Slovenská technická univerzita v Bratislave
Stavebná fakulta
Katedra geodetických základov

Diplomová práca

2003 **Bc. Martin Tomko**

Slovak University of Technology, Bratislava

Faculty of Civil Engineering

Department of Geodesy

Spatial Databases for Mobile GIS Applications

Master Thesis

Bratislava, 2003 **Bc. Martin Tomko**

Čestné prehlásenie:

Čestne prehlasujem, že som diplomovú prácu vytvoril samostatne s použitím literatúry uvedenej v zozname.

Declaration:

Herewith I declare, that I have elaborated this Thesis autonomously, with the use of the resources and references indicated in the list.

Bratislava, 26.5.2003 Bc. Martin Tomko

Poďakovanie:

Chcel by som sa srdečne poďakovať všetkým tým, ktorí mi boli nápomocní pri práci na tejto diplomovej práci, a ktorí jej vznik umožnili. Vďaka patrí doc. Ing. Ernestovi Bučkovi, PhD., vedúcemu diplomovej práce, za dôveru a za možnosť pracovať na tejto téme, Ing. Ivane Ivanovej, konzultantke, za podporu a cenné pripomienky počas celej práce.

Veľká vďaka patrí aj pedagógom sekcie GIS technológie na Katedre geodézie TU Delft, ktorí mi radami a vytvorenými podmienkami umožnili preniknúť hlbšie do témy a prakticky si overiť vedomosti počas polročného štúdijného pobytu, menovite prof. Ir. Henrimu Aaldersovi, PhD, Ir. Edwardovi Verbree a Dr. Wilkovi Quakovi, vedúcim a konzultantom počas môjho štúdijného pobytu.

Nakoniec by som chcel vyjadriť svoju vďaku zamestnancom firmy Geodan Holding v Amsterdame, menovite prof. Dr. Henk Scholtenovi, za neoceniteľné skúsenosti so skutočnou implementáciou lokalizačných služieb, ktoré mi umožnili získať počas stáže.

Acknowledgments:

I wish to acknowledge and thank all those who contributed to the work on my Master Thesis and made it's creation possible. I wish to thank doc. Ing. Ernestovi Bučkovi, PhD. who was the supervisor of this thesis for confidence and possibility to work on this topic. I would like to express my gratitude to Ing. Ivana Ivanova, thesis consultant, for support and valuable comments during the whole work on the thesis.

A special recognition is due to the members of the professors and STAFF of the GISt section of the Department of Geodesy, TU Delft, specially to prof. Ir. Henrimu Aalders, PhD, Ir. Edward Verbree and Dr. Wilko Quak, who supervised my work during my short stage.

Finally, I would like to thank the team at Geodan Holding, Amsterdam, the Netherlands, and specially prof. Dr. Henk Scholten, for the opportunity to get experience about real-life implementations of LBS, during my internship.

Table of Contents

Notation

3GPP – Third Generation Partnership Program

ADT – Abstract Data Type

A-GPS – Assisted GPS

API – Application Programmers Interface

ATM – Automatic Teller Machine

BSC – Base Station Controller

BTS – Base Transceiver Station

CGI – Common Gateway Interface

DBMS – Database Management System

E-CGI – Enhanced Cell Global Identity

EGNOS – Euro Geostationary Navigation Overlay Service

E-OTD – Enhanced Observed Time Difference

EPE – Estimate Position Error

ESRI – Earth Resources Research Institute

FTP – File transfer Protocol

GIS – Geographic Information System

GML – Geography Markup Language

GNSS – Global Navigation Satellite System

GPS – Global Positioning System

GSM – Global System for Mobile Communication

HLR – Home Location Register

HSCSD – High Speed Circuit Switched Data

HTML – Hypertext Markup Language

HTTP – Hypertext Transfer Protocol

IP – Internet Protocol

LBS – Location Based Services

LDT – Location determination Technology

LIF – Location Interoperability Forum

LMC – Location Measurement Centre

LMU – Location Measurement Unit

MIME – Multimedia Internet Mail Extension

MLP – Mobile Location Platform

MO-LR - Mobile Originating Location Request

MSAS – Multi-functional Satellite Augmentation System

MSC – Mobile Switching Center

MT-LR - Mobile Terminating Location Request

MU – Mobile Unit

NCGI – National Centre for Geographic Information

NMEA – National Marine Electronics Association

NSDI – National Spatial Data Infrastructure

OGC – Open GIS Consortium

OODBMS – Object Oriented Database Management System

ORDBMS – Object Relational Management System

PDA – Personal Digital Assistant

PDOP – Position Dilution of Precision

RDBMS – Relational Database Management System

SFS – Simple Feature Specification

SLoP - Spatial Location Protocol

SMLC - Serving Mobile Location Centre

SMS – Short Message Service

SMTP – Simple Message Transfer Protocol

SVG – Scalable Vector Graphics

TA – Timing Advance

TCP – Transmission Control Protocol

TOA – Time of Arrival

UMTS – Universal Mobile Telecommunication System

URL – Uniform Resource Locator

WAAS – Wide Area Augmentation System

WAP – Wireless Application Protocol

XML – Extensible Markup Language

1 Introduction

1.1 Background

The need of flexible and real time access of spatial data in the modern society, mainly in business applications, implies a development of a new family of services – Location Based Services (LBS).

The ability to provide data about location on site magnifies their value as the information finds its client in the right moment and on the right place.

The combination of information and mobility opens a large place for the development of applications, using the information about the current location of the mobile user. Such way of personalization of acquired data and the possibility to process them in the modern devices allows the development of different navigation, routing, and spatial management services. The core of these applications is a powerful technology, enabling users to access or use their spatial data on a remote basis, providing Geographic Information Systems (GIS) functionality available until now only to trough strong desktop GIS application.

Approaches enabling IS designers to develop mobile GIS applications without using special analytic middleware GIS applications, profiting of new data storage technologies, encoding and transfer standards, are already available.

Some of the basic concepts of geodetic positioning technologies are also used for Location Determination Technologies (LDTs) used in LBS. The specific problematic of the precision and accuracy of mobile devices positioning is also treated in [Nikolai, 2002] and [Retscher, Winter, 2001] and [Ludden, 2000] and will not be analysed thoroughly in this thesis. The analysis of the suitability of different LDTs for specific LBS application can be found in [Tomko, 2001].

This work investigates an approach to building simple services based on mobile GIS technologies, where strong emphasis is given to pure client/server architecture, where the data are stored in an integrated spatial database and standalone reusable modules that are easy to develop and modify. It builds on previous research as presented in [Oosterom, Quak et al, 2001], [OGC, 2000] and shows how to design LBS system in a Model Driven Approach (MDA) [OMG, 2003] and implement the concept of reusable, interoperable software modules based on object oriented (OO) framework principles. Parallels with real world concepts are shown, as in [Fonseca, Egenhofer et al., 2000].

1.2 Objectives

The objectives of this thesis are to:

- Introduce the concepts of the use of mobile GIS technology in the development of Location Based Services, and the whole chain leading from location determination to the display of spatially contextualized information.
- Analyse the various location determination technologies, determining the areas of use of LBS.
- Study the optimal data storage of spatial data in integrated databases and explain how data are stored, managed and retrieved.
- Explain the impact of LBS on communication technology and its integration with the Internet.
- Transform the presented approaches in architecture of the data flow using a modular concept of the system and present a simple concept of a LBS application fully exploiting the data storage and analytical advantages of integrated spatial databases in combination with a modular system architecture.
- Explain the assets and future development of web based and mobile GIS systems.

1.3 Thesis Outline

The Thesis consists of seven parts, each of them focusing on a specific part of a modular mobile GIS application, with specific focus on the use of integrated databases. A short introduction to each of the remaining chapters follows:

Chapter 2 – Mobile GIS and Location based services: In this chapter, I present the concept of mobile GIS technology standing behind Location based services, as a special type of IS profiting of the possibilities brought to us by the Internet and mobile communication technologies. I introduce the concept of modular approach to the architecture of these applications and mention the benefits of using integrated spatial databases to develop location based services, as a crucial part of the wireless information system architecture.

In Chapter 3 I briefly present the various Location determination technologies (LDT), suitable for LBS development. I analyse their pros and cons from the point of view of technical complexity and suitability for different kinds of LBS. I also tackle the structure of wireless networks to explain the way how these LDTs works.

Chapter 4 studies the benefits of using integrated databases and the path that lead to their development. It explains the concepts of Object – relation data models and the analytical and data management assets and specifics that the use of spatial databases brings, mainly in regards of dynamic data storage and LBS applications. Special queries used for LBS services are presented and explained.

Chapter 5 focuses on internet and wireless communication technologies needed for LBS development. First, it explains how user queries are transmitted over the internet and how the service replies. Than, it presents the architecture of GSM 2.5 generation networks, explains how these are coupled with the internet and how different kind of LBS services use those networks.

