cf. e.g., [AKMP02], [BBKK04], [GrHa04], [HiVo03], [HrHL03], [Kama03], [KrBB04], [PLMN01], [Roth02], [SePK04]). This chapter identifies the strengths and weaknesses of existing approaches. For this, an in-depth survey of existing mobile tourist guides is conducted providing the basis for next-generation mobile tourist guides. In contrary to other surveys like [BaCK05] this survey deals with mobile tourism guides which are *web-based*, applying a broad view on context-awareness comprising not only location and device capabilities but also personalization and other context properties like time or network. In the light of that, an evaluation framework comprising detailed evaluation criteria for context and adaptation as well as tourism-related criteria is used.

The following section (3.1) gives a brief overview on the evaluation framework. This evaluation framework is then applied in the next section (3.2) for in-depth investigation of nine mobile tourist guides. The lessons learned (3.3) finish this chapter.

3.1 Evaluation Framework

This section briefly gives an overview on the evaluation framework used as the basis for comparing the customisation capabilities of Web-based mobile tourist guides. This framework is characterised by two orthogonal dimensions, comprising context and adaptation, and the mapping in between represented by the notion of customisation (cf. Figure 2). To ensure traceability of the different criteria in the evaluation of approaches in Section 3.2, appropriate abbreviations are used for each criterion.

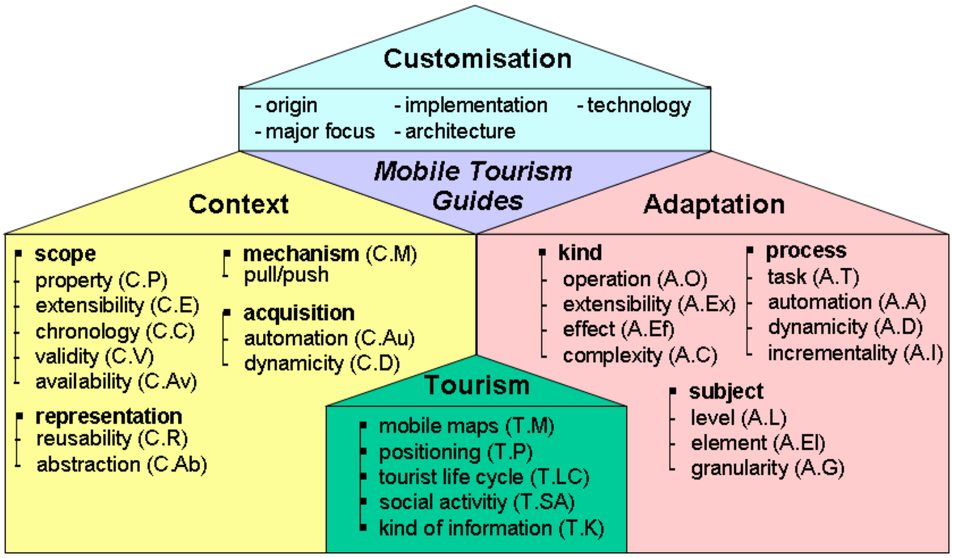

Figure 2. Evaluation Framework

In the following, the criteria for context (C) and adaptation (A) are outlined, resigning a description of the common criteria since they are self-descriptive. The criteria for context and adaptation are complemented with criteria which are typical for mobile guides in the area of tourism (T). Concerning the criterion “Architecture” it is distinguished between an internal customisation, meaning that customisation mechanisms are intermingled with the application to be adapted, whereas external customisation refers to a proxy-based customisation approach.

3.1.1 Context

Context characteristics which are relevant for the management of context data can be categorized into *scope* of context, its *representation* and *acquisition,* as well as the *access mechanism* used (cf. also [KaSR05], [KoKP01], [ScLa01]).

Scope. The *scope* of context comprises the different context *properties (C.P)* supported by the system (such as location, time, device, network, and user) together with the ability to *extend (C.E)* them to cope with unforeseen requirements. For each of those properties also the time dimension needs to be regarded, represented by *chronology (C.C) in terms of historical, current* and anticipated *future context* along with *validity (C.V)* and *availability (C.Av)* of context data.

Representation. The *representation* of context comprises two important issues. First, mechanisms for enhancing context *reusability (C.R)* by, e.g. drawing from various context sources like GIS or existing device profiles. Second, the level of *abstraction (C.Ab)* at which context is represented, separating physical context in form of sensed context data and logical context which is derived on bases of inference mechanisms or provided by profiles.

Acquisition. The *acquisition* of context can be characterised by the degree of *automation (C.Au),* considering *who* is responsible for acquiring the context that is either a human (manual), the system (automatic) or a combination thereof (semi-automatic) and the degree of *dynamicity (C.D)* in terms of *when* the context is acquired being either statically considered at system startup or dynamically at run-time.

Mechanism. The *mechanism (C.M)* by which context is acquired and made accessible to the services using the context can be either *pull-based* when a request to the system is issued or *push-based* when context changes.

3.1.2 Adaptation

The second dimension of the evaluation framework is covered by the notion of adaptation, characterised by the *kind of adaptation*, i.e., what changes have to be done, the *subject of adaptation* in terms of what to change and the *process of adaptation* characterising how adaptation is performed.

Kind. The *kind* of adaptation subsumes built-in *adaptation operations (A.O)* such as filter content, add links or change image resolution and possible *extension (A.Ex)* mechanisms to introduce new user-defined adaptation operations into the system. Adaptation can *effect (A.Ef)* the system in that certain parts of the system are added, removed or transformed and can be, as a series of adaptation operations, combined to *complex (A.C)* adaptations

Subject. The *subject* of adaptation can be characterised by looking at the *level (A.L)* of the Web application which is affected by the adaptation, comprising content level (i.e., domain-dependent data), hyperbase level (i.e., the navigation structure) and presentation level (i.e., the layout of each page together with user interaction facilities). Each of the levels contains several application *elements (A.El)* that can be adapted (e.g., pages, links, access structures, input fields or media types). Finally, the *granularity (A.G)* of adaptation indicates the number of application elements affected by a certain adaptation distinguishing between micro- and macro-adaptation.

Process. The *process* of adaptation comprises a number of *tasks (A.T)* which should be separated into, e.g., initiation, proposal, selection to allow a fine grained control of their degree of *automation (A.A).* The degree of *dynamicity (A.D)* defines whether the adapted versions are already available or generated at run-time [KoKP01]. Finally, adaptation can be either conducted *from scratch* or *incrementally (A.I)* meaning that adapted versions are made persistent so that subsequent adaptations are conducted on basis of results of previous adaptations.

3.1.3 Specific Criteria of Mobile Tourist Guides

Besides evaluating mobile tourist guides according to their context awareness and adaptation functionalities, mobile tourist guides can be investigated by comparing some general features they have in common. The features are either an essential part of the system in order to provide value-added services to tourists and therefore need to be pointed out or refer to the specific nature of tourism systems.

Mobile Maps (T.M)

Tourists traditionally use paper maps and guidebooks as their main utilities when they explore and visit POIs. Maps are used for wayfinding and help to build a mental model of the spatial surroundings, guidebooks offer detailed information relevant to the POIs. Mobile tourist guides have to provide an equivalent service to compensate for the paper maps and guidebooks and to be able to support mobile users during their trip. The main idea is that users can download information related to their current usage situation on the mobile device. This information consists of a map which displays their current surroundings as well as of a set of POI data. Both map and POI data are generated on the fly as soon as the user issues a request and are adapted to his/her current situation.

There are several advantages of mobile maps in comparison to paper maps and guidebooks. A mobile map does not only provide up-to-date information generated on the fly. It is somewhat like a snapshot of an environment at a certain location and time, but with highly selective information and integrated intelligence [MeRe05]. In this way, a map and its objects can be filtered and adapted to the user's current situation. Mobile maps can not only link text but also multimedia information to the objects on the map. The general statement that map use is a process affording highly cognitive abilities and training may be valid for conventional maps but not for mobile applications. Conventional maps are overloaded with lots of information to satisfy several users with different requirements. Mobile maps, however, focus on providing just the information (e.g. POIs) that is relevant to the user at the current situation, thus allowing for a fast and effortless capturing of the information essence. POIs are often structured and displayed in layers on a base map to prevent information overload on the small screen.

In spite of these advantages, traditional paper maps will continue being the main concurrent of mobile maps. Paper maps are portable, cheap, foldable, need no electric power, can be shared in groups and can be annotated easily by various forms of marker. To compensate their drawback of being static, recent technologies target on overlaying paper maps with digital information services. To combine paper maps and digital services, a printed map can be used as the primary user interface and digital overlays can link positions on the map to appropriate digital information and services. This way, the paper map provides geographical information which is rather static. The digital layers provide dynamic information, such as information about cultural events taking place in a city. For this case, almost invisible infrared-absorbing dots are printed on the map, encoding the x-y positions. A special digital pen serves as the medium to determine the positions on this map. In this way, people can access additional information which is dynamic and dependent on various context factors such as location and time [NoSi05]. Although this approach is a good way of adding dynamic information to traditional paper maps, the question still remains whether digital or analogue maps are sufficient to support mobile users. They act in a highly dynamic mobile environment and request different information dependent on their current activity, which is mostly gathered from different data sources [Reic05].

Positioning (T.P)

Mobile tourist guides provide their users with LBS. Determining the user's position is the pre-requisite for providing further services to the tourists as they are mostly based on the user's current position. Since the user's location is a key factor, several technologies for determining the user's position should be investigated. Positioning has to be adequate to the provided service and adapted to the tasks of users. As an example, it does not make sense to measure location accurate to a millimetre if the position is displayed on a small scaled map. Various methods of positioning are available for different levels of accuracy, such as satellite-based or network-based positioning methods [Gart04]. (For an overview of the basic techniques used for location-sensing, cf. e.g. [HiGa01]). In general, indoor LBS require a more accurate positioning method than outdoor positioning (such as GPS). For example, Infrared Beacons provide accurate positioning but they can only be used inside buildings because of their short range. Hybrid positioning methods, such as A-GPS [Swed99], can enhance the accuracy. A-GPS is short for assisted-GPS and combines cellular positioning methods with GPS. Especially in cities, satellite-supported positioning can cause problems due to alleyways blocking the signals from the satellites. Galileo [Mund04], the European Satellite Navigation System, will certainly lead to an improvement in this context, but will still have some restrictions (e.g. blocking of signals). The concept of Active Landmarks [GaUh05] can be an answer to that problem. Landmarks are prominent objects, which can be used as reference points. Landmarks help pedestrians in navigational problems and at decision-points [RaWi02]. The idea of active landmarks is that not the user is located but he/she moves from one known position (landmark) to the next known. As soon as the user comes into range of one of these landmarks, an ad-hoc network is automatically built up with the mobile device, enabling the user to identify his/her position and to get information.

Support of Tourist Life Cycle (T.LC)

Tourist applications should support the whole tourist life cycle, consisting of three phases: *before trip*, *on site* and *after trip* [HuZi03]. The main focus of mobile tourist guides is on providing tourists with up-to date information *during their trip*, e.g. when standing in front of a monument or looking for the next café. In the *before-trip* phase, tourists deal with planning the itinerary by gathering information about the destinations, hotels and interesting places. Pre-visiting as an activity also takes place while on the holiday itself. Tourists use maps and guide books before visiting a place. In this way, they can learn about the place and can decide whether it is worth going there. Besides pushing information to users when they are at a certain location, it is therefore necessary that mobile tourist guides provide tourists with "Virtual Visits" of places by enabling them to access and receive (multimedia) information about attractions before they actually get there. When the tourist *returns from a tour*, he/she likes to have a reminder of the holiday trip and share their travel experiences with other people at home. For this case, the tourist application should provide services that allow tourists to document their trip, e.g. by annotate their tour with pictures, videos or text [BrLa05].

Tourism as Social Activity (T.SA)

Tourism is a *social activity*. Tourists tend to travel in groups and typically collaborate around maps and guidebooks to locate their position and to discuss which attractions they should visit next. Thus, group-interaction is an important tourist activity during sightseeing to explore an attraction or learn about the environment in a collaborative way. Since mobile tourist guides are designed for mobile devices which offer a small screen or can only be used by one user at a time, they lack the ability to be shared during interaction in the way as paper maps can be. Collaboration around the small devices is difficult and still an open issue mobile devices are faced with. Mobile tourist guides can nevertheless enhance the co-visiting aspect. Tourist groups often split in smaller groups which makes the coordination between their activities rather difficult. Mobile tourists guides can support the coordination between the groups by allowing them to communicate their location or additional attractions, routes or recommendations. A friend finder service would provide such a feature. It lets each group see the location of the others' and thus helps getting back together again [BrLa05].

Kind of Tourism Information (T.K)

Points of Interests (POIs). Mobile tourist guides have to provide tourists with POI information. To visualize the spatial relationship of POIs, a good way is to mark their location on the map with a *POI symbol*. The spatial proximity can be used as a parameter to filter those attractions which are in a certain distance to the tourist's location, e.g. in form of a *recommending service*. Another functionality using proximity as a trigger are push-services, e.g. *pro-active tips*, which push a notification on the screen of the mobile device, when the user gets close to a sight or object of interest. But not only attractions in the near distance are relevant to tourists. Of even greater importance is the presentation of tourist activities in the form of *thematic overlays* on the map. The idea is that according to the activity, the user is interested in, different layers are superimposed on the map, such as a night life layer showing bars or clubs, a shopping layer or a restaurant layer.

Tours. Tourists use tours as a popular mechanism to learn in a structured manner about attractions. They are often pre-defined according to a certain theme, such as a Museum-tour but can further be personalized according to user's interest, proximity, historical period or type of monuments. Tours should not only show the shortest path from the user location to a POI but should also aim at promoting the attractions of a city and at offering a structured overview of interesting sights. Since tourists often use tours to get to know places which are worth visiting and do not strictly follow a suggested path, the system should allow for tour interruptions as well as re-calculating a tour according to the user requirements [HuZi03].

Navigation support. Although maps already provide some means for wayfinding, a routing service that calculates routes reduces the cognitive load of users and leads to better orientation. By using a routing service, tourists do not have to synchronize between map and reality all the time to see if they are on the right way but can simply follow the routing instructions. The quality of a routing service certainly depends on the ability of the mobile tourist guide to determine the user's location to a high accuracy [GaUh05].

3.2 Evaluation of Mobile Tourist Guides

There are numerous web-based mobile tourism guides proposed (cf., e.g. [AKMP02], [BBKK04], [GrHa04], [Kama03], [KrBB04], [PLMN01], [Roth02], [SePK04]). This survey focus particularly on those web-based mobile tourist guides offering the user a map-oriented interaction paradigm because such systems are more and more entering every day live. Furthermore, those approaches were chosen which offer unique features with respect to context-awareness and adaptation. Each of the nine selected approaches is described in the following in a separate section according to the evaluation framework. The results of the evaluation are presented in four tables in between, thus giving an overall understanding of each approach before discussing particular results of this investigation in Section 3.3.

3.2.1 COMPASS

COMPASS [SePK04] – short for COntext-aware Mobile Personal ASSistant – provides individual tourists (T.SA) during their trip (T.LC) with context-aware recommendations and services, based on the tourist's location. The location is either obtained from the mobile network or from a GPS sensor (T.P). The system provides tourists with a map and different objects displayed on the map. These objects offer additional services such as retrieving information about POIs, reserving a table in a restaurant or calling a friend (T.K). For delivering a map, it uses various external map services through proprietary interfaces, such as Microsoft's Mappoint for regular maps, a map service providing orthophotos and a map service providing cadastral maps (T.M). It builds on the open Web Architectures for Services Platform *(WASP)*[4], which supports context-aware applications based on web services. The WASP platform operates on top of 3G networks and requires a permanent network connection (e.g. GPRS). Realizing external customization, COMPASS uses a r*egistry* that contains information about the third party services providing the content such as museums and restaurants information. For service description, semantic web technology such as OWL is used.

The system retrieves and provides information about the user's context by contacting the appropriate context services (C.R). Location context (C.P) is considered as the primary criterion to select relevant services in the near surroundings of the user. User context (C.P) comprises a manually provided profile, which is described using an extended P3P specification indicating the user's interests that is further automatically (C.Au) updated by the system based on the user's feedback for specific POIs. New context properties like weather or traffic information can be incorporated into the system via web services (C.Ex, C.V). The time context (C.P) is considered in the following way: The more recent the user has been at certain POIs, the lower is the predicted relevance that the user wants to visit a POI of that class (C.C). Arbitrary logical context, for example, whether a user walks or drives can be derived on bases of the given current speed and the geographical properties by domain specific rules (C.Ab) not mentioning the underlying rule technology. A subscription mechanism is offered so that components of the system can be notified as soon as the context changes (C.M). The location information is obtained automatically (C.Au) through GPS or from the mobile network (e.g. GPRS, UMTS). Changing context both for physical as well as for logical context is considered dynamically (C.D). Validity is not explicitly supported (C.V).

The adaptation operations comprise filtering the services provided by external services based on the user profile and location (A.O, A.G). The map and the POI symbols are dynamically (A.D) and automatically (A.Au) updated as soon as the context changes. The adaptation process comprises several successive steps (A.C) such as filtering content, making recommendations as well as displaying the result

---

4 WASP Project. http://www.freeband.nl/kennisimpuls/projecten/wasp. 2005.

on the screen. All three levels are subject of adaptation. At the content level this comprises elements such as text, images, links and maps (A.El), which are added or removed according to the user's context (A.Ef). At the hypertext level, links of the POIs, which are relevant to the user's location and preferences, have to be generated. The presentation level is affected by displaying the map with the user's position as well as the POI symbols. The user can not engage in the adaptation process (A.T). Adaptation extensibility is not provided (A.Ex). The adaptation process is done fully automatically (A.Au) and non-incrementally (A.I).

		Origin	Major focus	Architecture	Implementation	Technology	Application
Approach	COMPASS	context-aware recommendation system	integrating context-aware and recommender systems in mobile tourist applications	internal/ external	implemented	Web Service Technology, XML, OWL, P3P	context-aware tourist guide
	CRUMPET	mobile services personalised for tourism	supporting nomadic users with services	internal/ external	implemented	FIPA Agents	location-aware tour guide
	The GUIDE system	location-based services	context-aware generation of tourist information	internal	implemented	HTML meta-tags	location-aware tour guide
	Gulliver's Genie	artificial intelligence and agent systems	agent-based, intelligent content delivery	internal	implemented	Agent Factory (AF), Java	location-aware tour guide
	LoL@	application development for restricted devices, i.e. GSM / UMTS phones	UMTS application	internal	implemented	Java, Java Applets, XML, XSL	location-aware tour guide
	MobiDENK	location-aware system	location-aware system for historic sites	internal	implemented	Java	tour guide for historic sites
	m-ToGuide	location-based services	promoting use of 2.5/3G cellular networks	internal	implemented	Microsoft's .net framework, Microsoft's C#	context-aware tourist guide
	PinPoint	context-aware web applications	making web-pages context-aware	internal	implemented	HTML meta-tags, Java	web-based tourist guide
	Sightseeing4U	adaptive multimedia	dynamic creation of personalised multimedia content	internal	implemented	Java	personalised city guide

Table 1. Overall Comparison of Approaches

3.2.2 CRUMPET

CRUMPET [PLMN01], [SPNZ02] is a EU project aiming at the "Creation of User-friendly Mobile Services Personalized for Tourism" relying particularly on agent technology. The system supports individual nomadic users (T.SA) with information and recommendations about tourist attractions, restaurants or hotels (T.K) while wandering along (T.LC). The user's location is determined with a GPS device (T.P). The system provides pro-active tips (T.K) when the user gets near a sight that might be of interest, supports interactive maps showing the position of the user as well as interesting sights (e.g. relying on OGC[5] standard interfaces such as the Web Map Service (T.M)). Its architecture foresees an external customisation approach allowing the integration of various service and content providers via a dedicated interface.

CRUMPET considers location, device, network, and user context properties (C.P). With respect to extensibility, it is not foreseen to incorporate further context information (C.E). Physical location context is provided in form of GPS sensor coordinates. However, no information is provided about how the other physical context properties are acquired. Transformation of physical location context to logical location context encompasses, in a first step, sending only "relevant" location changes to the server and based on that, utilizing geo-coding services to infer addresses from the given coordinates in a second step. The logical device context is taken into account in terms of the device type determining, e.g., size of display, colour depth, and raster vs. vector graphics capability. The logical network context is considered with respect to the quality of networking service (QoS) and the type of wireless connection that is available (WLAN vs. GSM). The user's interests representing the logical user context are learned dynamically (C.D) by tracking user interaction, thus taking into account the history of user context (C.C). Availability issues of context are not considered in CRUMPET (C.Av). The representation of logical context is different for the various context properties supported. Logical location context is represented explicitly in a GIS storing geographic data of the region. The logical user context is based on a domain taxonomy of tourism related services and a probability function of user's interests in that services (C.Ab, C.R). An explicit representation of logical device or network context is not stated. All context properties are acquired dynamically (C.D) and in an automatic way. The user context additionally can be accessed by the user and augmented manually (C.Au). As long as few data about the user's interests are available because little user interaction with the system has occurred, the system uses stereotypes to avoid unreliable information. Validity of other context properties is not accounted for (C.V). Context information is accessed both pull-based and push-based as soon as the user approaches a certain POI (C.M).

Adaptations are performed by locating and querying suitable content and service providers and adapting the query outcome to the context. Although device and networking conditions are considered as context - apart from stating that results are

5 Open Geospatial Consortium. http://www.opengeospatial.org. 2005.

combined into "digestible" information suitable for the user's device - there is no further information on how CRUMPET makes use of this information for adaptation purposes. CRUMPET allows to dynamically (A.D) perform a stepwise adaptation (A.C) on the results of the previous step, starting each time a service is requested on basis of the original query result (A.I). CRUMPET offers a series of adaptation operations covering sorting, filtering, highlighting and generation of new information (A.O), which cannot be extended (A.Ex). In principal, CRUMPET offers adaptation on all levels (A.L): On the content level, sorting is applied on the query results received from the content and service providers according to the user's preferences. Filtering is executed for the pro-active tips. New information is generated as personalized tour proposals are assembled. On the hypertext level, a list of links to POIs is generated dynamically (A.D) according to the user's position and interests. On the presentation level, the user's current position is highlighted on the map and objects of interest are highlighted. Thus, the adaptation operations considered by CRUMPET comprise adding, removing and transforming adaptations (A.Ef) focusing on elements such as texts, links, images, videos and maps (A.El). The application is adapted on a macro level by adaptation operations affecting the selection of services, the computation of tour recommendations and pro-active tips as well as the adaptation towards the connection type. On the contrary, sorting the list of query results and indication of the user's current position on the map represent examples of micro adaptations (A.G). All adaptations are performed automatically (A.A). The user is given some control over the adaptation process in that pro-active tips can be enabled or disabled and the assumptions about his/her interests can be influenced (A.T).

3.2.3 GUIDE

The *GUIDE system* [CDMF00a], [CDMF00b] stems from the area of location-based services. The focus is to provide tourists with up-to-date and context-aware information about a city via a PDA. The GUIDE system is based on a client/server architecture, with a Fujitsu TeamPad 7600 used as a terminal. Based on the closest access point (T.P), the client determines the approximate location of the user and provides individual tourists (T.SA) with information about sights, a 2D bitmap map (T.M) and the possibility of creating a tour (T.K). The GUIDE system's main feature is to support tourists during their trip, but the tourists are provided with the ability to search for information, irrespective of their current location (T.LC). The system also offers cooperative tools. Users can let other members of their group know their location. They can also share their experiences by leaving so called virtual stick-on notes at specific locations. To stay in contact, the GUIDE system offers a messaging service which enables users to either communicate with the Tourist Information Center staff or other GUIDE users (T.SA). The access points broadcast pages of information frequently accessed by users in the geographic area of the cell. The pages are cached on the end system.

Although the focus of the system is on providing location-based services, a more comprehensive logical context model in terms of profiles is provided (C.Ab), distinguishing between so-called *personal context* in terms of information about the user (e.g., preferences, current location and a history of already visited attractions (C.C)) and so-called *environmental context*, comprising information about attractions (e.g., links between nearby attractions, opening and closing times, relevance to user interests) (C.P). Thus, logical context is mainly specific to the tourism domain (C.R). There are neither mechanisms to extend the pre-defined context properties (C.E) nor inference mechanisms to automatically derive higher-level logical context (C.Ab). The current physical location context is gathered automatically (C.Au) at runtime (C.D), although the user is able to manually enter the current location in case that cell coverage is temporarily left. By providing the physical location in this way the location information can be regarded as valid (C.V). Logical context is, in principle, entered manually, certain information about the user, e.g., user interests are acquired semi-automatically based on the interaction history (C.Au). Context information is accessed in a pull-based manner (C.M). Availability of location context is regarded by informing the user about location information updates (C.Av).

Taking a look at the adaptation features of GUIDE, it is distinguished between *coarse-grained adaptation*, e.g., changing the language of the descriptions and *finegrained adaptations*, e.g., presenting information about the current context or filtering/sorting information depending on a certain context (A.O), (A.Ef). Thus a certain adaptation cannot only affect a single but rather numerous web pages as is the case when changing the language or when generating a complete guided tour (A.G). In particular, the subject of adaptation comprises all three levels (A.L), focusing on text and link adaptations without changing the modality (A.El). Adaptations can be complex, e.g., location-based filtering can be followed by a sorting operation before presenting the adapted web page to the user (A.C). Adhering to an integrated architecture, adaptations are realised by web pages intermingling with proprietary HTML meta tags which allow to query the context (e.g., determining the user's interest in that particular attraction which has a certain historical value associated) and to perform the appropriate adaptation (e.g., insert a user's location or insert nearby attractions). Extensibility of this pre-defined tag set is not foreseen (A.Ex). In principle, there is no separation between the different tasks of the adaptation process (A.T), adaptation is done fully automatic (A.A). The tags are interpreted on the fly, thereby realising dynamic adaptation (A.D) as soon as the user accesses a context-aware web page. Concerning dynamic adaptation however, there is a separation of tasks with respect to the computation of nearby attractions. This production task is done automatically, immediately after the location context has changed, whereas the presentation itself is done upon a user's request. Finally, incremental adaptation is not supported (A.I).

		Scope of Context											Representation of Context		Acquisition of Context					Access to Context	
		Property						Extensibility	Availability	Chronology		Validity	Reusability	Abstraction	Automation			Dynamicity		Mechanism	
		location	time	device	network	user	application			history	future				manual	semi-automatic	automatic	static	dynamic	push	pull
Approach	COMPASS	✓	✓			✓		✓	✓	✓		✓	✓	✓		✓	✓		✓	✓	✓
	CRUMPET	✓		✓	✓	✓				✓		✓	✓	✓		✓	✓		✓	✓	✓
	The GUIDE system	✓	✓			✓			✓	✓		✓		✓	✓	✓			✓		✓
	Gulliver's Genie	✓		✓		✓				✓	✓	✓	✓	✓		✓	✓		✓	✓	✓
	LoL@	✓							✓			✓	✓	✓	✓		✓		✓		✓
	MobiDENK	✓							✓	✓		✓	✓	✓			✓		✓	✓	✓
	m-ToGuide	✓	✓		✓	✓			✓	✓		✓	✓	✓		✓	✓		✓	✓	✓
	PinPoint	✓	✓	✓					✓				✓	✓	✓		✓	✓	✓		✓
	Sightseeing4U	✓		✓		✓		✓	✓			✓	✓	✓	✓		✓	✓	✓		✓

Legend: ✓ ... explicitly supported; (shaded) ... not explicitly supported

Table 2. Comparison of Context Characteristics

3.2.4 Gulliver's Genie

Gulliver's Genie [GrHa04], [HrHL03] is a prototype stemming from the areas of *artificial intelligence* and *agent systems* focusing on intelligent content delivery considering the tourists location and needs during their trip (T.LC) in a proactive manner. The location of the user is determined through GPS (T.P). Gulliver's Genie presents individual tourists (T.SA) a map (T.M) with their current position and personalised multi-media content about various cultural attractions. The assembled information includes overall sight information, follow-up links to more detailed information as well as images and audio files (T.K). These are pre-cached on the device over a wireless network connection. As soon as the tourist actually reaches one of them, the presentation of the certain POI is displayed on the screen including overall sight information, follow-up links as well as images and audio files. Additionally, the system provides the feature to allow the user to add personal comments - so called hotspots – to the information presented to tourists, which can further be shared with other tourists (T.SA). The system is realised as a *multi-agent system* using Java. On the server, a DB2 database manages the multimedia-, geo-spatial data (T.M) as well as the user model. The client application runs on a PDA, equipped with GPS, electronic compass and wireless communications support.

Gulliver's Genie considers location, in terms of the user's current position, the orientation and movement as well as user and device context (C.P). A GPS receiver provides the user's current location and an electronic compass provides the user's orientation automatically (C.Au). These context information is dynamically (C.D) monitored by an agent. The physical location is abstracted in that a set of GPS readings can be associated with a particular tourist attraction and that movements are only considered if more than 20 meters have been covered (C.Ab). In case of a considerable movement or change in the user's movement, e.g. from walking to standing, an update of the user's last position is requested (C.M). Another dedicated agent manages user context comprising demographic data such as age, gender, language and nationality together with an interest profile containing preferences about literature, art, etc (C.P). After once initialised by the user (C.Av) the profile which is stored in a database (C.R) is dynamically (C.D) and automatically (C.Au) updated by an observation agent. Additionally to the history of context also future context is considered by anticipating what the tourist will visit in the future (C.C). Validity of context is considered with respect to location information by performing a "Quality of Service audit" on the location sensor data received (C.V). Gulliver's Genie does not offer functionality to extend the considered context information (C.E).

The adaptation offered comprises several steps (A.C). An electronic map is displayed on the PDA with the user's position and orientation highlighted. Based on the prediction of the user's future location and personal preferences a new multimedia presentation is assembled for each POI the user is currently near (A.O) (A.Ef). Whereas the generation of the sight information affects the content level (A.L), the adaptation of the follow-up links affects the hypertext structure (A.L). Presentation concerns the adaptation with respect to the client's device (A.L). The sight information of the POIs in proximity is pushed automatically (A.Au) to the client for pre-caching and is kept up-to-date dynamically (A.D) which is presented on approaching one of these POIs requiring no user input (A.T). Since the presented information is always assembled from scratch (A.I), adaptation takes place on a macro level (A.G) affecting all content data including text, audio, video, pictures, links, maps, etc. (A.El). Introducing new adaptations operations is not possible (A.Ex).

3.2.5 LoL@

LoL@ [AKMP02], [Mich02] – short for Local Location assistant - is a research prototype of a location-based mobile application for GPRS/UMTS, providing tourists with multimedia tourism information about the city of Vienna. Its main focus is on supporting individual tourists (T.SA) during their trip (T.LC) with predefined tours, information about POIs, routing functionality and multimodal interaction (e.g., speech control) (T.K), based on a map (T.M). Virtual visits allow some pre-trip preparation and a tour diary allows the user to review the sightseeing tour after the trip (T.LC). A GPS sensor or telecommunication network technologies are used to get the user's position (T.P). Textual information resides in an SQL database and multimedia content is stored in the file system of the content server. An external map server provides routing and map preparation functionality. The application is designed as client/server architecture using Java Applets and Java Servlets together with XML-technology for content preparation. For communication with the server, a permanent connection via UMTS or GPRS is assumed, based on the HTTP protocol.

The system is limited to location context (C.P). The device (C.P) is not considered as a context property since the application is dedicated to mobile phones only. In addition, user context is currently also not considered apart from some simple settings. It is envisioned to support pre-filtering of virtual visits based on user preferences (C.P) as well as sorting tour details according to time context (C.P), e.g. opening hours. LoL@ does not envision to provide extensibility (C.E). Physical location context is identified dynamically (C.D) on bases of the cellId or GPS coordinates and automatically (C.Au) requested from the mobile telecommunication provider or GPS sensor each time the user issues a request (C.M). In case that automatic positioning does not allow a unique assignment of a POI to the physical location or the user has just started the tour, the physical context can be supplemented with manual user input (C.Au) indicating the current street the user is located in. This hybrid approach also relaxes the issue of inaccurate location identification (C.V) and possible unavailability of location context if positioning is turned off (C.Av). Chronology of location context is not supported (C.C). Location context is made available in a reusable (C.R) and abstracted (C.Ab) way by a separate component of the system which offers functionality for location estimation and control.

Highlighting of the user's current position on a map and a routing functionality pointing the way to the next point of interest in the tour are adaptations that are offered with respect to location context (A.O). In this way LoL@ adds new context-aware information (A.Ef) to the system in form of text, voice audio elements, images, links and maps (A.El). These are indecomposable monolithic (A.C) operations affecting the content and hypertext level (A.L). Although LoL@ provides the prerequisite for device adaptation through parameterized style sheets it demonstrates this ability for one dedicated restricted device only (A.L). LoL@ is fixed to those adaptations only and cannot be extended (A.Ex). To avoid overlapping several POIs are aggregated and represented on the map through an own symbol (A.Ef, A.L). The adaptation operations dynamically (A.D) adapt larger parts of the system (A.G) from scratch (A.I). The user can manually influence the adaptation by some basic settings (A.T). The adaptation itself is semi-automatic using automatic user positioning and user interaction to indicate the arrival at a certain route segment and initiating the retrieval of the next routing segment (A.A).

	Kind of Adaptation							Subject of Adaptation											Process of Adaptation						
	Operation	Extensibility	Effect			Complexity		Level			Element						Granularity		Tasks	Automation			Dynamicity		Incrementality
Approach			add	remove	transform	simple	complex	content	hyperbase	presentation	text	audio	image	video	link	map	micro	macro		automatic	semi-automatic	manual	static	dynamic	
COMPASS	✓		✓	✓			✓	✓	✓	✓	✓		✓			✓		✓		✓				✓	
CRUMPET	✓		✓	✓	✓		✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓				✓	
The GUIDE syster	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓	✓	✓		✓				✓	
Gulliver's Genie	✓		✓				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓				✓	
LoL@	✓		✓		✓		✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓		✓			✓	
MobiDENK	✓		✓	✓	✓		✓	✓	✓	✓	✓		✓			✓	✓	✓	✓	✓				✓	
m-ToGuide	✓		✓	✓			✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓				✓	
PinPoint	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓		✓		✓				✓	
Sightseeing4U	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓				✓	

Legend: ✓ ... explicitly supported; (shaded) ... not explicitly supported

Table 3. Comparison of Adaptation Characteristics

3.2.6 MobiDENK

MobiDENK [KrBB04], [BBKK04] – German acronym for mobile monuments - provides individual users (T.SA) both navigation support and up-to-date information about POIs. The visualization of the own position as well as of the locations of the POIs enable the user to get a quick overview of the existing monuments. The map which is available either as street map or orthophoto as well as POI data can alternatively be stored on the mobile device itself or be dynamically loaded via a wireless network connection from a GIS server through an OGC Web Map Service (T.M). In the latter case, active caching of content is not supported but the device can re-establish the network connection in case of a network interrupt. Information about the historical buildings is presented in multimedia form including historical images, which allow for comparing the historical with the current view (T.K). MobiDENK uses GPS as location sensing method (T.P). It supports users while they are sightseeing, there is no pre- or after-tour support (T.LC). It is built on the modular *Niccimon* platform. The platform is implemented in Java and requires a Java VM running on the client. The platform consists of various system components, which allow for rapid development of mobile applications.