Presented concepts and technologies are brought together in Chapter 6, where their synergy is presented in a draft of a modular mobile GIS pilot system, called Miracle. I present the objective of the application, the basic bricks of the system, their role in the architecture and explain how it could be implemented in a Java/Oracle environment.

Finally, Chapter 7 concludes the findings of my work, presents the pros and cons of the approach, mentions some similar approaches and predicts the development in the following years.

2 Mobile GIS and Location Based Services

In the last years we have noticed a huge development in the field of so called „new” technologies, significantly powered by the spread of internet technologies and a decrease of its prices.
As a follow-up, an incredible fast development begun in the area of mobile communication technologies, called “mobile solutions”, probably the most dynamically developing part of the market of the new economics.

Until now, the mobile solutions focused on ways, how to enable a certain form of mobile communication - the transport of voice, latter small amounts of text data (SMS). As the computing power of these devices, as well as the data transmission capacity of the wireless networks increased, more specialized services with extra added value could be developed. Location based services are one of those.

With the introduction of services that are spatially personalized, the users will reach unexpected possibilities. Cellular phones users, that, until now, did never hear about GIS systems, will today profit of the capacities of such technologies – by getting instant information about a car crash, occurring few kilometres further, or about a perfect price offer in the supermarket, in the neighbourhood of which they are right pass by. They will get all these information in the right time and place. They will have the opportunity to orient themselves in a foreign city and a spot will indicate their position and provide navigation instructions. Further information about special LBS applications and a SWOT analysis of the market can be found in [Tomko, 2001].

The users of desktop GIS solutions know the advantages of connecting non-spatial and spatial data of the explored phenomenon in a large number of different applications. However, LBS enables to personalise the information brought to the user using spatial information about him to provide more tailored feedback. That is of major importance with navigation or marketing applications. In this way, the specialized “geo” community makes a big step further to the non-specialised “clients”, by bringing them spatial information without the need to learn and understand maps, coordinates, scales and other characteristics that were needed for centuries.

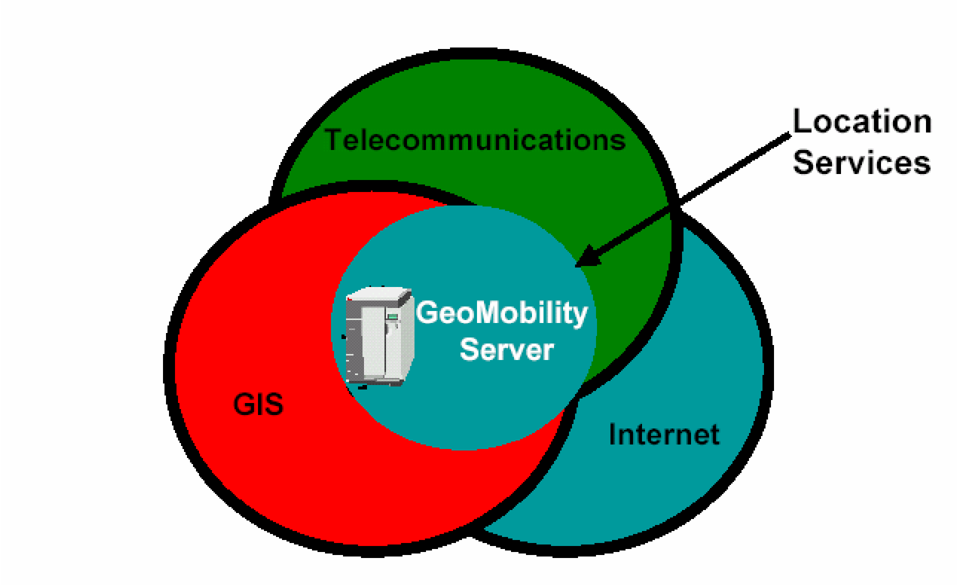


Figure 2.1 Location Services – Technology Context [Buehler, 2002]

“At the intersection of Web, wireless communication and Geographic Information System (GIS) technologies, Location Based Services are aimed at giving everyone the ability to exploit location information anywhere, anytime, and on any device” [Liu, 2002], [Buehler, 2003], Figure 2.1.

2.1 Definitions

In this work, two terms will widely be used: Mobile GIS and Location Based Services (LBS). It is important to clarify the distinctions between the two terms and specify when which one of those should be used.

Mobile GIS – is a GIS as defined in [Worboys, 1995], with some specific features – the whole system or its part is:

- mobile - this means that the data storage and processing part of the system, sensors, and the users are not working in the same place, same computer environment or even do not need to know that they are working together,
- interconnected trough wide area wireless networks, based on mobile telephony technology and the internet, (application or part of the application),
- spatial IS is a background technology – the analytic, data storage and retrieval and data collection technology are just background serving technologies for the whole service – a Location Based Service (LBS).

Sometimes the term **location-aware application** is also used.

Location Based Service (LBS, sometimes also referred as Mobile Location Service, Location Enabled Service or Location Contextualized Service) – is an information service, provided to mobile clients, using location determination technologies and spatial data and analysis to provide information contextualized on the basis of spatial presence of the target user.

Another common definition comes from the [OGC, 2003]:
"*A wireless-IP service that uses geographic information to serve a mobile user. Any application service that exploits the position of a mobile terminal.*"

This means that the information the user gets can be filtered (personalised) according to spatial information about himself or the queried phenomena and can provide information relevant to the user thanks to spatial analysis of the data. The resulting information can be brought to the client in textual, graphic or voice based form and can be in form of position information, routing directions, or even non spatial information – it can even launch actions (medical or traffic alert).

Further on, I will use the term Mobile GIS in two cases:

- while referring to modified versions of standard desktop GIS systems (such as ESRI's ArcPAD, Intergraphs Intelliwhere,…) modified to work in embedded environments – PDAs and palmtop computers,
- while referring to the whole chain of modules and the DBMS as system (application).

I will use the term LBS while describing the implementations of the systems, specific services or analysing those services to deduct what technical parameters should the mobile GIS technology fulfil.

Extensible RDBMS – as defined by [Worboys, 1995], is a standard RDBMS extended with spatial facilities, such has user defined data types and operations on data, indexing and access methods and database functions. As the development goes further, and RDBMS are not "extended" with special "cartridges" and "extension", but the possibility to work with spatial data types, functions and Object-Relation data models are implemented, this definition becomes obsolete and I will use the notion of "**Integrated spatial databases**".

Mobile Positioning System (MPS): A mobile positioning system is the combination of the Location Determination Technology (LDT) and the way how this technology is implemented in the wireless network (both physically and logically). This means, that a MPS designs the whole “position retrieval and communication system” in the framework of the wireless operator.

2.2 Interoperability and System Architecture – An Introduction to Concepts

2.2.1 Mobile GIS Data Flow

To understand how a simple LBS can be created, we should first have a look at all the tasks it has to fulfil to provide the client with the required information:

- Identify the sensors position - get the coordinates of the location, or otherwise acquire such information, using sensors (typically GPS), by preference. The need to enter the position manually would be a big restriction to the quality of the system, so such a possibility is not further discussed in our pilot example in chapter 6.
- Transmit the information about the position to the database query, typically to the client side program that ensures the communication with the spatial server.
- Transmit the query to the server on the web. There, all kind of wireless communication technologies are available.
- Process the query to the spatial database and perform the required GIS analysis. If pre-processing is needed (format, coordinate conversions, more advanced spatial analysis), this should happen ideally on the server side or before the query itself – the case of coordinate conversions and specific GIS analysis – or immediately after the database returns the answer – this is typically the case of data format conversions, such as generating SVG pictures from XML encoded data, coming from the database. In general, these procedures are visualization related.
- Send back of the results and its plotting on the screen of the mobile device.

So, while modelling a mobile GIS application, we can distinguish several basic problem areas requiring special attention:

- data storage and retrieval, including tools for data and transaction management, in integrated spatial databases, that could reconstruct the topology of the searched objects and use spatial queries for data analysis and knowledge extraction,
- location determination technology, that tackles also problems with precision, reliability, availability and coverage,
- remote data access and querying, the communication technology between different devices and in the network, data encryption and communication protocols. Here belongs also the part of displaying/rendering of the final information.

The only difference when compared with the components studied when designing Geo-Information infrastructures [Oosterom, Quak et al., 2001] is in the additional component of Location determination technology. This enables easy implementation of LBS services building on large geo-infrastructure projects, such as the Dutch NCGI.

2.2.2 Typology of LBS Services

There are two basic types of LBS, depending on the relation between the target mobile unit (MU, general notation for all mobile devices), according to the European Telecommunications Standards Institute (ETSI) definitions [ETSI, 2002]:

- LBS with Mobile Originating Location Request (MO-LR): any location request from a client MS to the Location Client Service (LCS – designing any web enabled application) Server made over the GSM air interface. While an MO-LR could be used to request the location of another MS, its primary purpose is to obtain an estimate of the client MS's own location either for the client MS itself or for another LCS client designated by the MS.
- Mobile Terminating Location Request (MT-LR): any location request from an LCS client where the client is treated as being external to the wireless network to which the location request is made.

2.2.3 Modular Architecture

After isolating each part of the chain leading from the query to the result, it is obvious that the whole process can easily be implemented as a system of interconnected standalone, encapsulated modules, communicating each with the other trough well defined interfaces. The benefits of this approach are:

- Independency: all modules are embedded (encapsulated) entities – usually “objects” in the terminology of object-oriented programming that can be modified or replaced without affecting the other modules.
- Interoperability: several datasets (data sources) can be connected to the system. This approach enables to connect datasets contained in the National Spatial Data Infrastructures and other datasets to serve a specific service.
- Reduction of system elements: using spatial databases with spatial analytical functions enables to reduce the number of modules (middleware) to minimum. This improves the flexibility of the system and eases modifications. Spatial data, if stored in integrated databases, can be accessed by a variety of different services, using different architectures and modules. They only need to know the data model (usually encoded in XSLTs or XML DTD – see Chapter 5 and 6) to read the database outputs in the industry standard – XML.
- Transparency: each module has a specific, well defined and limited role in the chain. It is easy to create each of the modules, and easy to debug as well.
- Implementation assets: modular approach enables to distribute the functions of the application between the client and the servers. So, it is possible to use one source of data with a standardised output, for different classes of clients – be it thick, medium or thin, as defined by the Open GIS Consortium in [OGC, 2000], Figure 2.2.

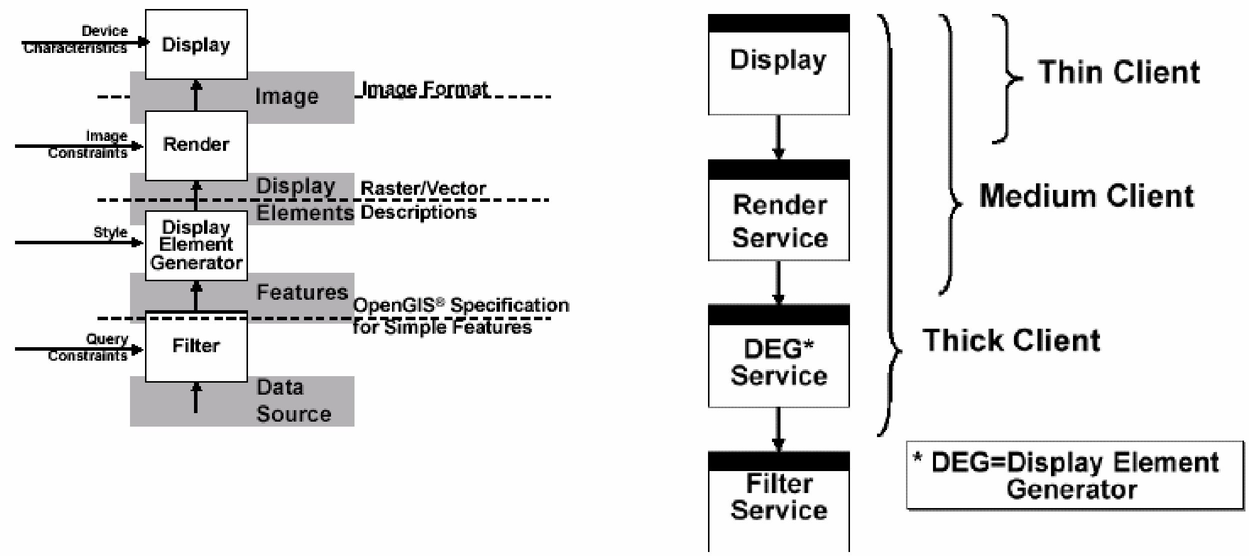


Figure 2.2 – Different levels of data processing in client/server spatial web applications as defined by [OGC, 2000].

Such a modular architecture using spatial databases enables to skip many problems, that were, until now, the building of web based GIS applications complicated – the translation of data between the non-spatial and/or spatial databases to the GIS middleware system, and then again to the interfacing language used to communicate with the clients on the web [Fonseca, Davis, 1999].

Building on the presented concepts, I propose the architecture of a simple LBS MIRACLE in Chapter 6.

3 Location Determination Technologies for LBS

3.1 Characteristics of Location Determination Technologies for LBS

The basic requirement on mobile GIS positioning methods is coverage and portability, sufficient accuracy of the positioning method, and simple interfacing:

- Coverage: this means that the signal should be accessible as widely as possible (in terms of geographic area), for a large sample of the population (service availability and accessibility – this means that is should not be restricted to special users), and costs.
- Portability: the sensors used should not obstruct the user's mobility in any way – such as high power consumption, high expectations on computing power, long observation times and, of course, dimensions.
- Accuracy: the technology used should provide a sufficient, consistent, fast and reliable information about the position of the sensor in a known and well documented spatial reference system precision in the whole covered area.
- Interfacing: Finally, the data output from the sensor should be well documented and easily readable by third party applications.

3.2 Typology of Location Determination Technologies

In general, Location based services rely on wireless location determination technologies, that are or terrestrial (with a limited operational area) or spatial (Global navigation systems).

We can split the LDTs to two major groups – terrestrial and global satellite navigation systems:

- Terrestrial LDTs are represented by different methods to determine the position of the mobile device in the wireless communication operator's networks, be it based on computations done in the handheld device or by the network's LMC (Location Measurement Center).
- Global Navigation Satellite Systems LDTs are currently represented mainly but Navstar GPS satellite based global positioning system and by the emerging Galileo system of the EU (the Russian GLONASS system is not fully functional and no small handheld receivers are available).

This chapter will analyse mainly the positioning functionality of the Navstar GPS system in relation to small handheld devices positioning, and the functionality of the different positioning technologies that can be implemented in 2.5G/3G GSM networks. Other networks and position determination technologies works on a similar basis, with different precisions and with slightly different impact on the operators infrastructure.

The location determination itself is possible in multiple ways, with various accuracy levels. These technologies differs mainly by the role played by the network:

- Network based LDTs (active network technology) – the localisation is performed in the mobile network centre, according to signal incoming from the device.
- Mobile unit based LDTs (passive network technology) – the localisation is performed in the mobile device via synchronized signals, emitted from the BTSs or coming from GPS. A modified mobile device is needed).

3.3 Basics of GSM Communication Networks

To explain how terrestrial LDTs works, it is necessary to understand how GSM communication networks are working. The MU communicates with the operators network by signal transmission to the Base Transciever Stations (BTS). It is a network of antennas, spread over the country, providing signal coverage - Figure 5.1.

The technical characteristics of the main wavelength (frequency) and the bandwidth determine the density of the BTSs in the terrain, depending on the terrain configuration and urbanisation of the area.

In urbanised areas, their number is higher and overlays of signal coverage are bigger (high numbers of people are communicating at the same time, and the bandwidth limits the maximum capacity of the station), the terrain is more complex (dense urban agglomerations with urban canyons and concrete structures inhibits the signal, shorten the served range while the same quality level must be preserved).

In the countryside, BTSs density is lower. The served amount of people is lower and in the case of flat terrain, less BTS are needed to assure homogenous coverage. The maximum range (radius) served by one BTS, in a GSM 900 standard network (the transporting frequency is close to 900MHz) is up to 30 km. Because of the band width, assuring better signal quality and higher transmission capacity, the GSM1800 standard network spreads in the urbanised areas (GSM 2.5). However, the area, served by BTS is only the half of the range served by 900 MHz GSM technology.

3.4 Network Based Location Determination

These technologies use the network of the operator to localise the mobile device. The load of the network increases significantly mainly in urbanised areas, where overload is highly possible. The implementation of network-based services is possible in a relatively short time, as far as standard cellular phones may be located. The expected accuracies of the network-based LDTs is in Table 3.1.

3.4.1 “Cell – ID” and “Enhanced Cell – ID” Technologies

Probably the easiest way, how to implement LBS in the offer of a mobile operator, is to use the localisation based on the Cell-ID technology. It is based on the knowledge of the BTS (or cell), serving the area, where the MU is operating. The position of each BTS is well known to the operator (has its coordinates in a defined reference system), as well as the area served by this BTS. The precision of the position determined with such technology is relatively low – can vary, according to the served area, from several kilometres to few hundreds of meters. Its great advantage is that it is immediately ready to operate.