MobiDENK makes use of location context only (C.P). The location module of the *Niccimon* platform receives dynamically (C.D) and automatically (C.Au) the physical location in form of GPS coordinates by a GPS sensor device. In addition to the GPS coordinates, the location module derives from the user's physical location values for speed and movement direction as logical location context. For getting speed and movement direction the context history is considered in that GPS signals are observed over a longer period (C.C). Position quality allows for receiving valid information for the context (C.V). Location context information is further refined and encapsulated in a *Niccimon* position event (C.R), thereby gaining abstraction from the raw positioning sensing (C.Ab). If there is no valid GPS signal available, the user can determine his position by marking it on the map (C.Av). The Mediator component of the *Niccimon* Platform disseminates context in form of events to all subscribed modules (C.M). Context extensibility is not regarded (C.Ex).

The application's adaptation operations comprise the displaying of a map with both user position and POI symbols (A.O). The varying quality of the user's location is addressed by adapting the position icon's visualization corresponding to the changing quality of the location information (A.Ef) thus making the location context's validity visible for the user. To support user orientation, the path of the user walking around in the environment is visualized by the system (A.Ef, A.C). POIs are grouped along different topics, specified by the system which can be selected in turn by the user (A.T). Apart from that, the user has no influence on the adaptation process, which is dynamic (A.D) and initiated automatically (A.A). The adaptation operations comprise adding, removing and transforming (A.Ef) focusing on elements of the content level such as text and images (A.El) and of the hypertext level (A.L) such as hypertext documents (A.El). MobiDENK supports both micro- as well as macro-adaptations (A.G). In principal, a POI can use any available multimedia format for its presentation (A.L). Adaptation is done in a non-incremental way (A.I).

3.2.7 m-To Guide

The *m-ToGuide* prototype [Kama03] has been developed within a European IST project promoting the use of 2.5/3G cellular networks with location-based services. Being explicitly designed for city travellers, m-ToGuide directs individual tourists (T.SA) via GPS (T.P) and offers location-specific multimedia information about POIs. The user can choose from different tours (museums, exhibitions, themed tours, etc.). The route as well as tour-related POIs are displayed on a map. m-ToGuide also allows for transactions (e.g. buying a ticket) by integrating external service providers (T.K). The system offers the user both street and picture maps, but it is not stated if an OGC service is used (T.M). Furthermore, it provides not only navigation support while on tour but also pre-planning a sightseeing trip by offering the tourist virtual tours of e.g. museums and after tour support by offering a personal diary function (a so called scrapbook) that allows the users to record personal impressions (T.LC). The prototype is implemented as a standalone thin client using Microsoft's .NET framework. It runs on a specific handheld device using GPRS network connection. XML technology is used to integrate the content from different content providers and exchange content between the client and server.

m-ToGuide considers location, network and time context as well as user context (C.P). Physical location information is either derived automatically (C.Au) from a GPS device and continuously updated (C.D) or entered manually (C.Au) by selecting the current position on the map, if the sensed location is incorrect or not available (C.V), (C.Av). Concerning logical context information, the system assigns the user a default personal profile (C.R, C.Ab) according to the kind of trip (e.g. family trip, business trip) the user indicated to be interested in. By mapping the user's specific travelling purpose to the preferences in the profile, those POIs the user might be interested in are selected. These preferences are dynamically (C.D) and automatically (C.Au) updated by tracking user behaviour (C.C) or further refined through user input (C.Au). Availability of location and network context (C.Av) is regarded by informing the user about location information updates and the current network status. The time context which is entered manually (C.Au), is used by the system for planning the length of a tour and opening hours are used to inform the user automatically (C.Au) when a POI is visitable. Finally, context is accessed both in a pull-based and in a push-based manner, in that location-based notifications are shown if the user approaches a POI (C.M). The context can not be extended (C.E).

Adaptations include filtering of content, making tour proposals, creating navigational routes and highlighting (A.O). The filtering adaptation operation is used to deliver information about POIs dynamically (A.D) in form of proactive tips. It affects the content level (A.L) by adding or removing (A.Ef) content elements such as text, audio, pictures, links and parts of the maps (A.El). The user can activate a tour (A.T), which is generated by the system (A.D) but allows for some refinement such as removing certain POIs. Therefore, also the hypertext level is subject to adaptation since POI symbols are links to more detailed information (A.L). The route is then recalculated and saved as a new personal tour (A.Ef), which is a complex process (A.C). In addition, the user can make further adjustments (A.T), amongst them selecting an interface or content language thus allowing for adaptations on a macro level (A.G), or toggling street names on and off, which takes place at the presentation level (A.L). Route calculation takes into account the means of transportation, e.g. by foot or by bus. Highlighting is a feature, which is used to display POI information, the user’s position and the route on the map and affects the presentation level (A.L). The adaptation process (A.C) is conducted automatically (A.A) giving the tourist no other influence on the adaptation process than by the few user preferences (A.T) mentioned above. The extension of built-in adaptation operations is not foreseen (A.Ex). Adaptation is done from scratch (A.I), since the system has to recalculate the information displayed on the device each time one of the context properties changes.

Approach	Mobile maps	Positioning	Tourist life cylce			Social Activity		Kind of Tourism information		
			before trip	during trip	after trip	individual	group	POIs	Tours	Navigation support
COMPASS	2D vector-based maps, external GIS	GPS, mobile network		✓		✓	calling friends	Multimedia information (text, images) about POIs such as museums, architectural buildings, etc. These are marked on the map as symbols. The POIs can provide additional services, e.g. reserving a table in a restaurant.		user position on map only
CRUMPET	2D vector-based maps, OGC Web Map Service	GPS		✓		✓		Multimedia information (text, images, videos) about POIs such as sights, restaurants, hotels in form of a recommendation service or pro-active tips. POIs of the selected category are displayed on the map as symbols.		shortest path between user position and POI
The GUIDE system	2D bitmaps	WLAN	search function to deliver POI information not bound to the location	✓		✓	virtual stick-on notes, context sharing, messaging service	Multimedia information (text, images) about POIs such as attractions, famous town houses, museums, etc. POIs are not displayed as symbols on map.	tailored city tours	textual instructions
Gulliver's Genie	2D vector-based maps	GPS		✓			hotspots	Multimedia information (text, images, audio, video) about POIs such as attractions, museums, churches, cafés, etc. is pushed in form of pro-active tips on the client device.		
LoL@	2D vector-based maps, GeoMedia Map Server	GPS, mobile network	virtual visits	✓	tour diary	✓		Multimedia information (text, images) about POIs such as attractions, museums, churches, cafés, etc. All POIs of the tour are displayed on the map as symbols.	predefined tours	textual and voice routing, enhanced with landmarks
MobiDENK	2D vector-based maps, OGC Web Map Service	GPS		✓		✓		Multimedia information (text, images) about POIs of monuments and historic sites. The user can select different thematic layers. All POIs of the selected layers are displayed on the map as symbols.		user position and path displayed on map
m-ToGuide	2D vector-based maps, external GIS	GPS	virtual tours	✓	scrap-book	✓		Multimedia information (text, audio, images) about POIs such as attractions, museums, etc. Those POIs are displayed on the map, which belong to the selected tour. The system also provides pro-active tips as well as a booking service, e.g. for tickets of a concert.	special tours (museums, exhibitions, themed tours), can be refined by user	navigational route from user's location to any other location, route is displayed on map and additional directions are given
PinPoint	2D vector-based map	GPS		✓		✓		Multimedia information (text, images) about POIs. These are presented to the user as a list of links.		user position on map only
Sightseeing4U	2D vector-based map	GPS	virtual visits	✓		✓		Multimedia information (text, images, audio, video) about POIs. These are grouped into categories such as museums or churches, which can be selected by the user. All POIs of the selected layers are displayed on the map as symbols.		user position and path displayed on map

Table 4. Comparison of Common Characteristics of Mobile Tourist Guides

3.2.8 PinPoint

PinPoint [Roth02] is a framework for developing context-aware web applications based on an external architecture. A web-based mobile tourism guide is implemented as a prototype using the framework. PinPoint is based on a client/server architecture, which assumes a permanent wireless connection. It runs on notebooks and PDAs and supports individual tourists (T.SA) during their trip (T.LC) by presenting them a map (T.M) of the near area as well as a list of POIs (T.K). A GPS sensor is used to get the user's position (T.P). The client hosts a web proxy and a so-called *Context Manager.* The proxy observes the data streams between browser and server and replaces dedicated tags with the corresponding context information.

Context data is provided by the Context Manager, which collects relevant context data such as user information or location. It is completely decoupled from the Web application and runs as a background process on the client. In addition, the Context Manager can query an external *Semantic Server* for geo-coding functionality. It can be looked up via a service discovery protocol, via broadcast messages or with the help of the DHCP protocol. The server runs as servlet providing for map calculations and storing the GPS coordinates of the POIs as well as the URLs of the corresponding web pages. The prototype implementation runs only on devices which support full Java. It is envisioned to transfer the PinPoint system to small systems supporting only Java Micro Edition. Each time, the browser requests or refreshes the map, a URL with some context-tags as parameter, e.g. the GPS position, is sent to the server. As a result, the browser displays a map of the user's near area as well as a list of interesting sights in the local area. The system provides zooming functionality such that the map can be displayed at two different levels of detail. On this map the user's current position is visualized.

The Context Manager considers the context properties location, time and device (C.P). Physical location information is provided in form of GPS coordinates sensed by the Context Manager. Precision of the GPS signal is mentioned to be measured

(C.V), but not further dealt with. The application can query a Semantic Server converting geographical coordinates into logical locations and vice versa (C.Ab). If no GPS signal is available, the coordinates may be derived from an alternative positioning system, if necessary, with the help of the Semantic Server (C.Av). The time context is considered in form of the current date, the time of day as well as additional information to the location context thus making available the age of the last measured location. Device context provides information about the type of device, the screen resolution and the capability to display colours. User context is considered only in form of login name and email address thus it is used rather for authentication purposes than for adaptation. It is not stated which technology is used for storing these context properties. The context properties location and time are considered dynamically (C.D) and provided automatically by the context manager (C.Au) whereas device and user information are determined in a static way (C.D). Context chronology (C.C) is not regarded by the system and there is no possibility to add further context properties (C.E). Context information is accessed in a pull-based manner either by the user making a request or by a JavaScript procedure refreshing the map regularly to visualize the current location while on the move (C.M). Since all context information is hosted by the Context Manager, it can be used by arbitrary web applications (C.R).

Adaptation is processed by analysing the context tags, which are either attached to a HTTP request or integrated in web pages and by performing the appropriate adaptation. The adaptation operations include displaying the current user's position on a map and presenting a list of interesting sights in the area correlating with the user's location (A.O, A.C). The subject of adaptation comprises all three levels (A.L), thereby focusing on elements such as text, images, links and maps (A.El). First, the POI data is filtered (A.Ef) according to the user's location (A.L). Second, the links leading to pages with more information about the POIs are composed (A.L). Third, the presentation is adapted according to the device's capabilities (A.L). In case of mobile devices with a small screen size, the amount of information is not only reduced, but a new map, suitable for the small screen, is generated (A.Ef). Therefore, adaptation can be complex (A.C) and concern various parts of the web application (A.G). Adaptation tasks are not separated (A.T) and the adaptation process is done automatically (A.A) and non-incrementally (A.I). The tags are processed on the fly as soon as the client makes a request or a page is transferred from the server to the client (A.D). Adaptation tasks are not supported (A.T). Extensibility of adaptation is not envisioned (A.Ex).

3.2.9 Sightseeing4U

Sightseeing4U [ScBo04] is a prototype[6] of a personalised city guide. Its foundation are the *mobileMM4U* framework and the *Niccimon* [BBK+04a] platform. The *mobileMM4U* framework provides generic components for developing applications providing the dynamic creation of personalized multimedia content for (mobile) devices. The *Niccimon* platform provides functionality for location-aware services and consists of several modules offering mobile navigation and orientation support, a multimodal user interface and location-based information (such as the location of the user, e.g. through GPS (T.P)) and services. It supports individual tourists (T.SA) during their trip (T.LC) by offering a street map (T.M) of the city of Oldenburg with thematic overlays, such as clickable POIs, containing information in form of text, video and audio (T.K). Since the access to POI information is not limited to the user's location, it is also possible to receive POI related information before starting the trip (T.LC). Map functionality such as panning or zooming is not supported. Sightseeing4U is based on a client/server architecture with a mobile device acting as client that receives up-to-date multimedia presentations from a server over a wireless communication network and renders them using a software player for standard multimedia formats. It is possible to pre-generate multimedia presentations on a desktop PC and copy them to the mobile device so that they can be used as a fallback during network connection losses. Parts of the framework could also be executed on the mobile device allowing to generate multimedia presentations using the minimal media assets previously copied to the mobile device.

Sightseeing4U especially considers the user and device context (C.P). Both are acquired once (C.D) at application start-up through a dialog which also can be used for later refinement (C.Au), (A.Av), (C.V). Concerning personalisation, users can state their interests by selecting several thematic layers, e.g. churches, museums and choosing their preferred language. Device context is collected by selecting output formats and target platforms from a list of different presentations. Besides user and device context, the underlying framework supports other context properties such as location and time, weather or noise information (C.P). In this respect, location context is provided by the *Niccimon* platform. This layered framework offers unified interfaces for abstracting from the actual access to context, allowing to integrate existing context profiles (e.g., CC/PP) and new context properties (C.R), (C.E). A uniform internal representation is used to make context available to the application

[6] http://www.offis.de/projekte/projekt_e.php?id=56

(C.Ab). Chronology of the context is not regarded (C.C). Access to context is performed in a pull-based manner (C.M) at each user request.

The Framework provides built-in, domain-independent adaptation operations (A.O) which can be aggregated to form new complex, probably domain-dependent adaptations (A.Cx), (A.E). The subject of adaptation comprises all levels (A.L): First, multimedia elements such as text, links, video and audio elements (A.El) are selected with regard to the user's interests and device context. This is realized by matching the user's interests with meta data associated to the POIs. The composition of the multimedia elements in time and space as well as generating POI symbols as clickable links comprise the second step. Third, they are transformed into different output formats for mobile devices with respect to the mobile device's capabilities, such as the screen size. Besides HTML and FLASH, the system offers special support for the mobile versions of SMIL (Synchronized Multimedia Integration Language) as well as SVG. The adaptation process comprises different tasks such as scaling the map image to fit on the display of the mobile device and recalculating the positions of the POI symbols on the map. The system recognizes changes of user preferences and adapts its appearance by adding or removing POI symbols on the map. Thus, adaptations encompass adding, removing and transforming certain parts of the application (A.Ef). As for granularity, mobileMM4U supports both micro- as well as macro-adaptations (A.G). All adaptations are performed automatically at runtime (A.A), (A.Dy). Separation of adaptation tasks (A.T) and incremental adaptation are not supported at all (A.I).

3.3 Lessons Learned

In this section, the results of the comparison are summarized by pointing out the major issues of the approaches surveyed by reporting on lessons learned.

Tourism as a social activity not considered. Social factors are important in tourism since people tend to go sightseeing in groups. Some systems provide limited support by offering so called "friend-finders" or the possibility to add personal information to POIs and share this with other people. However, social activities comprise more functionalities than just displaying positions of personal information. Future systems should consider communication technologies, e.g. the integration of instant messaging services.

Social zoning of cities not regarded. Tourists often tend to head towards an area to find a specific type of attraction (such as a restaurant) without having a specific one in mind (e.g. the "Dragon Restaurant" in China Town). In case of searching a restaurant, they might explore the city centre which usually accommodates many restaurants. This search strategy is the so called social zoning of cities. Although it is not the perfect way to find particular amenities, walking around exploits the tendency of certain facilities to be clustered in particular areas [BrLa05]. Mobile tourist guides should visualize those identified clusters by marking the social structure of the city [HuZi03]. For example, only the most relevant attractions of a specific zone could be presented on the map as symbols, while all salient streets can be highlighted in addition. Although most of the evaluated tourist guides show the location of attractions on the map, none of them supports the concept of social zoning.

Balance between thin and thick clients problematic. Limited mobile devices, which suffer from few CPU power and memory, should act as a thin client, i.e., all application logic resides on the server-side and the client just renders the presentation through a browser. In this case, however, a permanent network connection is required which might be problematic due to well-known problems such as loss of network connection or limited bandwidth. In contrast to that, a thick client is capable of processing content data and composing a presentation on the client side, calling for appropriate computational power. When caching technologies are applied, a thick client can overcome the before mentioned network limitations [BoKS04]. Caching of pre-selected and pre-composed data with high user relevance guarantees a fast access when actually needed by the user [GrHa05]. Thus, a careful balance is necessary between both approaches. Only Sightseeing4U supports both thin and thick clients.

Potential of incorporating external content not exploited. Most of the evaluated approaches are based on an internal customisation architecture, using their own repository for storing content data. Only COMPASS and CRUMPET exploit the possibilities of incorporating external content via Web Services.

OGC standards for exchanging geospatial information widely ignored. Most of the evaluated approaches receive geospatial information from a GIS server using a proprietary interface. Only MobiDENK use the OGC Web Map Service. Apart from the Web Map Service, the OGC (cf. Section 2.2) delivers a full range of other standards for querying geospatial data. The Web Feature Service (WFS) is an OGC standard to access and retrieve geospatial data elements from a GIS Server, which are then transformed in the Geographic Markup Language (GML), an XML-based OGC standard to transport and store geographic information. The OGC OpenLS (Open Location Service) initiative provides access to core location services including route planning, location utility, presentation and directory services.