An improved version of this method is called Enhanced Cell-ID. Only few BTS serve a circular area of 360°. A more common situation is that the BTS serves a section of the circular area. This enables to lower the surface, where the position of the MU is determined.

This technology can further be improved by using a specific parameter of the GSM technology – called Timing Advance (TA). This determines the advance in time, when the signal has to be sent from the MU to the BTS, to arrive there in the right, allocated time slot (or window). It depends on the distance (range) from the BTS to the MU that has to be passed. The speed of the signal is well known. As far as the smallest measurable part of the signal equals to the time needed to transport

1bit of data (= 3.69μs). As far as we are measuring only the half of the distance BTS – MU – BTS, we can determine the position in an area centred in the BTS (circular section) with the width of 550m (=1.85μs), when combined with the E-Cell-ID technology [Retscher, Winter, 2001], [Ludden, 2000], [Nikolai, 2002].

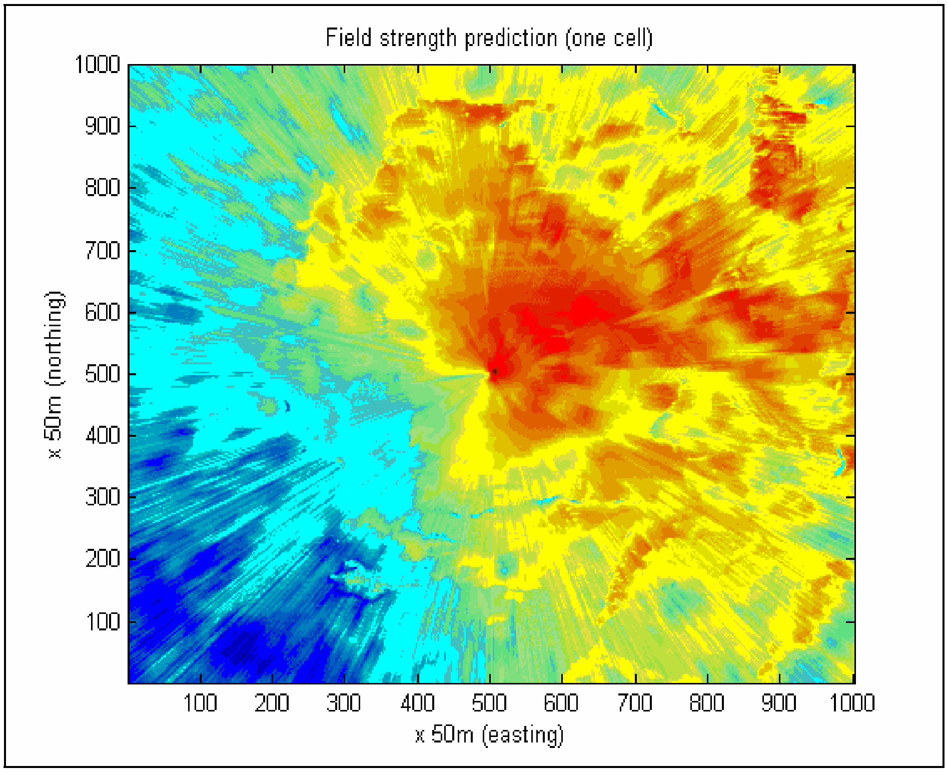
This technology might be used already with a small modification of the network of the operator and without the need of new types of mobile devices at the clients end. If combined with a suitable application software and datasets, we can improve the precision of localisation thanks to semantic information from a spatial DB. (the cars always moves on roads,…). This topic will be discussed later in chapter 4 and 6 and 7.

It is necessary to mention, that the use of TA improves the localisations precision significantly. It's precision is also influenced by the MU, as far as each of them handles the signal in a slightly different way, so the delay might vary and include errors in the localisation determination.
In the UMTS standard network, a similar technology, called Round Trip Time is used. It enables a significantly higher localisation precision, equal to 1/16 of the time slot, so approximately 5m.

3.4.2 Enhanced Cell Global Identity – E-CGI

Another technology, using the network activity, is called Enhanced – Cell Global Identity (E-CGI). If compared with the mentioned technologies, it needs a bigger modification of the network equipment. It determines the position of the device using so called "measurement reports". Every cell of the network observes the intensity of the signal, coming from the phone that communicates with it. The reason is to estimate the intensity of the BTS signal near the phone - Picture 3.1.

These measurements are not performed only by the active cell, but also by the neighbouring BTSs (that do not provide the data transfer). By comparing the intensity of the signal in relation to these cells, we can determine the position of the MU. A model of probable locations is used (based on a robust spatial analytical software in the operators centre), modelling the places where the signal intensity is estimated equal to the measured intensity of signal. The result of the analysis is a prediction of the most probable location of the MU. With a standard BTS antenna height of 30 m and a 1.4dB error in the signal strength intensity determination (equals to a 10% error in range determination) this method allows to determine the position with a higher accuracy than "Cell ID + TA" [Ludden, 2000], [Nikolai, 2002].



Picture 3.1 – E-CGI Mobile Unit prediction model [Nikolai, 2002]. Prediction of signal strength in the surrounding of a BTS directional (red – strong signal, blue – weak signal).

Basic equation for E-CGI:

Signal strength at the device = signal strength emitted from the BTS + BTS antenna gain – Pathloss + device antenna signal gain

The most difficult part is the determination of the signal looses during the transmission. We may, for example, use the so called Hata-Okumora model, that defines the Pathloss by the function of the transmission distance, BTS and mobile unit antennas heights and operating frequency. This topic is further discussed in [Ludden, 2000].

3.4.3 Uplink Time of Arrival

The most precise network based technology is called Uplink Time of Arrival (U-TOA). Its principle is similar to the TA method. The network forces the device to emit a handover request signal (the handover is the act of the mobile unit when changing the serving BTS station), that is then processed in a U-TOA LMU (Time of Arrival Location Measurement Unit). A Location

Measurement Unit (LMU) is an additional device located usually at the BTS or BSC (Base Switching Centers – coordinating units for several BTS) of the network, with known coordinates, assuring the location determination for network based methods. Usually it is a time synchronisation and measurement unit (coordinated trough GPS time signals) with processors for calculating the position of MUs. The time of arrival of the signal is measured. As far as the distance of the device to the LMUs is different, there will be time delays in processing the same signal in the different LMUs. The delays are proportional to the distance of the device. Then its position may be determined with a method based on trilateration.

The advantage of the method is a high level of precision and the compatibility with any mobile device. However, the equipment of the network must be modified significantly, what complicates it introduction. A big number of LMUs must be installed (these are exact time measurement devices, best if synchronised – for example GPS receivers). An installation of such devices would enable to introduce later applications based on the Assisted GPS method (later in the document), DGPS corrections dissemination,… The activity of at least 3 LMUs is needed for position determination. The consequences are in a significant increase of the network load. Also the impact of signal interference is high, when more localisation actions are being performed in one single cell.

The U-TOA method was designed as the method to fulfil the legal requirements of the E911 standard in the USA. This expects a localisation of all the emergency calls with the precision in location determination of at least 125m with a reliability of 67%.

3.5 Mobile Unit Based LDT

These methods perform the calculation position inside the MU itself. The coordinates are then sent to the network centre, or the LBS, for further processing. The load of the operators network is practically the same as when standard data or voice transmission is being performed. This is an advantage mainly in areas with heavy telecommunication network load. The inconvenience is that the user needs a specially modified device.

Two main technologies belong to this category: Enhanced Observed Time Difference (EOTD) and GPS/Assisted GPS (A-GPS). The expected accuracies of the following methods are presented in Table 3.2.

3.5.1 Enhanced Observed Time Difference

EOTD (Enhanced Observed Time Difference) is based on the measurement of incoming time of signals, arriving from the network to the phone. A signal from at least 3 and more BTSs is needed. The position is determined by classical ranging methods, similar to the GPS principle. The network must be equipped with BTS stations with a synchronised time system (not all the BTS must have a LMU, one per 3-5 BTS is sufficient), what keeps the technology less expensive than the TOA LDT. The BTS are spreading an analogy to the GPS message – their identification codes and a coordinated time signal with a time stamp. Thanks to this information the mobile device is able to identify its position and send it to the network for further processing – for example to an emergency service dispatcher.

Even if the method is very exact and doesn't charge the network with by an enormous load, the inconvenient is the fact that special application software calculating the position must be running in the memory of the mobile device. The E-OTD method was performing good results also when the cellular phone was situated indoors. It could be an appropriate method for urban areas.

3.5.2 GPS and Assisted GPS LDTs

A-GPS is a localisation method, using the GSM network in combination with the Navstar GPS system, to improve the standard GPS localisation possibilities in areas, where it can not provide good results (urban areas), or to speed its performance. It enable to rise the precision (or even enable the functioning) indoors.

The role the GSM network plays is not standardised and several models of assistance exists. The network can transmit some of the standard GPS navigation data - the ephemeris, clock corrections and others (for example the WAAS augmentation corrections), what enables a faster initialisation after signal fall-down, after a loose of satellite signal or while switching off/on - the "time to first fix". The battery consumption is lowered (results in longer life time in stand by mode) thanks to the "push" element of the GSM network. The sensitivity of the system is higher in bad atmospheric conditions, urban canyon areas, or between trees.

The BTSs can also act as pseudollites, what is suitable for indoor environments, where normal GPS positioning is not possible. The problem is, that they are much closer than "planned" in the

Navstar GPS system (22000km), so problems of time measurement may occur (so called near-far problem).

It is also necessary to mention the possibility to use standard GPS receivers embedded in cellular phones, mainly if the used receivers are Wide Area Augmentation System (WAAS), or in Europe Euro Geostationary Navigation Overlay System – EGNOS) enabled.

The principle is to reduce the influence GPS signal errors caused by ionospheric disturbances, timing and satellite orbit errors. These influence significantly every GPS positioning, but are due to system/atmospheric errors and cannot be reduced by improving the device or in single phase C/A measurements. As far as the values of these corrections are the same for wider areas, the idea of broadcasting the corrections for a given areas raised. The accuracy of standard C/A code positioning should rise to more than 3 m in 95 percents of cases while using WAAS or it homologue technologies EGNOS and MSAS [www.gpsinformation.net].

WAAS consists of approximately 25 ground reference stations positioned across the United States that monitor GPS satellite data. Two master stations, located on either coast, collect data from the reference stations and create a GPS correction message. This correction accounts for GPS satellite orbit and clock drift plus signal delays caused by the atmosphere and ionosphere. The corrected differential message is then broadcast through one of two geostationary satellites, or satellites with a fixed position over the equator. The information is compatible with the basic GPS signal structure, which means any WAAS-enabled GPS receiver can read the signal – actually, the L1 GPS carrier is used.

3.5.3 The Accuracy of Mobile LDT

The mobile LDTs accuracy is in general dependent on the GSM signal quality (coverage, in the case of Cell-ID also the dimension of the cell), and the frequency of the communication technology. With the frequency rising, also the linearity of propagation of the wave increases. There are less reflections from the atmosphere and objects (reduction of the multipath effects and burst). This is an advantage not only in the case of voice and data transmission, but allows also a better localisation of the mobile device. While using the network based mobile LDTs, we can neglect the impact of the atmospheric characteristics, the corrections, resulting of it, would not have a big influence on a more exact localization (the distances are very short and layers of atmosphere can be thought as homogenous in these applications – in contrary of the exact geodetic applications). However, with rising the frequency, a new problem appears – the near far problem. The time intervals between the emission of the signal and its reception are so short that even a very small error in its measurement causes considerable errors.

Method	Precision in urban areas	Precision in suburban / countryside areas
Cell- ID	500m-5km	1-35 km
E –Cell-ID + TA	550m wide area to 1 km	550m wide area
E-CGI	50-550m	250-8km
U-TOA	cca 100 -120m (5 LMU)	cca 35m (5 LMU)

Table 3.1 - Rough estimates of network-based LDT accuracy [Ludden, 2000].

Method	Precision in urban areas	Precision in suburban / countryside areas
E-OTD	80m (5 BTS)	60m (5 BTS)
A – GPS	5-10, 25m indoors	3 - 10m (depending on WAAS availability)

Table 3.2 - Rough estimates of MU based LDT accuracy [Ludden, 2000].

In general, GSM networks were not designed for location determination purposes (as well as it is the case for data transmission). Every network operator tries to minimize the number of ground cells with keeping the coverage and telephony functionality. This explains why there are so few overlaps between cells -making GSM positioning very difficult.

Anyway, it can be used for some positioning applications, with sufficient accuracy for public applications. GSM networks can also very well assist other positioning technologies to get higher accuracy, shorten time to first fix or provide additional information.

Network based positioning performs the best in urban areas, where cell overlaps are frequent and the accuracy is at applicable levels. GSM BTSs can also act as pseudolites for indoor areas or urban canyons and so widen the use of other technologies. With the arrival of UMTS technologies, that are already designed to provide positioning, we can expect high performance of mobile device positioning also in these conditions. It is expected that UMTS will provide accuracy of about 17m, that can very well compete with the performance of GPS in difficult conditions.

4 Spatial Databases for LBS

For years, spatial representations (data) of about boundaries, parcels or road networks were handled separately from their attribute (non-graphical) data, in various raster or vector files, often without any topological modelling of the relations between the various elements contained. Attribute data were stored in relational databases. Those in an unmodified state are not suitable for spatial data management [Worboys, 1995] of object-based models (models where information is treated as a discrete entity with a georeference), mainly due to the lack of spatial data structures support (topological modelling), indexes for faster spatial data organisation and searches, and efficient data query and analysis tools – SQL 92 doesn't support spatial operations in its standard syntax.

The gap between attribute data and their graphic representation caused a number of data management problems. Limitations such as scale (users had to decide in which scale they want to present their data, so they can specify the level of details required by both – graphical and attribute data), difficult handling of heterogeneous data, problems with applications that needed automated topology reconstruction, automated generalisation, spatial indexing and support for heterogeneous client device rendering. Further, standard data management tools, present in relational databases, were missing in special GIS applications that were used for spatial data storage (such as ArcINFO). Transaction and integrity management of large datasets and data interchange problems needed to be solved, mainly with the rise of web based and mobile applications.

"*Until recently the spatial data management was handled by the GIS software outside the DBMS. As DBMSs are being spatially enabled, more and more GISs (Arc/Info, Geomedia, Smallworld) are or will soon migrate towards an integrated architecture: all data (spatial and thematic) are stored in the DBMS. This marks an important step forward that took many years of awareness creation and subsequent system development.*" [Oosterom, Quak et al., 2001]

To achieve this, several approaches were used – or new data types became native in new versions of DBMS, or special extensions (modules, cartridges), that support spatial data storage, querying and analysis can be installed. Those also implements spatial indexing, clustering and hashing, features, that are needed for reasonable data querying and retrieval in large spatial applications. This functionality traduces itself in new clones of data access tools - SQL languages (currently SQL3 is being developed).

The goal of this chapter is to provide insight in the basics of spatial modelling, querying, data structures and algorithms necessary for spatial applications and mainly LBS, in the environment of integrated spatial DBMS.

This work focuses on the implementation of object-based models (vector), implemented as Object-relational conceptual models in DBMS, with special focus on the implementation in the web environment for LBS. Dynamic (also called kinetic – [Basch, Guibas, 1997]) data handling will also be treated, as this is of a major concern for LBS.

4.1 Integrated Spatial Databases – Key Concepts

As mentioned earlier, we can define a spatial database system as a database system, supporting spatial data types and its data models and query language, spatial indexing and spatial joins algorithms

To be able to extend relational RDBMS with object-based data models and create integrated DBMS, spatial data types needed to be implemented. While creating the conceptual model of the real world, that is free of implementation details and express and is user-comprehensible [Worboys, 1995], the space needs to be decomposed into geo-objects, each identifiable, relevant and describable with its properties, behaviour, structural characteristics and georeference [Oosterom, Quak, 2002], [Worboys, 1995]. This could be described by

- spatial (or semantic – a real world fence is a boundary between two polygons),
- graphical (it can be represented by a point),
- temporal (the parcels were divided in April 2002) and
- textual characteristics (the fence is made of wood, it is brown and high 2 m)

of this geo-object.

Then, we need to describe an implement the relations between such spatial geo-objects – such as functions for measuring distances and defining topological relations.

The geo-objects of the real world (houses, boundaries, roads) need to be modelled in the databases as spatial objects – represented by geometric primitives (geometries). Those are implemented in the databases as special spatial data types and their collections – points, lines and polygons. A high level of standardisation was reached by designing “simple features”, in the OGC Simple Features Specification [OGC, 1999], [OGC, 2001], [Oosterom, Verbree et al, 2001]. However, complex features (supporting topological relations) still needs to be modelled in a standardised manner.

4.2 Spatial Data Type Implementation

4.2.1 Geometry

“*Spatial data represents the essential location characteristics of real or conceptual objects as those objects relate to the real or conceptual space in which they exist. The spatial component of a spatial feature is the geometric representation of its shape in some coordinate space. This is referred to as its geometry.*” [Oracle, 2001].

A geometry is an ordered sequence of vertices that are connected by straight line segments or circular arcs – it is in fact the representation of a real world spatial feature, modelled as an ordered set of primitive elements. The semantics of the geometry are determined by its type. A standardised (and in the different implementations enlarged) set of these primitives is called Simple features [OGC, 1999]. For example, Oracle Spatial supports several primitive types and geometries composed of collections of these types, including 2-dimensional. The geometries that are collections of simple (primitive) types, can be homogeneous (a line string consists only of line segments) or heterogeneous (a compound polygon can be a set of line string and circular arcs). Even multi- features are possible, such as polygons with an island/hole.