Time and Network context are seldomly used. For a comprehensive adaptation of the application to the user's situation, a full range of context properties should be considered. All approaches provide location-based services and use satellite or networks sensing (WLAN, GSM etc.) based technologies to determine the current user's position. User preferences play an import role, whereby often dynamic adaptation of initial user profiles is done by monitoring the user interaction with the system. As a positive effect, usability of the system is increased by reducing user interaction. Although time is an important property in the tourism area, it is widely neglected by the evaluated approaches. Only m-ToGuide uses time in an effective manner by informing the user about opening hours or planning the length of a city tour. It is surprising that also network context is hardly considered. Especially in mobile environments with network fluctuations, the network context would be very important.

Potential of combining context properties not exploited. Most approaches treat the information obtained by the different context properties independently and do not aggregate them and generate new context information. They just combine location and user context allowing for a better adaptation of the system to user requirements. For example, COMPASS uses location as the criterion to filter POI data according to the user's position. A recommendation service then assigns scores to the POIs indicating the predicted relevance of the POI for the user. A combination using more factors would enable the system to react even more appropriately to the current user's situation. In its simplest form this includes the fusion of different sensors to enhance the value of a single context factor. For example, by collecting the positioning data from more than one positioning module, e.g. GPS, WLAN or UMTS, it is possible to calculate a more accurate position [NoSP02]. Of even greater value is the more complex way of combining multiple diverse context factors which may result in a model that better characterizes the situation of the user [GeSB02]. Further context can be derived by analysing and abstracting from the sensor data. Given the speed of the user and the geographical properties of his/her location (city street, highway, etc) the application can infer whether a user walks or drives [SePK04].

Context Chronology not widely supported. The majority of current approaches use historic contexts for updating the user profile only, future context is not supported. Extrapolating future context would allow to proactively pre-cache the appropriate content data on the user's mobile device based on anticipated user location, thereby addressing the fluctuating network connections as mentioned above.

Context availability partly regarded. Context availability allows the system to determine whether a certain context property is currently available. In a mobile, outdoor environment, location and network context are at a high risk of being not available. Some evaluated systems, e.g. m-toGuide, provide means to indicate the location, e.g. number of satellites used, and network status, e.g. network quality, by symbols on the screen. In this way, usability is improved by informing the user about the system's possible dropouts. If using GPS for determining the user's position, the accuracy of the position information can vary due to disturbances in the atmosphere or fluctuations in the satellite orbit. LoL@, for example, addresses this problem by supplementing the physical position information with manual user input indicating the current street the user is located. None of the approaches uses a combination of different location sensing technologies, such as assisted GPS (AGPS) [Swed99].

Proprietary representation of context data. All evaluated approaches store context data in proprietary formats, preventing an exchange of context information between systems. Sightseeing4U is the only approach, which provides so-called User Profile Connectors for accessing user profile information in different formats (CC/PP, user profile DB, etc.) and stores them in an internal data model. Representation of context data in a standardised format (e.g. XML) would facilitate exchanging of context data with other applications, (e.g., cf. [KrBo05]).

Varying automation of context acquisition. Automatic context acquisition is predominant for location context. In contrast, user context is usually derived in a semi-automatic way, in that the user fills out an initial profile, which is further updated by the system. Different to that, m-ToGuide assigns the user a default personal profile according to the kind of trip (e.g. family trip, business trip) the user indicated to be interested in. By mapping the user's specific travelling purpose to the preferences defined in the profile, those POIs the user might be interested in are selected. These preferences are dynamically and automatically updated by tracking user behaviour.

Push-based access to context not widely supported. All evaluated approaches employ pull technologies for accessing context data, only some of the approaches realize push-based access too.

Dynamic adaptation of guided tours not provided. Several approaches offer a “touring guide” function, i.e., presenting a pre-determined tour based on user interests. Those approaches, however, do not dynamically adapt the tour if a context change occurs. If, for example, the user chooses to interrupt the tour or the weather changes unexpectedly, the tour is not adjusted to the new conditions. Ideally the system should recommend tours more suited to the permanently changing context conditions.

Extensibility of adaptation operations is not commonly recognised. Extensibility of adaptation seems not to be an important issue. Adaptation operations are not seen as self-contained, explicit built-in operations but are somehow intermingled into the application logic. Only mobileMM4U, the framework underlying Sightseeing4U which stems from the area of multimedia offers a plug-able architecture which allows for an easy integration of new adaptation operations (e.g. a CityMap Operator).

Adaptation is mostly done in a monolithic way. None of the approaches supports incremental adaptation and the adaptation is done in a fully automatic way. Only few approaches, such as m-ToGuide allow for some very basic user intervention.

Chapter 4

A Light-weight Framework for Location-based Services

This chapter presents an approach for an efficient development support for location-based services based on a generic, light-weight framework. The framework was developed with the main focus on location as context factor. Since location-based services are an important domain in the area of context-aware services, the first section (4.1) gives an overview of those services.

Location-based services are developed for mobile devices with specific constraints. Therefore, the second section (4.2) lists some design guidelines which have to be followed when designing applications for mobile devices. The third section (4.3) describes the developed framework and presents its core features, architecture and functionality. The fourth section (4.4) shows the application of the framework in the area of tourism.

4.1 Location-based Services

This section discusses some major issues of location-based services.

4.1.1 Definition

In the broadest sense, a location-based service (LBS) is any service that provides information based on the user's current location [Reic04]. Another definition stems from [LiCh04], understanding LBS as the provision of geographically-orientated data and information services to users across mobile telecommunication networks. LBS comprise different research areas and can be seen as a convergence of (1) New Information & Communication Technologies (NICTs) such as mobile telecommunication systems, location aware technologies and handheld devices with (2) the Internet as well as (3) Geographic Information Systems (GIS) and spatial databases (cf. Figure 3).

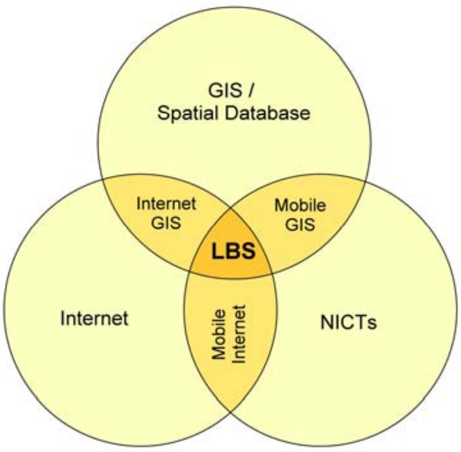

Figure 3. Convergence of Technologies for Creating LBS [Brim02]

4.1.2 Potential

LBS have a great potential since they offer a wide range of new possibilities and services for mobile users. Based on the location, those services present the user up-to date information in an individual and dynamic way and offer the potential to adapt information presentation to the current location, thus achieving a more user focused service [HaSK04]. According to [PrCW01], the successful implementation of LBS depends on following three issues:

- Exposing location-based data and GIS operations efficiently.

- Creating searching mechanisms based on semantics, rather than simply syntax or text-based searches.
- Providing a simple mechanism for content developers to create and deliver new location services, as well as to integrate location into existing applications.

4.1.3 Push- or Pull-based Approach

Location-based services can follow a push- or a pull-based approach [Koep02]. "Push" refers to location services that utilize the position of the mobile device to identify the holder as a recipient of a service and provides information in an unsolicited way. Examples are traffic alerts which notify users of the status of their travel routes or mobile advertisements. The project Ad-me [HrHa04] aims to deliver non-intrusive and personalized advertisements by overlaying these upon a mobile tourist guide. "Pull" refers to location services that provide information whenever and wherever a user issues an explicit request. Types of pull services include mobile yellow pages such as "Where is the nearest restaurant?" or travel directions, e.g. "I'm lost, where is the nearest Metro station?".

4.1.4 Elementary LBS Functions

LBS focus mainly on a few primary functions that form the basis for many other types of applications. According to [Reic03], mobile users typically conduct the following elementary mobile actions when carrying out a certain activity, e.g. planning a route to a destination.

- *Locating* (e.g. Where am I?): Providing the position of persons and objects is one of the key features of LBS and makes more complex services available. For example, locating a vehicle provides the basis for tracking and fleet management systems.
- *Navigating* (e.g. How do I get to the city museum?): Tourists are foreign in a city and therefore need instructions how to get to a certain point of interest (POI). Navigation tasks involve route services from the current user's position to an object or to a certain location. It is also possible to provide navigation functionality between any pair of objects or locations independent of the user's position, e.g. routing between two POIs.
- *Searching* (e.g. Where is the next Chinese restaurant?): LBS have to provide services for finding objects or people located near the current user's location.

- *Identifying* (e.g. What kind of building is this in front of me?): Representing objects through their geographical coordinates, e.g. N 520896 / E 5352318 is not very meaningful for users. The LBS should convert them into "semantic" locations. This process is known as reverse geo-coding in contrast to geo-coding, which is the process of assigning geographic identifiers to map features and data records. "You are in front of the Old Church" is obviously more meaningful than providing its coordinates.
- *Checking* (e.g. What takes place at the theatre tonight?): Checking deals with finding information about what happens at a certain place and is responsible for delivering object state and event information.

4.1.5 Commercial and Non-Commercial LBS

The main driver of LBS was certainly the E911 mandate of the Federal Communications Commission [FeCo] in the United States. This mandate aimed at improving the quality of emergency services by locating wireless 911 callers. In recent history, performance and capabilities of mobile devices and networks improved steadily and pushed many LBS applications to the market.

According to [Hunt01], they can be divided into user-solicited information (e.g. local traffic information, weather forecasts and local services), instant messaging for communication with people within the same or nearby localities, real-time tracking, mapping/route guidance, emergency services and location-based tariffs.

In Austria, several LBS have been created or are currently being developed. LoL@ [PoUM02] (cf. Section 3.2.5) was a research project aiming at the development of a mobile tourist guide for the city of Vienna. WLANd [WLANd] is a project for providing tourists with LBS in the city of Salzburg. The service should use mobile telecommunication networks for sensing location and provides tourists with a map and multimedia information about POIs. Tourists can document their tour during a trip by making pictures, annotations or sound files, which can be used to review the sightseeing tour after the trip. An integrated recommendation system provides the feature to rate POIs such as restaurants by assigning scores to each POI. The Vienna company Wanderman [Wand] currently develops a LBS-application which interactively presents cultural information about POIs to tourists making a boat trip on the Danube between Vienna and Budapest.

In contrast to these LBS with a commercial motive the Urban Tapestries project [UrTa] of London is a non-commercial version of a location-based wireless application allowing users to access and author location-specific multimedia content such as personal memories, pictures or short sound files that are linked to familiar locations. Urban Tapestries aims at providing a platform for ordinary people to annotate the urban spaces that they inhabit and pass through every day. By collecting their stories and make them accessible to everyone, Tapestries enables a community to become aware of their specific knowledge and memories [Jung04].

4.1.6 Future Developments of LBS

The fusion of mobile phones and PDAs, the so called "SmartPhone", together with the increased use of mobile networks technologies like HSCSD (High-Speed Circuit-Switched Data), GPRS (General Packet Radio Service), UMTS (Universal Mobile Telecommunications System) and HSDPA (High-Speed Downlink Packet Access) is expected to push ahead the further distribution of LBS. New areas of applications will be created and it can be expected that location based services emerge as the next "killer applications" of the mobile commerce [PoBS01].

4.2 Design Guidelines for Mobile Devices

This section lists some design principles which have to be followed when designing web applications for mobile devices.

4.2.1 Constraints of Mobile Devices

Mobile devices have to be small, handy, robust and easy to use to efficiently assist the user in his/her daily activities. Apart from their significant advantage of being mobile, mobile devices suffer from some limitations: usually, they have a small battery capacity, small memory and limited processing power, no permanent network connection, a small screen and no keyboard and mouse. In most cases virtual keyboards, a touch-sensitive screen or pen stylus' widen the choice of interactions.

Mobile devices have a small screen size in comparison to a desktop monitor. For example, the full screen size of a Pocket PC is not more than 240x320 px. In addition, the usable content area is even smaller because most Web Browsers have a menu bar on the bottom of the screen and a caption bar on the top which can not be used for content in the browser. Moreover, some place is reserved for the horizontal and vertical scroll bars. The usable content area in Microsoft's Pocket Internet Explorer (PIE) is reduced to not more than 229x255 px.

With regard to the bandwidth, mobile devices suffer from unstable network connections with low throughput. These limitations have to be considered during the development of the web pages. The pages have to be kept simple and small in size avoiding high resolution images and videos with a high frame rate as well as unnecessary gimmicks like colourful company logos.

4.2.2 HCI Guidelines for Mobile Devices

A careful design of the user interface is therefore a necessary prerequisite. This way a high user acceptance as well as easy usability is ensured. As for now, the discipline of Human Computer Interaction (HCI) mainly focused on traditional PC and web applications. The existing guidelines have yet to be complemented with those defining the usability of mobile devices.

The current guidelines mainly focus on stationary usage with users working on a desktop PC, thus ignoring a multitude of environmental factors that mobile users are confronted with. The mobile user also has to interact with a changing environment. After all the user is using the device in different geographic locations and during different activities. A complex and feature rich user interface may actually prove counter-productive in such situations. The user is not capable of giving full attention to the mobile system and may not be willing to fully explore the capabilities of the device to obtain the information that he or she is currently needing. The full potential of the device is therefore not exploited and the usability is reduced. For example, the user is not interested in navigating a feature rich search engine to find the nearest tourist attraction. That means that too much direct interaction is not possible in mobile contexts. A too intrusive system would be disturbing or make the parallel performance of activities impossible. The application should therefore be capable of providing the user with relevant information without much user interaction. This fact is also described by Smailagic and Siewiorek [SmSi02]: *A ubiquituous computing environment that minimizes distraction should therefore include a context-aware system able to 'read' its user's state and surroundings and modify its behaviour on the basis of this information. The system can also act as a proactive assistant by linking information such as location and schedule derived from many contexts, making decisions, and anticipating user needs. Mobile computers that can exploit contextual information will significantly reduce demands on human attention.*

4.2.3 HCI Requirements for Mobile Devices

The design principles for the development of user interfaces for mobile devices must be:

Easy to learn. The user should be able to understand the system functionality without much effort. The functional buttons should therefore be displayed in form of small icons to give the user a good idea of their functionality and make him/her familiar with the interface in a quick way.

Consistent screen design. The user interface should be consistent throughout the application. The user expects that each feature of the system works in the same way as the other features. Screen layout definitions ensure consistent arrangement of screen elements, e.g. menu or content section.

Avoid scrolling. Scrolling decreases usability since the user has to interact with the system more often to view the whole page. It is therefore important to reduce the amount of scrolling. Vertical scrolling may be acceptable by pages containing textual information. Horizontal scrolling should be avoided at all.

Non-modal interactions. Non-modal application windows are used when the requested information is not essential to continue. The window can be left open while work continues elsewhere. In contrast, modal interactions force the user to provide some information before he or she can go on using the system. The application should be developed based on non-modal interactions – the user can open a map screen and various POI information screens at the same time and can switch between them.

Provide status information. The user should be informed about current network and location status. In this way, the user can appreciate the affect of changes in connectivity on system functionality. Hiding the issue of connectivity from the user would make the system completely unpredictable and inconsistent. Predictability is a pre-requirement for the usability and affects perceived reliability of the system. If users perceive the application as unreliable, then it is likely that they do not use it [CDMF00b].

Information Retrieval. Based on the experience of the GUIDE system [CDMF00], information presented to the user should not be restricted to the current user's location. GUIDE users became frustrated when the system provided only information concerning their immediate surroundings, without allowing them to request further information. Mobile location-based systems should not be rigid in the way in which they support interaction with the user. In general, the decision about what information to retrieve should be taken by the user, not by the system. The user should have the possibility to select a POI on the map and hence retrieve information about it. This pull-based approach can be complemented with a push-based approach to extend the usability of the application. Based on the user's current position and preferences the system can proactively generate output, e.g. by pushing notifications of interesting events on the screen of the mobile device, which the user might otherwise have overlooked. Finding a good balance between these two approaches is a key challenge and enables users to choose the interaction style best suited according to their preferences: A push-based approach does not require much effort to find information whereas a pull-based approach allows for an individual and specific information retrieval [BaCK05].

4.3 The Proposed Framework

LBS have been developed with different problems in mind as to support different application scenarios, such as presenting map-based city routing. According to the evaluation results, current systems suffer from the following main limitations: First, existing approaches are often based on proprietary interfaces to other systems such as a Geographic Information System – GIS, and do not make use of already available information sources such as existing databases or web pages, thus falling short in portability and depending on time consuming content maintenance. Second, often thick clients are used, admittedly offering advantages like rich user functionality, but lacking out-of-the-box usage. Third, existing solutions are sometimes inflexible concerning configuration capabilities of the system.

To overcome these deficiencies, this work aimed at the creation of a light-weight framework for LBS. By integrating standard web technologies, interfaces for exchanging geographical information and existing data sources, it is possible to design a framework that is characterized by easy configurability and a wide area of application. Since location is the most important context factor (cf. Section 2.1.1), the main focus was on this factor. The framework can however be extended to take further contextual data such as time into account (cf. Chapter 5).

The framework provides the following core user features:

- A map to display the user's location.

- POIs superimposed on the map and classified into layers according to their topic.
- Detailed information about the POIs.

The framework makes extensive use of external data sources. Geographical information (the map) is queried from an existing GIS server using an open standard for exchanging geo-spatial data. Existing web pages serve as content source for providing additional information about POIs. When using this framework, one only has to manage the database, which stores POI specific information. In detail, this database contains the geographical coordinates of the POIs as well as the corresponding links to the web pages providing more information about them. The framework is equipped with a web-interface for populating the database with the location dependent information.

The following three subsections present details of the framework. The first subsection (4.3.1) lists its core characteristics. The second subsection (4.3.2) shows some screen templates for defining the user interface and the third subsection (4.3.3) presents the functionality of the framework.

4.3.1 Core Characteristics of the Framework

In the following an overview of the basic features of the framework is given.

Support for LBS. The framework supports the creation of LBS tailored to the user's position and preferences for arbitrary application domains, e.g. tourism or infrastructure management. The user is provided with a map of his/her surroundings together with a set of POIs and the current position obtained through GPS.

Integration of external data-sources. Integration of existing external data sources is enabled by incorporating existing GIS servers as well as by augmenting the POIs with existing Web content (cf. e.g. [KaSR05]).

Exploitation of GIS standards. The framework uses the open OGC WMS standard (cf. Section 2.2.2) for retrieving geospatial information in form of maps. Thus, geo-information from different GIS servers providing a WMS connector can be integrated in a flexible way.

Application of a thin client approach. A thin client approach is employed allowing to run the application out-of-the-box. On the client side only a graphical, ActiveX enabled browser, an Internet connection and a GPS sensor is required.

Configuration of external content inclusion. The framework offers the possibility of configuring the inclusion of external Web content similar to [KrBo05]**,** through a Web-based interface (cf. Figure 4). First, POIs can be added, deleted and updated in the framework's repository (cf. Section 4.4.1.4). Second, for each POI a title, the location in form of geographical coordinates and an URL pointing to external Web content describing the POI in more detail, can be configured. Third, POIs can be assigned to possibly nested categories, thus forming thematic layers such as gastronomy including the sub categories of cafes, fast food, and bars. Forth, for each category the visualization of the corresponding POIs can be chosen in terms of a pictogram (cf. Annex A1).

Delete POIs

Select POIs	POI Title	POI Category
☑	Altes Rathaus	Buildings
☑	Alter Dom	Churches
☐	Landestheater	Museums
☐	Dreifaltigkeitssäule	Places
☐	Lentos	Museums
☐	Kaffeehaus Glockenspiel	Coffeeshops
☐	Hotel Wolfinger	Hotels
	Delete	

Figure 4. Part of the Web-Interface

Configuration of external GIS inclusion. In addition to configuring the inclusion of the external content in terms of POIs, the WMS request to the external GIS server can be configured again through a Web-based interface. This comprises the URL of the GIS server, the type of map (e.g. aerial photography), which area to retrieve (in terms of coordinates), the spatial reference system (e.g. Gauss-Krüger), the desired output format (i.e. JPEG, GIF, PNG or SVG) and the output size of the map in pixels.

4.3.2 Screen Layout

The general limitations when developing web pages (cf. Section 4.2.1) for mobile devices have to be considered when defining screens for the application's front end. Two different screen templates build the basis of the user interface:

- Template for Map screen
  In the map view (cf. Figure 5) the screen is composed of three elements. At the bottom, functional buttons (in form of Soft Keys) provide general application control options. (For detailed information about these buttons cf. Figure 8). In the middle of the screen, an interactive map (street map or aerial view) is shown, allowing the user to derive further functionality by clicking on a certain position (e.g. choosing a POI symbol). At the top, information about the location context's availability is presented as well as a "Preferences" button, which allows to configure additional settings. The map-view screen is basically used for navigation and orientation.

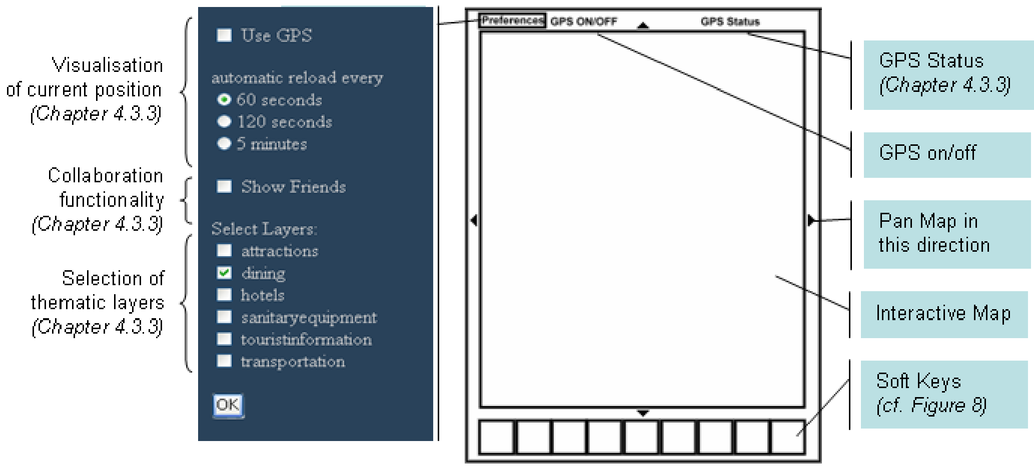

Figure 5. Map Screen

- Template for Information screen
  When the user selects a POI on the map and presses the "Info Button" (cf. Figure 8) in a second step, the information screen (cf. Figure 6) is invoked presenting detailed information of the selected POI. The screen is split up into two frames. The top frame consist of a small menu, allowing to use as much screen size as possible for the content frame beneath. The link "Web Page" links to a web page in PDA format with further information about a selected POI. The link "Guestbook" presents comments of other users about the selected POI in the content frame. By clicking on the link "Map Screen", the

user gets back to the screen which displays the map with POIs and can then select another POI.

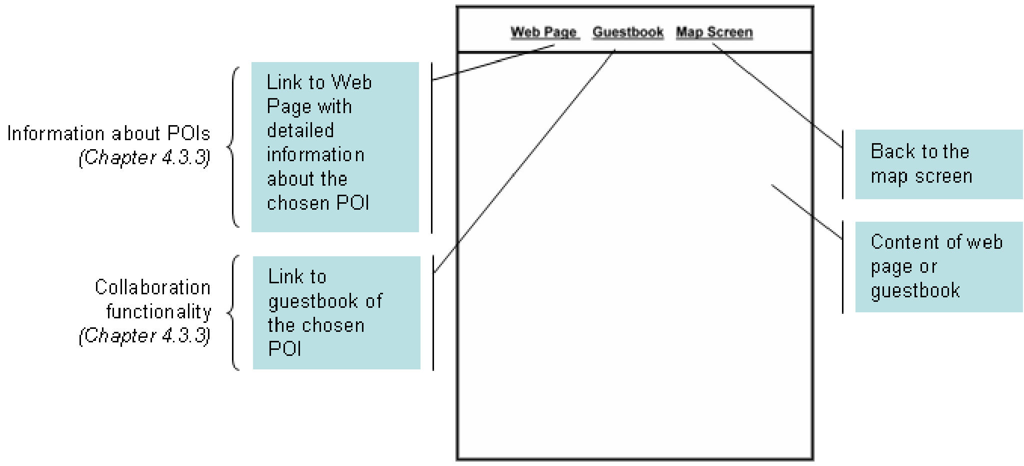

Figure 6. Information Screen

Figure 7 shows different screenshots of the prototype application (cf. Section 4.4.4), based on the two template screens mentioned above. The left screenshot shows the street map of the city center of Linz with some POI symbols. When a POI is selected (e.g. the POI "Dreifaltigkeitssäule", displayed on the map of the left screenshot through the flag-symbol), the information screen (picture in the middle) is loaded and presents detailed information. The screenshot on the right demonstrates the facility of annotating POIs with comments.

Figure 7. Different Screens when Selecting a POI Symbol

4.3.3 Functionality of the Framework

The main functionality provided for the user is described in the following. A screenshot of the system's graphical user interface is depicted in Figure 8.

Figure 8. User Functionality

The GPS receiver of the mobile device locates the user's position as x-y-z coordinates. If the user asks for information relevant to his/her current position, the system uses this location information to present an overview of the user's activity area in form of a map.

Basic map functionality. The user interface supports basic map functionalities like panning, selecting or changing the zoom level. The user can switch between a street map and an aerial photo. One advantage of a street map is that it helps getting an orientation of the surroundings by presenting important roads, paths and building blocks. On the other hand, an aerial photo is useful to give a general impression of the geography and get a basic overview over the region [Reic04].

Selection of thematic layers. Information provided to the user is organized as layers in order to avoid the small screen being cluttered. The user can select different thematic layers he/she is interested in. The corresponding set of POIs including those

POIs associated with sub-categories is then superimposed on the map through self-explaining, pictogram-like symbols at each user request. This way it is ensured that the user understands the meaning of the POIs' symbolisation in an easy manner. Thereby the user is enabled to view them in geospatial reference to each other and to build a topographical mental model of his/her surrounding (cf. Figure 9).

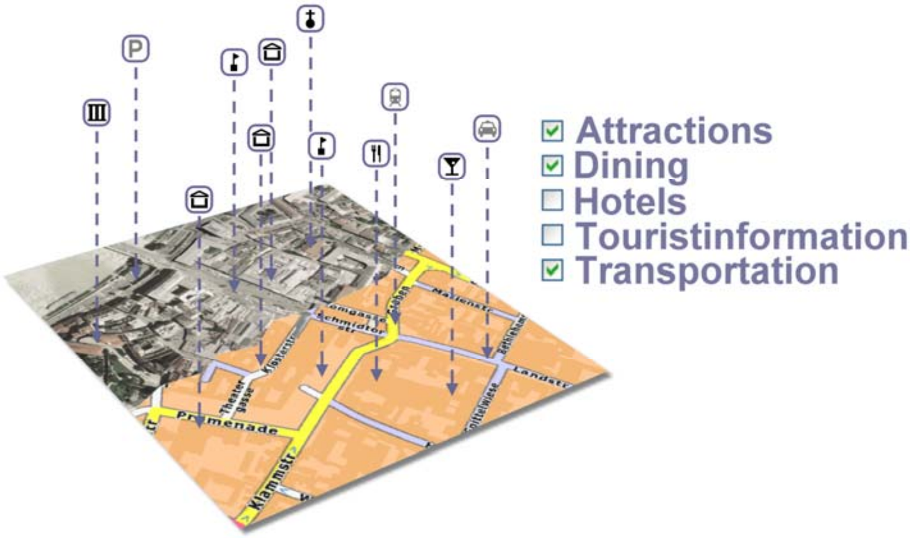

Figure 9. Thematic Layers

Information about POIs. If a POI contains more information, its symbol is displayed with full opacity, indicating that it is clickable. By clicking on it further information, e.g., historic information, opening hours or pictures can be requested (cf. Figure 10).

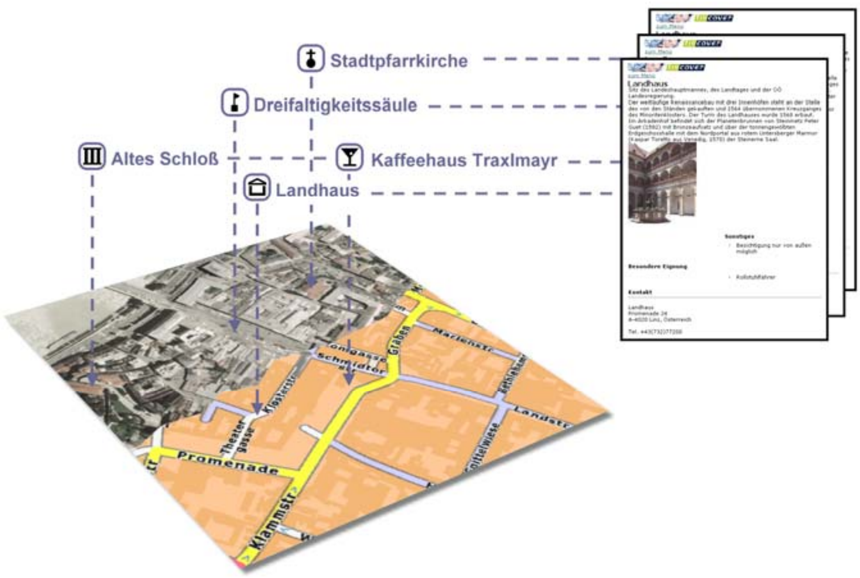

Figure 10. Inclusion of External Web Content

Search of POIs. The user is able to search POIs by indicating a thematic layer and a certain distance as search radius. Based on this information and the current user's position, all those POIs are filtered and displayed on the map whose type correspond to the selected layer and which are situated within the selected radius. If the user's location cannot be determined, the center of the currently displayed map is assumed to be the current user's position (cf. Figure 11).

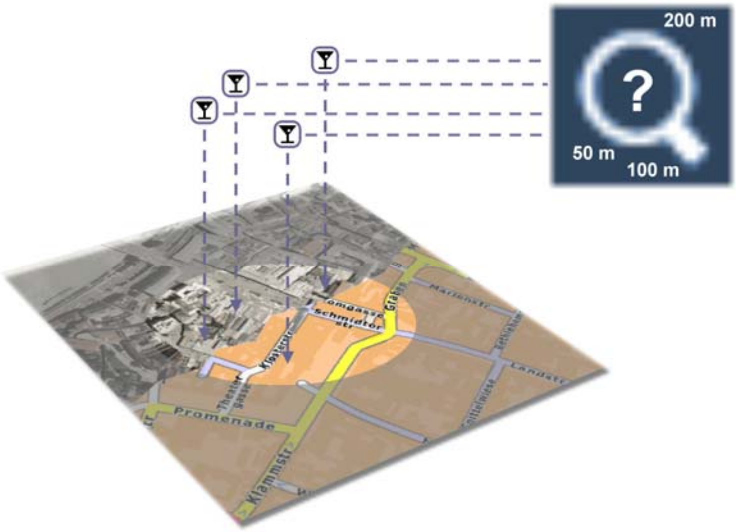

Figure 11. Display all "Bars" within a Certain Distance

Refresh of maps. The displayed map is refreshed automatically either as soon as the user performs an explicit map action, e.g. panning, or periodically to reflect the movement of the user. The user is able to adjust the refreshing period to his/her speed of movement by specifying a certain time interval after that a new map is displayed on the screen of the device (cf. Figure 5). The user also has the possibility to completely turn off this feature.

Visualization of current position. Getting a GPS signal in the historical parts of a city can be difficult. Alleyways and thick stone walls of historic buildings may block signals of satellites located near the horizon. The unavailability and changing accuracy is dealt with in the following way:

First, a circle is used as symbol of the uncertainty of the user's current position. Even if the GPS device can calculate a valid position, the real user's position can differ up to 5 meters because of disturbances in the atmosphere or fluctuations in the satellite orbit [HiBo01].

Second, the user is continuously informed about the quality of the received GPS signal. The number of satellites visible, the number of satellites which are used to fix the position and the positional dilution of precision (PDOP) value are continuously measured and displayed in the upper right corner of the screen (cf. Figure 12). The PDOP value [LaPi98] indicates the quality of the geometric orientation of the satellites being used to compute the position. The best satellite constellation is one satellite directly over the user's head and the others spread around in the four directions. The closer the PDOP value is to one, the better the geometry of the satellites and the position accuracy. In general, it should be less than six to provide a good position accuracy.

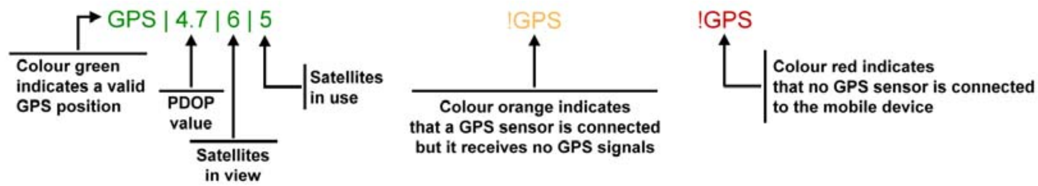

Figure 12. GPS Status Bar

Third, in the absence of a GPS signal, the centre of the currently displayed map is assumed to be the user's current position.

Collaboration functionality. Since social factors are important in several application domains of LBS, some basic collaboration functionality is provided. First, POIs provide a link to a guestbook (e.g. cf. GeekBlog [Geek]) allowing users to annotate POIs (e.g. in the tourism domain rating the quality of a restaurant) that can be shared with other users (similar to e.g. COMPASS, cf. Section 3.2.1). Second, the system offers the visualization of other users simultaneously using the system. The positions of registered friends are marked on the map, thus providing the prerequisite for subsequent interaction. So far, this feature is only realised in the system by simulating the positions of other users, cf. Section 4.4.1.4.

In this context the question for privacy and security is raised. Privacy and security refer to the following aspects, but are not further detailed in this framework. Maintaining the privacy of the users is vital so that they do not feel monitored by the system or third party providers. It is important that the users have the overall control of their private data and location information. This can be achieved by informing them of how the position information is used and logged. In general, all external services should have to request permission from the user to obtain the position. Hence, the user can grant access to the position information for one time only or for a longer period by making a contract with the external service provider. Additionally, the system should provide a function to deny access to location information in an easy manner, by overriding all given rights or contracts [NoSP02].

4.4 Implementation of the Framework

This section is divided into four subsections. The first subsection (4.4.1) gives an overview of the framework's architecture. The next two subsections define the hardware and software requirements of the client (4.4.2) as well as server-side (4.4.3). The last subsection (4.4.4) presents the implemented tourist guide LiMoG.

4.4.1 Architecture of the Framework

The client/server architecture of the framework consists of internal components in terms of a *LBS Engine* and a *Repository* (cf. Figure 13).