4.2.2 Simple Features

Simple features are features with spatial and non spatial attributes, representing entities of the real world. In the simple features standard, spatial attributes represent elementary 2D geometries with linear interpolation between vertices. This signifies that no arc or circles are possible (which is in contrast with some industrial implementations). Let’s focus more in depth on the geometries implemented by the OpenGIS Consortium’s (OGC) Simple Feature Specification (SFS) [OGC, 1999] - Figure 4.1, that became the standard platform for other, proprietary implementations:

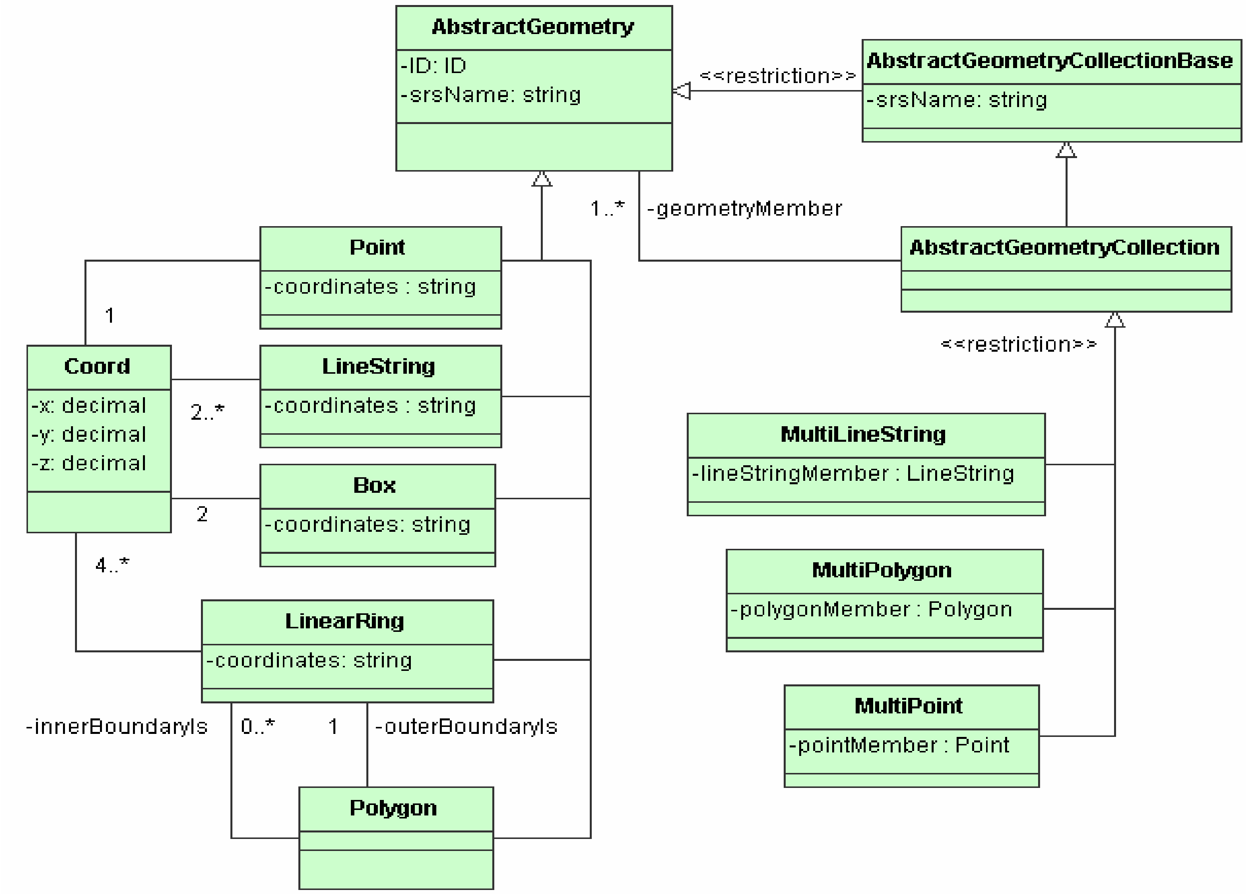


Figure 4.1 – OGC Simple Features Geometry class hierarchy UML schema

The schema provides a clear overview of the hierarchical relationships between the most simple features (coordinates, points), and the way how those are composed to create feature collections.

"Simple geospatial feature collections will conceptually be stored as tables with geometry valued columns in a Relational DBMS (RDBMS), each feature will be stored as a row in a table. The non-spatial attributes of features will be mapped onto columns whose types are drawn from the set of standard ODBC/SQL92 data types. The spatial attributes of features will be mapped onto columns whose SQL data types are based on the underlying concept of additional geometric data types for SQL. A table whose rows represent Open GIS features shall be referred to as a feature table. Such a table shall contain one or more geometry valued columns. Feature table implementations are described for two target SQL environments: SQL92 and SQL92 with Geometry Types." [OGC, 1999]

In the extended relational models (Object relational) or even object oriented data model, methods have to be taken into account. Not only the mutual relationships of the records (tuples), but also their behaviour is important. This includes support for simple spatial analytical methods (operators), such as “neighbours with”, “lies within” or “distance is” or, in complex systems, for the relations that are automatically remodelled between parcels when their boundaries are modified (the reconstruction of topology is automated) where the system doesn’t allow to modify the boundaries incorrectly – for exampling by letting and “open” boundary to be introduced,… A topology is modeled as a complex feature with references between its different components. These features should be provided by complex features support.

4.3 Spatial Data Management

Under spatial data management, all spatial analytic and data manipulation can be mentioned. This is in practice realised trough the implementation of spatial methods (operators) in the query languages, providing tools for indexes, spatial join algorithms and metadata description, and by defining a structure of tables containing the data and metadata -Figure 4.2.

4.3.1 Metadata

Both the SQL92 and the SQL92 with Geometry Types implementations extend the SQL92 Information Schema in a uniform manner so as to support standard Metadata Queries that return:

1. The list of feature tables in a database.
2. The list of geometry columns for any feature table in the database.
3. The Spatial Reference System for any geometry column in the database.

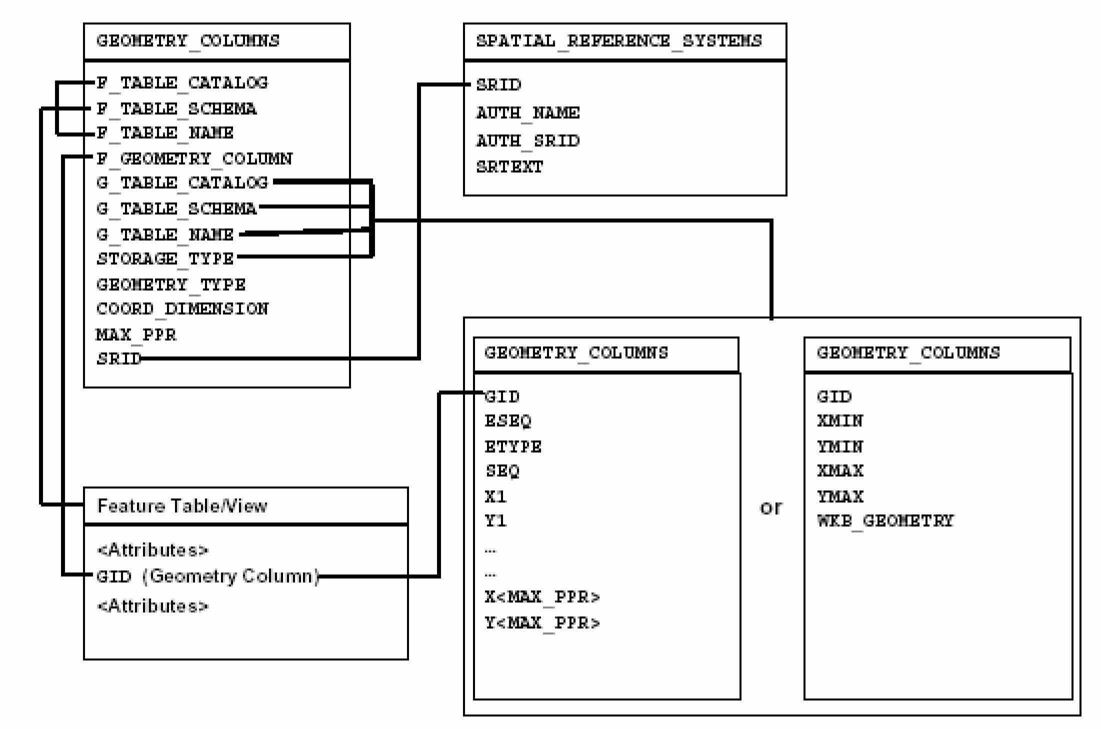


Figure 4.2 – Schema of SF tables in SQL92

This is crucial for implementation purposes, as far as those tables keep track of all the spatial data in the database. This is the only way how to distinguish alphanumerical records from specialised spatial record in spatial tables.

4.3.2 Methods

The operators that needs to be implemented in general groups among three types – basic methods on geometries, methods for testing spatial relations and methods for spatial analysis. The implementation of those methods in SQL and their full list is available in [OGC, 1999]. Here are some of the most important ones:

- Basic methods on geometries: Dimension, GeometryType, SRID, Envelope (minimum Bounding box)
- Methods for testing Spatial Relations between geometric objects: Equals, Disjoint, Intersects, Touches, Crosses, Within, Contains, Overlaps, Relate
- Methods that support Spatial Analysis: Distance, Buffer, ConvexHull, Intersection, Union, Difference,

Each of the specific features has also some specific methods – as for example area, centroid, or length.

For spatial relation analysis, the OGC SFS implements Relational operators based on the Dimensionally extended Nine-Intersection Model, where relationships between two objects are mapped according to the relations of the interiors, exterior and boundaries of the respective objects. Each of the types of spatial relationship is defined by a specific combination of the entities.

4.3.3 Spatial Reference Systems

Every geometry column is associated with a Spatial Reference System. The Spatial Reference System identifies the coordinate system for all geometries stored in the column, and gives meaning to the numeric coordinate values for any geometry instance stored in the column.

In the SFS notion, Spatial Reference System use a well known text representation according to the EPSG (European Petrol Surveying Group) coordinate system data model. A spatial reference system, also referred to as a coordinate system, is a geographic (latitude-longitude), a projected (X,Y), or a geocentric (X,Y,Z) coordinate system. Each coordinate system is composed of several objects: Text representation, nature of the system (projected, geocentric, geographic), SRSID, DATUM, UNIT, each followed by the defining, comma-delimited, parameters of the object in brackets. Some objects are composed of objects so the result is a nested structure. This allows to create custom made projections and datums, use their combinations. The full definition of a system may look as follows - Table 4.1, [OGC, 1999]:

```
PROJCS[‘NAD_1983_UTM_Zone_10N’,
GEOGCS[‘GCS_North_American_1983’,
DATUM[ ‘D_North_American_1983’,SPHEROID[‘GRS_1980’,6378137,298.257222101]],
PRIMEM[‘Greenwich’,0],UNIT[‘Degree’,0.0174532925199433]],
PROJECTION[‘Transverse_Mercator’],PARAMETER[‘False_Easting’,500000.0],
PARAMETER[‘False_Northing’,0.0],PARAMETER[‘Central_Meridian’,-123.0],
PARAMETER[‘Scale_Factor’,0.9996],PARAMETER[‘Latitude_of_Origin’,0.0],
UNIT[‘Meter’,1.0]]
```

Table 4.1 – Definition of the UTM Zone 10 SRS

The SPATIAL_REFERENCE_SYSTEMS table stores information on each Spatial Reference System in the database. The columns of this table are the Spatial Reference System Identifier (SRID), the Spatial Reference System Authority Name (AUTH_NAME), the Authority Specific

Spatial Reference System Identifier (AUTH_SRID) – which is a conventional code assigned to each of the standard SRS, and the Well-known Text description of the Spatial Reference System (SRTEXT). The Spatial Reference System Identifier (SRID) constitutes a unique integer key for a Spatial Reference System within a database. Interoperability between clients is achieved via the SRTEXT column which stores the Well-known Text representation for a Spatial Reference System – for example RD for the Dutch SRS - as described (Including parameters) in [OGC, 1999]. Unfortunately, in [OGC, 1999], S-JTSK, the Czech and Slovak national SRS was not included, but its implementation is coming in the next versions. It also is implemented in the EPSG catalogue and Oracle 9i.

The SRSID attribute of each geometry feature is crucial for a number of spatial analytic and conversion tasks, that can be now fulfilled directly in the database and needn't to be implemented in a middleware GIS application as it was the case until now. This has a big importance in the area of LBS, where data collection is usually done in the WGS84 SRS, and often needs to be coupled with data from different sources in national SRS. Also, the data may need to be presented in a different projection for publication purposes. All this may be solved by implementing a function and generating a view on the same data in a different SRS, or by converting the table to a new one with a new SRS.

The specification of a SRS is also an important consistency control factor, as it helps to omit errors arising from joining tables with different SRS values, or buy adding geometries with coordinates in a non-specified/unknown/different SRS to the table.

New / non-standard SRS may also be created by adding an entry to the SPATIAL_REFERENCE_SYSTEMS table and by entering the values of the SRS. This is solved individually in each implementation of spatial DBMS in a different way.

4.3.4 Further Implementations of the SFS

The OpenGIS consortium made a significant effort to promote the Simple Features Specification and related implementations. One of the most important with regard to integrated databases and LBS, is the one meant for standardised data interchange between various DBMS and geo-applications. It is called the Geography Markup Language (GML, current release 3.0). Referring to the Figure 2.2, GML is used at the Thick client level in the DEG service (Data Exchange

Generator). Not only it enables to interchange the data, but is also use to send the data for further processing and rendering on thick client applications.

4.4 Data Access

Until now, I have presented only the definitions of the standardised data types and the way how they should be stored in spatial databases, including metadata entries. However, one of the most important aspects of spatial data storage is implementing spatial access methods to the stored data - spatial indexing and clustering (hashing). This has to be implemented to the DMBS so that we can talk about Integrated spatial DBMS.

This is not standardised in the OpenGIS standards and each DMBS vendor chooses its own approach. However, it is possible to present the basic expectations a user has on the indexing capabilities of a DBMS.

Indexing – data indexing is a mechanism to limit searches (basically primary searches) to a smaller amount of matching records. The goal is to speed up the retrieval of data in larger databases. Usually the index assigns integer values to records according to their spatial extent and relative position, based on an algorithm that enables faster retrieval of close or related objects. The index values are then searched, usually based on the simple binary tree search. Index based search results are often referred as primary search, where the secondary (exact matching) search is performed later on, on the filtered subset of data.

By spatial clustering, we understand physical storage of data close each to another on disk fields, which is very useful for large spatial datasets, that are stored on multiple hard disks. They are physically stored as “neighbours” when there are also close each to another in the real world. This approach enables to rise the efficiency of spatial searched trough the datasets and rise the performance of the system. It is important mainly in very large datasets, such as aerial imagery or cadastral data. More information on performance of spatial indexes and clusters can be found in the benchmarking reports of the Geo-Database Management Centre of TU Delft in [Oosterom, 2001].

Spatial indexing is used to assign geographically close objects with index of close (and specific) value, so that while querying they can be retrieved fast. Several ways of indexing exists, suitable for different kind of dataset. In general, databases are indexed according by one of the attributes (rows in relational databases). Spatial indexing brings the problem of assigning a single value to a multidimensional object (2D at least, in general). This can be solved by assigning a single code to an area of the covered space. The problem of assigning this value – or also, how to order the discrete areas in space – lead to different algorithms, among the most popular are various Space Filling Curves, that can be tuned to provide best performance for different kinds of searches. No all indexes are space filling curve based – as it is the case of the R-Tree. The Region Quad tree, on its turn, can be ordered using numerous space filling curves, as described in [Worboys, 1995], [Oosterom, Quak, 2001], [Oosterom, Vijlbrief, 1996] - Figure 4.3.

The performance of different indexes is evaluated by the nature of the data contained in the dataset (field representation vs object-based), and the intensity and type of use of the database. The later depends mainly on the frequency of the use of the main functions: Search, Insertion and Deletion of records.

The most common indexing technologies are based on R-tree, Quad-tree or Space filling curves, but many more exists, including modifications and combinations of the mentioned approaches – as for example the Dutch Spatial Location Code – a unique approach to encode both feature extent and position into a single number, using a combination of Quad-tree, Morton's code (space filling curve) and Field tree.

<u>Space Filling Curves</u> – SFC is a way of ordering the cells from n dimensional to a single dimension value, trough bijective assignation. The principle is to chose and ordering of the cells (their numbering) in a way best fitting the purposes of the application and the data contained - Figure 4.3. They are usually used in combination with quad-trees (defining the hierarchy and granularity of space division) to customise the order of numbering – the index creation process. Their performance related to a specific implementation is assessed trough several characteristics: their total length, segment length variability, and distance of neighbour cells trough the curve (both average and maximal).