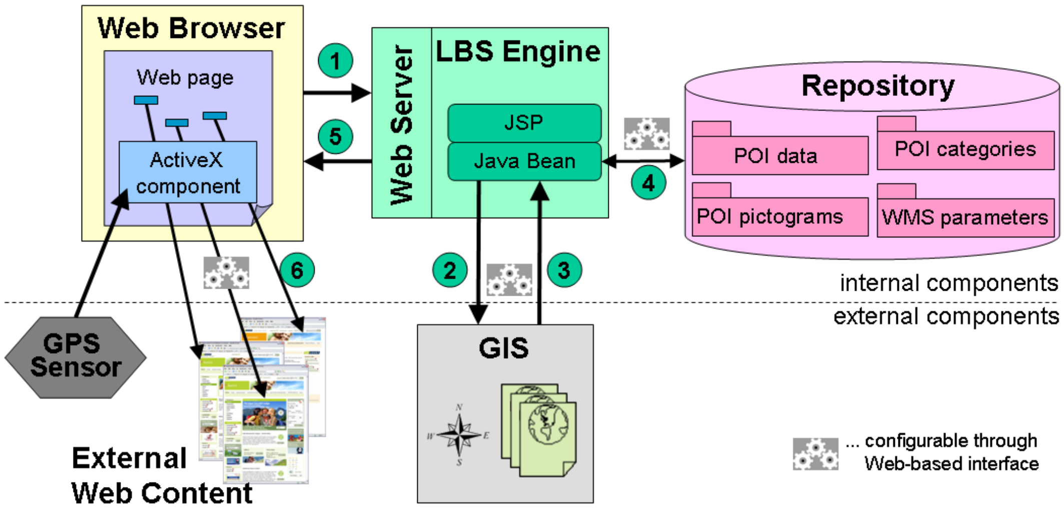

Figure 13. Architecture and Data Flow

The LBS Engine is implemented using JSP and Java Beans and realizes the core functionality of the framework. Its task is to handle the user's requests and the communication between the different components. The Repository utilizes a database (cf. Section 4.4.1.4) for storing configuration information. External components comprise a *GIS Server* and external *Web Servers* to deliver further information about POIs.

4.4.1.1 Dataflow

On each request the system performs the following six steps:

Step 1: The Web Browser communicates with the Web Server over WLAN and is continuously informed about the user's location through an ActiveX component included in the corresponding HTML-page. As the user invokes actions, e.g. clicking on a POI, a request with the GPS data as additional parameter is sent to the Web Server and forwarded to the LBS Engine for processing.

Step 2 and 3: Corresponding to the user's position and kind of request, the LBS Engine retrieves the WMS parameters from the Repository and requests geo-data from the GIS Server using the OGC WMS specific *Get Map Request*. The GIS Server processes the request and returns an image based on those parameters.

Step 4: Before being deployed on the Web Browser, the map image is augmented with information about POIs retrieved from the Repository, based on the user's preferences and location. The POIs' geographical coordinates are transformed into picture's pixel coordinates to correctly locate the POI pictograms on-top of the map.

Step 5: The user receives the image in the form of a MIME-type encoded picture, which contains the map and thematic layers.

Step 6: When the user requests further information of a specific POI, the corresponding Web page is retrieved from the appropriate external Web Server.

In the following, the internal components of the framework (the Client, the LBS Engine and the Repository) are presented in more detail.

4.4.1.2 The Client

Every mobile device with wireless connectivity, a Web Browser capable of using ActiveX controls and a GPS sensor can serve as a client. Apart from obtaining GPS coordinates from the GPS sensor, all application logic resides at the LBS Engine on the Web Server. This way a client with little memory and CPU power is capable of running the web application. GPS is an ideal location sensing method for outdoor applications since it can be used worldwide 24 hours a day and is available for free, except for the costs of a GPS receiver. Position readings from the GPS receiver are sent every two seconds in form of NMEA 0183 v2.0 sentences via the COM-Port of the Pocket PC to an ActiveX control element that parses them. NMEA (National Marine Electronics Association) [NMEA] is a standard protocol, used by GPS receivers to transmit data. NMEA information is transmitted from a 'talker' to a 'listener' device in form of sentences, with a maximum length of 80 characters. Each NMEA sentence starts with '$' and terminates with CR/LF. For an overview of the different sentences cf. [Star01]. The following figure shows an example of a GPGGA (Global Positioning System Fix Data) sentence, which contains information about the time, longitude and latitude, quality of the signal, number of satellites used as well as the altitude.

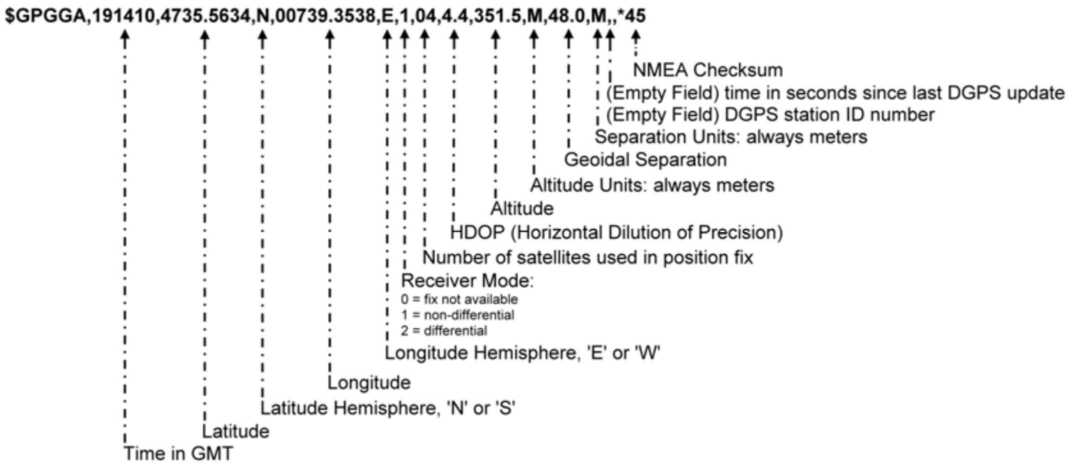

Figure 14. Example of a GPGGA Sentence

At each user request, the location as well as further parameters concerning the user's preferences are transmitted to the LBS Engine over HTTP.

4.4.1.3 *The LBS Engine*

The LBS Engine on the Web Server is the core of the framework. It accepts requests from the client as input, queries the appropriate content and invokes certain adaptation methods. Based on the user's location and preferences it gathers content consisting of spatial and non-spatial data from different data sources. Spatial data in form of a map builds up the basis information and is queried from an external GIS server using an OGC WMS request. Each time the user carries out an action concerning the map presentation, e.g. zooms in, a completely new map with the appropriate scale and level of detail is requested from the GIS server. Thus it is ensured that the image resolution of the map is displayed correctly. The LBS Engine then accesses the Repository to retrieve the POIs. All POIs that belong to the visible section of the map and to the categories selected by the user are displayed on the map. The size of the POI symbols (cf. Annex A1) is adjusted according to the selected zoom level of the map. In addition, a transformation has to be applied to the POI coordinates. In the Repository, the POI's location is encoded in the "Bundesmeldenetz M31" spatial reference system and has to be converted to the map image's pixel coordinates. The map augmented with the POIs is then transferred to the client in the form of a MIME-type encoded picture.

4.4.1.4 The Repository

The Repository holds all information about the POIs and the WMS *GetMap Request* specific parameters in a database on the Web Server. All POI data as well as the parameters for configuring a valid GetMap Request can be managed over a web-based interface (cf. Figure 4). The POIs' symbols (cf. Annex A1) are stored directly in the file system of the Web Server.

The Repository uses a HSQLDB [HSQL] database. HSQLDB (the hypersonic SQL database) is a small, open-source, relational database written in Java. Although it is not as powerful and popular as MySQL and DB2, it provides a rich set of features which can satisfy the needs of many Java applications [Loza03]. It is therefore used in many Open Source Software projects, such as in OpenOffice.org 2.0[7]. Since all HSQLDB components (the database engine, JDBC driver, server processes, etc) are provided in a single JAR package with a size about 260 kb, it needs little memory and processor requirements and can be easily integrated in Java applications [Edli03].

The database stores all information about the POIs and the parameters for making a WMS map request in six tables. For a detailed view on the database tables cf. Annex A2. Figure 15 shows the corresponding UML data schema.

The table POI_DATA stores information about the different POIs. Besides defining the name, a short description and the spatial location, some indication must be given as to what type of category the POI is assigned to. Additionally, two links can be associated with each POI. First, a link to a web page containing more detailed information about a POI, and second, a link to a guestbook allowing users to share their comments with others. Since there are many different types of POIs, they are grouped according to their topic such as monuments, churches or restaurants. A POI can belong to one or more categories, expressed through the table POI_MAPPING. For example, the "Old Castle" of Linz may belong to the category "Museums" as well as to the category "Restaurants" as it contains both a museum and a restaurant. Each category is associated with a pictogram-like symbol which is used to depict the POIs of this type on the map. (For a list of categories and symbols cf. Annex A1). Each POI can be displayed on the map through only one symbol, even if multiple categories are assigned to the POI. The boolean value "DefaultSymbol" indicates the most important category of the POI and uses this symbol to represent that POI on the

[7] http://www.openoffice.org

map. POI categories can be nested, e.g. the subcategory "Museums" belongs to the category "Attractions". The table POI_SUBCATEGORY stores all subcategories and the table POI_MAINCATEGORY stores all main categories. The table WMS_DATA stores all parameters which are necessary for making a WMS *GetMap* request. The coordinates stored in the table SIM_DATA are used to demonstrate the functionality of other users simultaneously using the system. They represent waypoints to simulate the route of another user. If desired, his/her position is marked on the map and the user can see where his/her friend is situated at the moment.

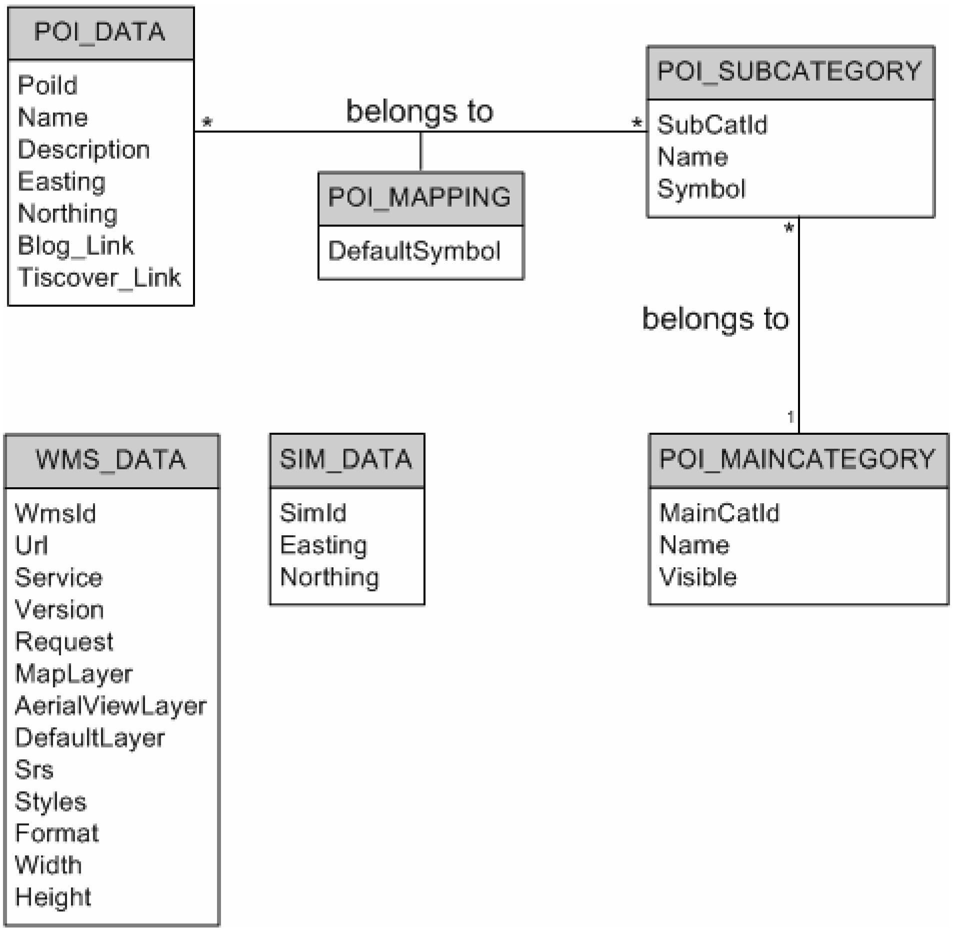

Figure 15. UML Data Schema

4.4.2 Client Hardware and Software

For demonstration purposes (cf. Section 4.4.4), a Pocket PC (IPAQ 5450) equipped with a GPS receiver (SysOn GPS CF plus II) was used.

4.4.2.1 GPS Tool

To obtain and pass the NMEA GPS sentences as well as for converting the position between different coordinate systems, the prototype uses the GpsTools ActiveX of Franson[8]. This tool enables to access NMEA data. After setting the baud rate and a COM-Port, it starts searching for a GPS connected to this serial port.

The GPS position is obtained as WGS84 (the World Geodetic System 1984) coordinates and has to be converted by the GpsTools into the coordinate reference system used by the WMS Server. In case of the prototype application, the WMS server uses the Austrian "Bundesmeldenetz" (BMN) M31 reference system. Therefore, the WGS84 position (latitude/longitude) has to be converted by the GPS tool to the BMN M31 grid coordinates (northing/easting).

The Austrian BMN M31 is based on the Mercator projection, a cylindrical map projection and uses a geodetic datum based on the Bessel ellipsoid. In general, map projections [KOWO] are attempts to portray the surface of the earth or a portion of the earth on a flat surface. Geodetic datums [KOWO] define the reference systems that describe the size and shape of the earth. Since the earth has about the figure of an ellipsoid, it can be best described using an ellipsoidal model. A position on the earth's surface is specified through its coordinates on a reference ellipsoid. The geodetic datum defines the lengths of the ellipsoid's axis as well as its origin relative to the earth's geocenter. The WGS 84 datum, for example, defines the centre of mass of the earth (the geoid) as its ellipsoid's origin, and the direction of the earth's axis as the minor axis of the reference ellipsoid. The WGS 84 datum fits perfectly to describe the shape of the whole earth, but not parts of its surface. For that reason, there are hundreds of locally-developed geodetic datums around the world, usually referenced to some convenient local reference point. Due to the influence of the Alps, Austrian reference systems are based on the datum MGI ("Militärgeographisches Institut") Austria, that better fits the geographic situation (with the Alps) in Austria than the datum WGS 84.

The GpsGate tool of Franson[8] provides a GPS simulator which is used for testing purposes during development and for indoor demonstrations. After defining a set of waypoints, it simulates a user travelling between those waypoints in the selected speed.

---

[8] http://franson.com

4.4.2.2 Web Browsers for Mobile Devices

One of the biggest limitations is the small functional range of mobile Web Browsers. Developers of web pages for mobile devices have to go without the rich functionality of existing Web Browsers for desktop PCs. To choose a mobile Web Browser that fits best for the prototype application, Microsoft's PIE was compared with two other popular Web Browsers for mobile devices, namely MultiIE and Netfront.

Pocket Internet Explorer (PIE)

PIE's functionality is still based on the desktop Internet Explorer version 3.x, neglecting the fact that Internet Explorer is now at the version 6.x. and that the latest W3C HTML guidelines are at version 4.01. PIE still implements the HTML 3.2 standard and is therefore not able to render perfectly HTML code conforming to the latest W3C standards. But there are further limitations: Apart from providing some InnerHTML/InnerText functionality, PIE cannot cope with developments in the document object model and has limited JavaScript support. In addition, it does not support Cascading Stylesheets which are important for the absolute or relative positioning of elements and which are a basic requirement for many applications of DynamicHTML. As a consequence, elements cannot overlap and positioned on each other. Figure 16 shows how PIE displays a web page with a map as basis and several POI symbols superimposed on the map at different positions, each within an own layer. As can be seen, PIE displays the three POI symbols beneath the map and not on their actual positions on the map.

Figure 16. PIE's Poor Behaviour of Displaying Nested HTML Elements

PIE has further shortcomings: It lacks Java and Flash support and it does not support a full screen or multiple windows. With these limitations in mind, the question of using another Web Browser capable of filling the gaps of PIE's missing functionality raises.

MultiIE and Netfront are two further popular Web Browsers for mobile devices. In the following, their features as well as their pros and cons are presented. Table 5 shows the main features of all three evaluated Web Browsers. Finally, one Web Browser is chosen, which is the most appropriate for the prototype application.

MultiIE

SouthwayCorp's MultiIE[9] added additional features to the core functionality of PIE. It allows for opening multiple windows using a tabbed interface comparable to Mozilla Firefox. Each site has its own tab and thus the user can easily switch between different sites whereas PIE users are forced to use the back and forward arrows to go back and forth whereby the web pages have to be reloaded again. It fully supports Windows Mobile 2003 Second Edition including VGA-resolution screens. Windows Mobile 2003 Second Edition as well as NYDITOT's Virtual Display program allow for screen rotation from portrait to landscape orientation. But most importantly, it enables to switch into a full screen mode whereby the address bar as well as the scroll bars can be hidden, thus offering the whole screen size (240x320 px at a Pocket PC). Additionally, MultiIE allows for assigning different browser functions to the hardware keys, e.g. toggle full screen. This improves usability since the Pocket PC can be hold and controlled with one hand. MultiIE offers the option of bringing a web page to the foreground when it is loaded. This feature can be useful for web applications that automatically refresh their pages, e.g. in order to update the user's position. In this way, the user is immediately informed about changes even in the case of being occupied with another software program by bringing the web page to the foreground. Another useful feature is the option of keeping the backlight on, thus being able to read the web pages without being forced of tabbing the screen each time. But there are also some shortcomings. Just like PIE, it does not address the problem of displaying web pages properly that have not been designed to be viewed on a small mobile device. In addition, in its current version it can be installed only in main memory and does not provide a sufficient help support.

[9] Version 3.1. http://www.southwaycorp.net/multiie.htm

NetFront

Access's NetFront[10] is a Web Browser based on a modular and scalable architecture allowing developers to choose only components needed for their target devices. Modules such as WML (Wireless Markup Language) or SMIL (Synchronized Multimedia Integration Language) can be removed or added according to the requirements of the application and available memory space. The user interface and application layer are decoupled from the core browsing engine and thus can be modified to suit particular requirements or customization. NetFront supports multiple windows as well as a full-screen mode and offers a built-in rendering engine which allows for displaying web pages not designed for the small screen of mobile devices. The text size is adjustable and a zoom function offers the possibility to zoom from 25% to 125 % that can be used to display details of an image. NetFront supports parts of the Cascading Style Sheets and Document Object Model, the HTML standard 4.01 as well as Java. In this way, NetFront enables access to almost all web pages of the Internet. The functionalities of NetFront can be extended by integrating external plug-in modules such as Macromeda Flash or Adobe Reader. Its features can be extended by multimedia options, including a SVG player (SVG Tiny compliant) or a SMIL player, which complies with 3GPP SMIL and W3C SMIL Basic.

Comparison of the three Web Browsers

	PIE	MultiIE	Netfront
Full screen modus	No	Yes	Yes
Multi-windows	No	Yes	Yes
Programmable Hardware Buttons	No	Yes	Yes
HTML	HTML 3.2	HTML 3.2	HTML 4.01
DOM	InnerText only	InnerText only	Part of DOM Level 1 and 2, DynamicHTML
CSS	No	No	Part of CSS 1 and 2
ActiveX controls	Yes	Yes	No

10 Version 3.2. http://www.access.co.jp/english/news/index.html

Table 5. Overview of the Web Browsers' Functionality

Netfront is the most sophisticated Web Browser offering a full range of different features. But since it does not support any ActiveX controls it could not be used as default Web Browser for the developed prototype. ActiveX support is necessary because an ActiveX control element is used to pass the GPS data from a GPS sensor.

Using MultiIE for the Prototype

The prototype was tested with the MultiIE Web Browser that provides many useful features but also some limitations, which were eliminated in the design phase.

Useful features of MultiIE for the prototype:

- By providing a full screen mode, the prototype can make use of the full display of the mobile device (cf. Figure 8).
- MultiIE's feature of programmable hardware buttons can be used to toggle between full screen and 'normal' mode by pressing a single hardware button.
- Tourists often switch between a map to see where attractions are and a guidebook to see what the attractions are. It is therefore essential to support this comparison of information by allowing users to quickly move between related pieces of information, such as a map with the POI symbols and the corresponding pages showing detailed POI information. MultiIE provides the capability of loading and displaying web pages in multiple windows at the same time. Thus, it is possible to have one window displaying the map and the others listing the information about each POI.
- If the user reads information about a POI and his/her location is updated on the map, MultiIE can be configured to bring the web page with the map in the foreground, thus informing the user about the context changes.

The limitations of MultiIE were reduced in the following way:

- By designing the user interface for the small screen of a mobile device, it was not necessary to have a Web Browser capable of rendering web pages designed for desktop PCs to mobile devices.
- By developing the framework using the HTML 3.2 guidelines, the web application can be displayed without any errors with the MultiIE Web Browser.
- Nested HTML layers which cannot be properly displayed by PIE and MultiIE are avoided.

MultiIE is therefore sufficient to be the default Web Browser for the prototype application.

4.4.3 Server Hardware and Software

Server-side, an Apache Tomcat (version 5.5) application server provides the servlet engine. The LBS Engine, realising the adaptive component of the prototype, is implemented using Java Servlets and Beans, based on the JDK version 1.5. The content, comprising geospatial data from a GIS server and POI data from the Web Server's Repository, is adapted to the user context based on information sent with the client request and transferred back to the client. The Repository's database is accessed using the JDBC:ODBC bridge. For the communication with the client a permanent wireless connection in form of WLAN is assumed.

4.4.4 Prototypical Application – Linzer Mobile Guide (LiMoG)

The applicability of the framework is demonstrated by realizing a tourist guide for the city of Linz called LiMoG - short for Linzer Mobile Guide. For the test area of LiMoG 40 POIs with attributes were captured, including e.g., historic sights, churches and cafes along traditional sightseeing tours through the historic centre of Linz[11]. POIs are divided into 6 main categories, including Attractions, Dining, Hotels, Sanitary-Equipment, Tourist Information and Transportation, but they may include additional subcategories. For an overview of the categories and their corresponding pictograms cf. Annex A1. DORIS[12], the GIS of the federal state of Upper Austria, provides the geospatial data in form of an OGC WMS. Each POI links to an existing Web page in PDA format provided by Austria's official destination information and booking system TIScover[13]. For example in case of a restaurant, TIScover provides information about facilities, cuisine, opening hours, prices as well as photos. For demonstration purposes, a Pocket PC (IPAQ 5450) equipped with a GPS receiver (SysOn GPS CF plus II) is used. For accessing the Internet, a publicly available WLAN hotspot in the city centre of Linz was utilized.

[11] Linz. http://www.linz.at/tourist/stadtrundgang.asp

[12] DORIS, Digitales Oberösterreichisches Raum-Informations-System, http://doris.ooe.gv.at

[13] TIScover Destination Management System, http://www.tiscover.com

Chapter 5

Outlook

During the development of the framework and the implementation of LiMoG, some points for improvement and extension of the framework were identified to offer additional features.

Incorporate more context factors. Giving the experience gained it is intended to extend the notion of context beyond location and user preferences to consider also other context properties such as time, network, user activity as well as a combination thereof. Including the context-factor time, for example, would allow to inform the user about e.g. opening times, by displaying the symbols of all POIs which are closed with lower opacity value to reflect the disabled status of these POIs. This shift from LBS to context-aware systems has a better chance to enhance the service's relevance for the user and to improve user satisfaction.

Incorporate pro-active behaviour. Pro-active tips are a nice feature to inform the user about interesting sights by using proximity as a trigger to push a notification on the display. For example, when the user walks across a camera store having a special offer of that camera type, he/she is interested in, the application can place a notification on the display of his/her mobile device. Pro-activity can also be treated in a broader way in that the mobile application itself becomes pro-active and thus minimizes user interaction. First, the application can discover and order services, which appear useful according to the user's context. Second, it can pre-configure them based on the user's profile and past behaviour so that the user has immediate access to them. Third, it can even pre-execute those services, which are most likely to be accessed by the user [Stüt05].

Provide ontology services to add external data sources. When the LBS has to query additional external data sources such as XML files, databases or web pages, an ontology can be useful to find a matching between the context data and the data profiles of the different external data sources to search for, e.g., information related to their geographical vicinity only [SpAY05].

Add user-defined adaptation operations. To increase flexibility of the adaptation process, users should be able to define adaptation operations on the stationary computers, which they can then include into the application. For example, the user can specify several profiles on the desktop PC, e.g. one for home use and one for work, which are then transferred to the mobile device.

Add adaptation operations for external web sources. An adaptation operation is envisioned, which can transform the content of existing web pages before being displayed to the user.

Extend adaptation to different context factors. Adaptation of content to location and user intention (e.g. select layers of interest) can be extended to reflect further context factors. Adaptation to the social context, e.g. the language, can be done by identifying the user's native language and display the content and user interface in this language. Displaying all POIs that have currently closed with lower opacity is a way to adapt to the current situation, in this case to the opening times. Adaptation to the user activity can be achieved, for example, by superimposing a thematic overlay of the activity area on the map. When the user wants to go out with friends in the evening, a "nightlife" layer can show the regions of the city comprising bars and pubs. To present these overlays, polygons or rectangles can be used. The map itself can be adapted to enhance the personal relevance of maps and thus improve its usability. Several techniques can be used, including the concept of egocentric maps [Reic05], focus maps [ZiRi02], chorematic focus maps [KlRi04] and variable-scale maps (e.g. fish-eye view) [HaSL02].

Extend functionality through integrating recommendation systems or event notification services. Users appreciate getting information from other people for own decision making. Including a recommendation system, where users can allocate scores to different POIs according to their like/dislike would allow for a better (pre)planning phase, cf. e.g. [SePK04]. Event notification services inform users about relevant events or places to visit. When a location event occurs, e.g. the user passes an interesting POI or an external event occurs (e.g. "The concert today has been cancelled"), the system triggers a certain action such as selecting information data depending on the user's location, time, his/her personal profile as well as the event history and forwards this to the user [HiVo03].

Employ generalisation methods to ensure legibility of the map. Map generalisation is a graphic and content-based simplification of the geographic data. It has the purpose of preserving legibility. For example, if POI symbols overlap each other due to the small screen of a mobile device, the user is not able to recognize the spatial relationships of the POIs. Thus, the number of POIs has to be decreased by aggregating two or more symbols and replacing them with a new one [EdBW05]. LoL@ [PoUM02], the UMTS tourist guide for Vienna introduced the concept of Regions of Interests (ROIs) to prevent overlapping of symbols. In this way, a ROI comprises various POIs in the near area and is displayed with an own symbol on the map.

Define user-relevant POIs. The mobile application should facilitate the annotation of additional, user-relevant POIs. This can happen either manually by the user by selecting a position on the map and setting a new POI, or by the system learning the user's frequently visited locations. If a user often frequents a certain location, the system can identify this location to be important for the user and can prompt the user to name it. By naming it, the user indicates to the system that it is of importance to him/her. In this way, the system can add user-relevant POIs to the generic, pre-existing set of POIs [MaSc01].

Offer pre-defined tours with routing functionality. According to the location-based tourist guide m-toGuide [Kama03], users should be provided with special tours, e.g. a Museum Tour, which they can further refine by removing certain POIs form the proposed tour. For more information about tours cf. Section 3.1.3. LiMoG users currently have to use the street map to orientate themselves. A routing service would facilitate a better support for moving between the POIs.

Provide after-tour support. Users should have the possibility to document their trip, e.g. by annotating their tour with pictures, short texts and sound files similar to m-ToGuide's "Scrapbook" [Kama03]. In this way, they are able to share their travel experiences with other people at home and reminder of their holiday trip.

Embed SVG or SMIL for presenting the map or the content. Scalable Vector Graphics (SVG) and SMIL (Synchronized Multimedia Integration Language) are both XML based W3C standards. SVG is used to describe 2D vector graphics and SMIL is used for the interactive presentation of multimedia elements.

Provide multimodal interaction and presentations. The highly mobile environment and the small screen call for additional interaction and presentation forms apart from using a stylus as input medium and a 2D map for visualisation of information. The system should provide multiple modalities for input and output so that a more intuitive user interface is reached [ReSc03]. Certain user activities ask for hands-free operation, e.g. navigating while driving a car. In this case, visual information is not adequate since the user has to concentrate on a main task. Therefore the system should allow for speech input such as speech commands by the user and speech output such as voice routing [MaZi00]. Voice routing in form of reading the routing instructions to the user, e.g. “Turn left in 100 meters”, can also be helpful when the user follows a tour and has to concentrate on looking at the street instead of the display (cf. e.g. LoL@ [PoUM02]). For the presentation of spatial information on small displays, alternative presentation forms such as 3D graphics are an important means to provide a more realistic representation of the user’s surroundings. Studies have proved that 3D maps lead to a better understanding of the topography than conventional paper maps [Dick05]. In the EU-project TellMaris [ScCL05], 3D maps were used with the result that users were able to match between the buildings on the map screen and the reality faster [ScCL05]. With more powerful devices and wireless networks, the current disadvantages of a low frame rate and few textures can be compensated. Photographs, panoramas and videos are another kind of presentation form, but they have to be carefully designed and used in the context of small displays. Anyhow, additional integration of photographs and panoramas for landmark identification during navigation help users when reaching a decision point [GaUh05].

Annex

This section contains information about the POI symbols used in the LiMoG application as well as the different database tables of the framework.

POI Symbols

In the following the POI categories of the LiMoG application and their corresponding pictograms are presented. The first table shows the pictograms of the categories “Attractions” and “Dining”, the second lists the symbols of “Hotels” and “Sanitary-Equipment” and the last one the symbols of “Tourist Information” and “Transportation”.

Category: Attractions		**Category: Dining**	
	Buildings		Bars
	Churches		Coffeeshops
	Museums		McDonalds
	Places		Restaurants

Table 6. Pictograms of Categories “Attractions” and “Dining” Including Subcategories

Category: Hotels		**Category: Sanitary-Equipment**	
	Hotels		Toilets

Table 7. Pictograms of Categories “Hotels” and “Sanitary-Equipment” Including Subcategories

Category: Tourist Information	**Category: Transportation**
 Tourist Information	 Bus
	 Railway
	 Parking
	 Taxi
	 Watertransportation

Table 8. Pictograms of Categories “Tourist Information” and “Transportation” Including Subcategories

A.1 Data tables

The database stores all information about location-dependent information (e.g. the POIs) as well as the parameters for composing a WMS map request in six different tables. For a UML data schema and a short description, cf. Section 4.4.1.4.

POI_DATA table

PoiId	Automatically generated number which uniquely identifies the POI.
Name	Name of the POI.
Description	Short description of the POI.
Easting	Distance east from the zones central meridian in meters[14].
Northing	Distance north from the equator in meters[14] .
Blog_Link	Link to the guestbook page for the certain POI.

[14] Coordinate Reference System used: EPSG 31295 [ESRI]

Tiscover_Link	Link to a webpage containing further information.

Table 9. Attributes of Table POI_DATA

POI_MAINCATEGORY

MainCatId	Automatically generated number which uniquely identifies the POI.
Name	Name of the POI main category.
Visible	A boolean value indicating whether the category is visible at the start-up of the application.

Table 10. Attributes of Table POI_MAINCATEGORY

POI_SUBCATEGORY

SubCatId	Automatically generated number which uniquely identifies the POI.
Name	Name of the POI (sub) category.
Symbol	Name of the symbol, which presents the POI on the map. All symbols are stored in a JAR file in the file system on the web server.
MainCatId	Id of the main category to which this POI category belongs.

Table 11. Attributes of Table POI_SUBCATEGORY

POI_MAPPING

POIId	Id of the POI.
SubCatId	Id of the category the POI belongs to.
DefaultSymbol	A POI may belong to more than one category, e.g. the “Old Castle” of Linz may belong to the category “Museum” as well as to “Restaurant” if it includes one. In this case, the boolean value “DefaultSymbol” indicates which of these two categories is used to present the POI “Old Castle” on the map.

Table 12. Attributes of Table POI_MAPPING

WMS_DATA

WmsId	Automatically generated number which uniquely identifies the WMS.
Url	Url Prefix of WMS Server.
Service	Service Type, e.g. WMS.
Version	Request version.
Request	Request name.
MapLayer	Name of the street map layer.
AerialViewLayer	Name of the orthophoto layer.
DefaultLayer	Indicates which of the layers is the default one.
Srs	Spatial Reference System.
Styles	Comma-separated list of one rendering style per requested layer (optional attribute).
Format	Output format of map.
Width	Width in pixels of map picture.
Height	Height in pixels of map picture.

Table 13. Attributes of Table WMS_DATA

For more information on WMS specific parameters e.g., cf. [Kolo03].

SIM_DATA

SimId	Automatically generated number which uniquely identifies the WMS.
Easting	Distance east from the zones central meridian in meters[15].
Northing	Distance north from the equator in meters[15] .

Table 14. Attributes of Table SIM_DATA

[15] Coordinate Reference System used: EPSG 31295 [ESRI]

References

[AKMP02] H. Anegg, H. Kunczier, E. Michlmayr, G. Pospischil, M. Umlauft: LoL@: Designing a Location Based UMTS Application. ÖVE-Verbandszeitschrift e&i, Springer (2002)

[BaCK05] J. Baus, K. Cheverst, C. Kray, A Survey of Map-based Mobile Guides. In A. Zipf, L. Meng, T. Reichenbacher (eds) Map-based mobile services - Theories, Methods and Implementations. Springer, Berlin, 2005

[BBKK04] J. Baldzer, S. Boll, P. Klante, J. Krösche, J. Meyer, N. Rump, A. Scherp, H. Appelrath, Location-Aware Mobile Multimedia Applications on the Niccimon Platform. In 2. Braunschweiger Symposium – Informationssysteme für mobile Anwendungen, 2004

[BeLL03] S. Berger, H. Lehmann, F. Lehner, Location-based Services in the tourist industry, International Journal on Information Technology & Tourism, Cognizant, 5(4), 2003

[BoKS04] S. Boll, J. Krösche, A. Scherp, Personalized Mobile Multimedia meets Location-Based Services. In: Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 - Informatik verbindet, Band 2, Beiträge der 34. Jahrestagung der Gesellschaft für Informatik -- Workshop "Multimedia-Informationssysteme", Ulm, Deutschland, LNI, 51(2), GI, 9/2004

[Brim02] A. J. Brimicombe, GIS: Where are the frontiers now? Proceedings GIS 2002, Bahrain: 33-45

[BrLa05] B. Brown, E. Laurier, Designing electronic maps: an ethnographic approach. In A. Zipf, L. Meng, T. Reichenbacher (eds) Map-based mobile services - Theories, Methods and Implementations. Springer, Berlin, 2005

[CDMF00a] K. Cheverst, N. Davies, K. Mitchell, A. Friday, Experiences of developing and deploying a context-aware tourist guide: the GUIDE project, Proceedings of the 6th annual international conference on Mobile computing and networking, August, Boston, 2000

[CDMF00b] K. Cheverest, N. Davies, K. Mitchell, A. Friday, The Role of Connectivity in Supporting Context-Sensitive Applications. Distributed Multimedia Research Group, Department of Computing, Lancaster University. 2000

[DeAb99] A. Dey, G.D. Abowd, Towards a Better Understanding of Context and Context Awareness. Technical Report, GIT-GVU-99-22, Georgia Institute of Technolgy, 1999

[Dick05] F. Dickmann, Effectiveness and efficiency of tourism maps in the World Wide Web and their potential for mobile map services. In A. Zipf, L. Meng, T. Reichenbacher (eds) Map-based mobile services - Theories, Methods and Implementations. Springer, Berlin, 2005

[EdBW05] A. Edwardes, D. Burghardt, R. Weibel, Portrayal and Generalisation of Point Maps for Mobile Information Services. In A. Zipf, L. Meng, T.

Reichenbacher (eds) Map-based mobile services - Theories, Methods and Implementations. Springer, Berlin, 2005

[Edli03] S. Edlich, Fliegengewicht. Open Source-Perlen: Die Sourceforge Datenbank HSQLDB. Java Magazin Ausgabe 3.2003

[ESRI] ArcGIS9. ArcXML Programmer's Reference Guide. http://downloads.esri.com/support/documentation/ims_/ArcXML9/Support_files/arcxmlguide.htm. Last accessed July 2005

[FeCo] Federal Communications Commission. Enhanced 911 - Wireless Services. http://www.fcc.gov/911/enhanced. Last accessed July 2005.

[Gart04] G. Gartner, Location-based mobile pedestrian navigation services – the role of multimedia cartography, ICA UPIMap 2004, Tokyo

[GPSP04] F. Garzotto, P. Paolini, M. Speroni, B. Pröll, W. Retschitzegger, W. Schwinger, Ubiquitous Access to Cultural Tourism Portals, Proc. of the Third International workshop on Presenting and Exploring Heritage on the Web (PEH'04), in conjunction with DEXA, Zaragoza, Spain, Aug/Sept. 2004.

[GaUh05] G. Gartner, S. Uhlirz, Cartographic Location Based Services. In A. Zipf, L. Meng, T. Reichenbacher (eds) Map-based mobile services - Theories, Methods and Implementations. Springer, Berlin, 2005

[Geek] GeekBlog. http://geekblog.sourceforge.net. Last accessed July 2005.

[GeSB02] H. W. Gellersen, A. Schmidt, M. Beigl, Multi-Sensor Context-Awareness in Mobile Devices and Smart Artifacts. Mobile Networks and Applications 7, 341-351, 2002

[GrHa04] M. J. O'Grady, G. M. P. O'Hare, Gulliver's Genie: Agency, Mobility & Adaptivity. Computers & Graphics, Special Issue on Pervasive Computing and Ambient Intelligence - Mobility, Ubiquity and Wearables Get Together, Vol. 28, No. 4, Elsevier. 2004

[GrHa05] M. J. O'Grady, G. M. P. O'Hare, Mobile Devices & Intelligent Agents- Towards a New Generation of Applications and Services, Journal of Information Sciences, Special Issue on Intelligent Embedded Agents, Vol. 171, No. 4, pp. 335-353. 2005. Elsevier.

[Harr03] L. Harrie, Generalisation and integration for location-based services. Presented at Schloss Dagstuhl, 29 September – 4 October, 2003

[HaSL02] L. Harrie, L. T. Sarjakoski, L. Lehto, A variable-scale map for small-display cartography. Proceedings of the Joint International Symposium on "GeoSpatial Theory, Processing and Applications" (ISPRS/Commission IV, SDH2002), Ottawa, Canada, July 8-12, 2002

[HaSK04] M. Hazas, J. Scott, J. Krumm, Location-Aware Computing Comes of Age. IEEE Computer Magazine, vol. 37, no. 2, February 2004, pp. 95-97.

[HiBo01] J. Hightower, G. Borriello, A Survey and Taxonomy of Location Systems for Ubiquituous Computing, University of Washington, Computer Science and Engineering, 2001

[HiVo03] A. Hinze, A. Voisard, Locations- and Time-Based Information Delivery in Tourism Advances in Spatial and Temporal Databases, 8th International Symposium, LNCS 2750, Santorini Island, Greece, 2003

[HrHa04] N. Hristova, T.M.P. O'Hare, Ad-me: Wireless Advertising Adapted to the User Location, Device and Emotions. hicss, vol. 09, no. 9, p. 90285c, Proceedings 2004.

[HrHL03] N. Hristova, G. M. P. O'Hare, T. Lowen, Agent-based ubiquitous systems: 9 lessons learnt. In Workshop on System Support for Ubiquitous Computing (UbiSys'03), 5th International Conference on Ubiquitous Computing (UbiComp), Seattle, WA, USA, 2003

[HSQL] HSQL Database Enginge. http://hsqldb.org. Last accessed July 2005

[HuLe95] W. E. Huxhold, G. Levinsohn, Managing Geographic Information System Projects, Oxford University Press, New York, 1995

[Hunt01] P. Hunter, Location, Location, Location. Computer Weekly May Issue: 64, 2001

[HuZi03] S. Hunolstein, A. Zipf, Towards Task Oriented Map-based Mobile Guides. In: Schmidt-Belz et al: HCI in Mobile Guides. Workshop at MobileHCI. Fifth Int. Symp. 2003

[Jung04] K. Jungnickel, Urban Tapestries sensing the city, Proboscis cultural snapshots number eight, 2004

[Kama03] A. Kamar, Mobile Tourist Guide (m-ToGuide). Deliverable 1.4, Project Final Report. IST-2001-36004, 2003

[KlRi04] A. Klippel, K.-F. Richter, Chorematic Focus Maps. In Georg Gartner (Ed.), Location Based Services & Telecartography, pp. 39–44, Geowissenschaftliche Mitteilungen. Technische Universität Wien, 2004

[Kolo03] K. Kolodziej, OpenGIS Web Map Server Cookbook. April 28, 2003. OGC Document Number: 03-050. Draft Version: 1.0.0. Copyright (c) Open GIS Consortium Inc. 170 pages.

[KPRS03] G. Kappel, B. Pröll, W. Retschitzegger, W. Schwinger, Customisation for Ubiquituous Web Applications – A Comparison of Approaches. Int. Journal of Web Engineering and Technology (IJWET), 2003

[KaSR05] E. Kapsammer, W. Schwinger, W. Retschitzegger, Bridging Relational Databases to Context-Aware Services, Proc. of the CAiSE Workshop Ubiquitous Mobile Information and Collaboration Systems (UMICS2005), Porto, Portugal, 13-14 June 2005.

[KoKP01] A. Kobsa, J. Koenemann, W. Pohl, Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships. The Knowledge Engineering Review, Vol. 16, No. 2, 2001

[Koep02] I. Koeppel, What are Location Services? – From a GIS Perspective, ESRI Whitepaper, 2002

[Kolo03] K. Kolodziej, OpenGis Web Map Server Cookbook, OGC Document Number: 03-050, Version 1.0.0, 2003

[KOWO] http://www.kowoma.de/gps/geo/Projektionen.htm. Last accessed July 2005.

[KrBB04] J. Krösche, J. Baldzer, S. Boll, MobiDENK-Mobile Multimedia in Monument Conservation. IEEE MultiMedia, 11(2), 2004

[KrBo05] J. Krösche, S. Boll, The xPOI Concept. In: International Workshop on Location- and Context-Awareness (LoCA 2005) in conjunction with Pervasive 2005, Oberpfaffenhofen, Deutschland, 12.-13. May 2005.

[LaBT04] R. Lake, D. S. Burggraf, M. Trninic, Geography Mark-Up Language (GML) Foundation for the Geo-Web, Wiley 2004

[LaPi98] C. LaPierre, Personal Navigation System for the Visually Impaired. Master thesis. Department of Electronics, Carleton University, 1998. http://lapierre.jammys.net/masters/chapter5.html. Last accessed July 2005

[LiCh04] C. Li, Spatial Ability, Urban Wayfinding and Location-Based Services: a review and first results, Centre for Advanced Spatial Analysis, University College London, 2004

[Loza03] F. Lozano, Eclipse and HSQLDB: Embedding a relational database server into Eclipse, 2003

http://www-128.ibm.com/developerworks/opensource/library/os-echsql. Last accessed July 2005

[MaZi00] R. Malaka, A. Zipf, DEEP MAP - Challenging IT research in the framework of a tourist information system. In: Fesenmaier, D. Klein, S. and Buhalis, D. (Eds.): Information and Communication Technologies in Tourism 2000. Proceedings of ENTER 2000, Barcelona. Spain. Springer Computer Science, Wien, New York. 15-27.

[MaSc01] N. Marmasse, Ch. Schmandt, Location Modeling - workshop position paper. MIT Media Laboratory 20 Ames Street, 2001

[MeRe03] L. Meng, T. Reichenbacher: Geodienste für Location Based Services. Tagungsband zum 8. Münchner Fortbildungsseminar Geoinformationssysteme, 2003

[MeRe05] L. Meng, T. Reichenbacher, Map-based mobile services. In A. Zipf, L. Meng, T. Reichenbacher (eds) Map-based mobile services - Theories, Methods and Implementations. Springer, Berlin, 2005

[Mich02] E. Michlmayr, Flexible Content Management for the LoL@ UMTS Application. Masters Thesis. Vienna: University of Technology Vienna, 2002

[Mund04] T. Mundt, lecture notes of the course “Location-Based Computing”, University of Rostock, Germany, 2004

[NiSa03] A.-M. Nivala, L. T. Sarjakoski, An Approach to Intelligent Maps: Context Awareness. In: Schmidt-Belz, B. and K. Cheverst, (eds.), Proceedings of the workshop W1 "HCI in Mobile Guides 2003". In conjunction with: Fifth International Symposium on Human Computer

Interaction with Mobile Devices and Services, Mobile HCI 03, September 8-11, 2003, Udine, Italy, pp. 45-50

[NMEA] http://www.kowoma.de/gps/zusatzerklaerungen/NMEA.htm. Last accessed July 2005

[NoSP02] J. Nord, K. Synnes, P. Parnes, An Architecture for Location Aware Applications. In Proceedings of the 35th Annual Hawaii international Conference on System Sciences (Hicss'02)-Volume 9 - Volume 9 (January 07 - 10, 2002). HICSS. IEEE Computer Society, Washington, DC, 293.

[NoSi05] M. C. Norrie, B. Signer, Overlaying Paper Maps with Digital Information Services for Tourists. In Proceedings of ENTER 2005, 12th International Conference on Information Technology and Travel & Tourism, Innsbruck, Austria, 2005

[NoSP02] J. Nord, K. Synnes, P. Parnes, An Architecture for Location-Aware Applications. Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS-35'02), 2002

[OGC] OGC. Open Geospatial Consortium. http://www.opengeospatial.org. Last accessed July 2005.

[Oper94] R. Opermann, Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software. In. G. M. Olson, J. S. Olson & B. Curtis (Eds.) Computers, Coginition, And Work, p.253 pub. Lawrence Erlbaum Associates, 1994

[PLMN01] S. Poslad, H. Laamanen, R. Malaka, A. Nick, P. Buckle, A. Zipf, CRUMPET: Creation of User-Friendly Mobile Services Personalised for Tourism. In: Proceedings of: 3G 2001 - Second Int. Conf. on 3G Mobile Communication Technologies. 26-29.03.2001. London. UK. 2001

[PoBS01] G. Pospischil, E. Bonek, A. Schneider, UMTS Applications Development - Designing a "Killer Application". Next Generation Wireless Networks, Editor: TEKINAY, S., Verlag: Kluwer, 2001, Kap.8, pp.255-263

[PoUM02] G. Pospischil, M. Umlauft, E. Michlmayr, Designing LoL@, a Mobile Tourist Guide for UMTS, Mobile HCI 2002, LNCS 2411, 2002.

[PrCW01] M. Preston, P. Clayton, G. Wells, The importance of location in the location-transparent medium of the Internet. SATNAC 2001

[PrRe00] B. Pröll, W. Retschitzegger, Discovering Next-Generation Tourism Information Systems - A Tour on TIScover, Journal of Travel Research, Sage Publications, Inc., Vol. 39/2. November (2000)

[RaWi02] M. Raubal, S. Winter, Enriching Wayfinding Instructions with Local Landmarks. In GISscience 2002, Lecture Notes in Computer Science, Berling: Springer, 2002

[Reich02] T. Reichenbacher, SVG for adaptive visualisations in mobile situations. Proceedings of the SVG Open, 15. - 17. Juli 2002 (CD-ROM), Zürich

[Reic03] T. Reichenbacher, Adaptive Methods for Mobile Cartography, In: Proceedings of the 21st International Cartographic Conference (ICC), Durban, 2003

[Reic04] T. Reichenbacher, Mobile Cartography - Adaptive Visualisation of Geographic Information on Mobile Devices. Verlag Dr. Hut, München, 2004, ISBN 3-89963-048-3

[Reic05] T. Reichenbacher, Adaptive egocentric maps for mobile users. In A. Zipf, L. Meng, T. Reichenbacher (eds) Map-based mobile services - Theories, Methods and Implementations. Springer, Berlin, 2005

[ReSc03] G. Reitmayr, D. Schmalstieg, Collaborative Augmented Reality for Outdoor Navigation and Information Browsing. In LBS & TeleCartography, G. Gartner, Ed. Geowissenschaftliche Mitteilungen, vol. 66, 2003, pp. 53-59, 2003

[Roth02] J. Roth, Context-Aware Web Applications Using the PinPoint. IADIS International Conference WWW/Internet 2002, Lissabon (Portugal), IADIS Press, 3-10, 2002

[ScBo04] A. Scherp, S. Boll, mobileMM4U-framework support for dynamic personalized multimedia content on mobile systems. In: Proc. des Techniques and Applications for Mobile Commerce (TaMoCO) Track der Multi-Konferenz Wirtschaftsinformatik 2004, Essen, Deutschland, März 2004, 3, Aka GmbH, S.204-215, 2004

[ScLa01] A. Schmidt, K. van Laerhoven, How to Build Smart Appliances?. IEEE Personal Communications, Vol. 8, No. 4, 2001

[ScCL05] A. Schilling, V. Coors, K. Laakso, Dynamic 3D Maps for Mobile Tourism Applications. In A. Zipf, L. Meng, T. Reichenbacher (eds) Map-based mobile services - Theories, Methods and Implementations. Springer, Berlin, 2005

[SGPR05] W. Schwinger, Chr. Grün, B. Pröll, W. Retschitzegger, A light-weight Framework for Location-based Services. Accepted for publication at the OTM 2005 Workshop on Context-Aware Mobile Systems (CAMS'05), 30 October - 4 November 2005, Agia Napa, Cyprus

[SmSi02] A. Smailagic, D. Siewiorek, Application Design for Wearable and Context-Aware Computers, in: Pervasive Computing, IEEE. Publication Date: Oct-Dec 2002 Volume: 1, Issue: 4 On page(s): 20- 29

[SPNZ02] B. Schmidt-Belz, S. Poslad, A. Nick, A. Zipf, Personalized and Location-based Mobile Tourism Services. Workshop on "Mobile Tourism Support Systems" in conjunction with Mobile HCI '02. Pisa. 2002

[Star01] Starlink Invicta 210 Serial Protocol Definition Document. http://www.ravenind.com/RavenPrecision/GPS/Literature/invictaprotocol.pdf, 2001

[Stüt05] H. Stüttgen, Context-Aware Applications – The Next Step for Mobile Services. Presentation Notes. Industriekolloquium des

Sonderforschungsbereich 627, Universität Stuttgart, March 2005. http://www.nexus.uni-stuttgart.de/i-kolloq05. Last accessed July 2005.

[SURA] SURA. http://twiki.sura.org/bin/view/Main/OGC101#Web_Feature_Service_WFS_. Last accessed July 2005.

[Swed99] G. Swedberg, Ericsson's mobile location solution. Ericsson Review(4), 1999

[Tara03] P. Tarasewich, Designing Mobile Commerce Applications. Communications of the ACM, December 2003/Vol. 46, No.12

[UrTa] Urban Tapestries Project. http://urbantapestries.net. Last accessed July 2005.

[SePK04] M. van Setten, S. Pokraev, J. Koolwaaij, Context-Aware Recommendations in the Mobile Tourist Application COMPASS. In Nejdl, W. & De Bra, P. (Eds.). AH 2004, LNCS 3137, Springer-Verlag, 2004

[SpAY05] S. Spaccapietra, L. Al-Jadir, S. Yu, Somebody, Sometime, Somewhere, Something. International Workshop on Ubiquitous Data Management (UDM2005), in conjunction with IEEE ICDE 2005, Tokyo, 4 Apr 2005

[Wand] Company Wanderman. http://wanderman.net/start.htm. Last accessed July 2005.

[Weis96] Some Computer Science Issues in Ubiquitous Computing. Communications of the ACM, July 1996/Vol. 36, No.7

[WLANd] WLANd Salzburg Project, http://wland.at. Last accessed July 2005.

[ZiJo04] A. Zipf, M. Joest, User Expectations and Preferences regarding Location-based Services - results of a survey. 2nd Symposium on Location Based Services and TeleCartography. 28. - 29. January 2004. Vienna. Austria

[ZiRi02] A. Zipf, K.-F.Richter, Using Focus Maps to Ease Map Reading. Developing Smart Applications for Mobile Devices. KI Special Issue Spatial Cognition, 02 (4), pp. 35–37, 2002