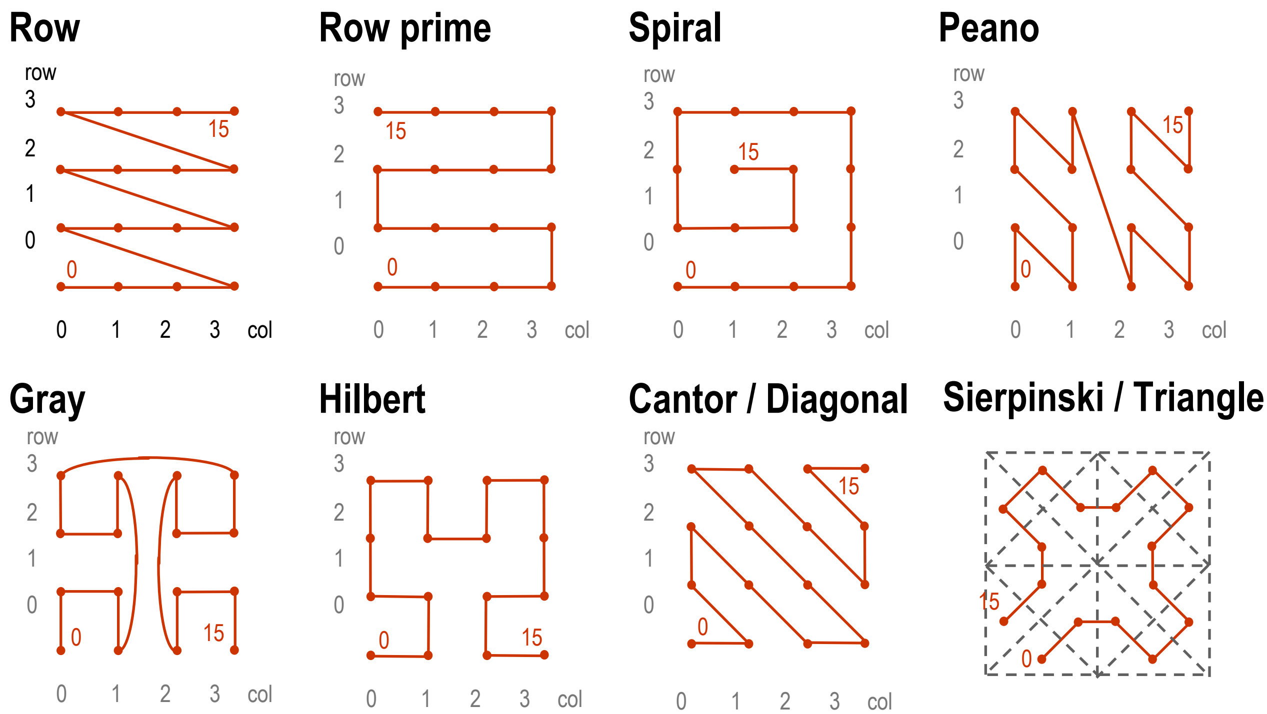


Figure 4.3 - Most common space filling curves

As mentioned before, frequent manipulation with data (insertion and deletion), can cause that the index will need to be rebuild. If some nodes were deleted or added, the whole algorithm of assignation of unique code needs to be run to rebuild it and so keep the performance of the database searches. The frequency of rebuilds is determined by the capacity of the index to adopt to missing entities (for example in some of R-trees not all bounding boxes needs to contain the same number of entities).

The “depth” of search in every tree type is different and while designing the database, it is necessary to select the algorithm best fitting with the data stored, according to its type (points, lines), typical searches (neighbouring searches, routing, shortest path) and spatial distribution of the data. For unevenly distributed data, balanced trees (such as quadtrees) may not be a good option, as it is easy to predict the worst case search scenario, but the overall performance is usually lower than with R-trees.

R, R*-trees and region quad-trees are the most commonly implemented algorithms for spatial indexing in commercial spatial databases.

4.5 Topology

The topology reconstruction is a crucial role of every object-based GIS system, often used for parcel boundaries reconstruction, network analysis and other specific spatial functions. Until now there is no standardised fully functional topological implementation in DBMS [Oosterom, Verbree, 2000]. An attempt is made by the OGC in its spatial schema [OGC, 2001] ,to define the aspects of topological modelling at an abstract level. The problem with the implementation of topological functionality to SQL dialects is in its unsuitability to implement do-while or similar iteration procedures with unknown number of steps. This is until now solved in GIS middlewares where such functions can be implemented, or in OODBMS experiments, where “methods” in the full OO programming meaning are available. This means, that current DBMS are able to model topological relations and implemented, for example, oriented faces, but topological consistency check methods are not available as a standard yet.

As published in [Oosterom, Verbree, 2000], it is necessary to move this functionality from middleware to the DBMS, as far as this is a general requirement and differences in implementations may lower the integration possibilities of different datasets.

As mentioned, several known systems of topological modelling are implemented at various levels. The most known are the doubely-connected-edge-list (DCEL), arc-node, or winged-edge [Oosterom, Lemmen, 2001] (“chain+whell”) topology, being standardised at a national level for the TOP10 vector dataset of the Netherlands as NPR3611 - Figure 4.4.

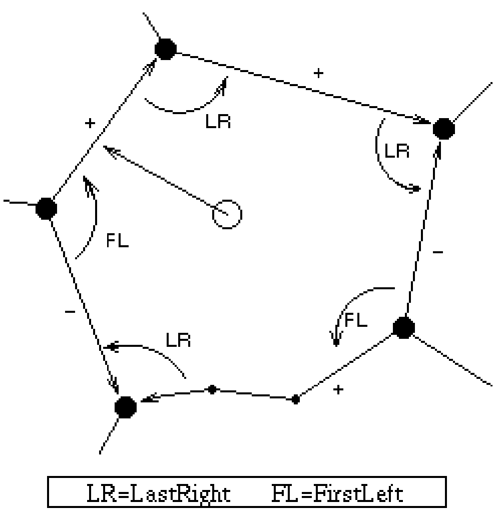


Figure 4.4 – Winged edge topology structure

All those models have their pros and cons, because of having different performance depending on the type of data (data model) implemented in the database. Few are implemented in commercial DBMS, usually only trough foreign keys linking tables with geometries. Then the topology reconstruction is performed trough stored query triggers and select statements. More often, this task is let to the middleware GIS application running on top of the database. Oracle is planning to release topological support in the next version of it DBMS and otherwise relies on 3rd partners topological engines [Lemmen, 2003].

4.6 Kinetic Data

The presented model of indexing spatial data is suitable for handling static data – without needing to update the index frequently due to frequent insertions and deletions of entries, which could make an index unbalanced. However, this is not suitable for data with high temporal variation – typically point position data from MU in LBS, with high variation in space in short intervals (or other applications, where the temporal changes are high – for example temperature monitoring).

The R-tree, which is often implemented in spatial DBMS, is not suitable for applications with frequent updating, due to declining balance of the tree. Meanwhile, it is very suitable form many spatial applications, including nearest neighbour and distance queries.

The idea is to find a possibility to store trajectories instead of punctual data, enabling to predict future positions and adapt the indexing techniques to that. Trajectories based queries would also be useful for navigational and purposes and enter/leave/cross analysis, which are very complex with punctual data [Pfoser, 1995].

An approach to handling mobile data is described in [Saltenis, 1998] and [Wolfson, Xu et al., 1997]. It suggests to implement data with their kinetic characteristics (speed), and generate the position as a function of time. Then, updating the index would only be necessary when one of the parameters changes. This lead to the implementation of a R-Tree based TPR tree index – Time Parameterized Rectangle Tree (also STR tree – spatio-temporal R-tree), where rectangles are predicting the future direction of the object to reduce the need of update.

Such an approach could also be implemented in some of the existing DBMS, where for example Oracle supports indirect, function-based indexing. This means, that one can index a table without geometry stored in the table itself, but, instead, implementing a function that has as result a geometry. This can be used in several cases – a standard one is the indexing of older tables, where for example lat/lon values were stored as integers, and not as geometries. On the other hand, it can as well be used on tables, where coordinates of a point are resulting from a function depending on time and other – spatial or non spatial – columns.

An aspect greatly influencing the performance of those indexes is the nature of the movement – depending if it is a constrained (pedestrians, cars), unconstrained (birds) or network based (railways) movement.

It is clear that a standard implementation of indexes and data types for kinetic data will be more difficult to find than for static data. Its importance is enormous, as LBS for fleet management, navigation and movement monitoring are being more and more common. Combined with datamining technologies, it could help with tasks as traffic jam prevention or alternative route generation, not only based on network data, but also using spatio-temporal aspects of the phenomena. At last, a problem of online index update arises – it is necessary to rebuild it when the service is running, and do so repeatedly. These issues are currently hot research topics.

4.7 Future Development – Object Oriented DBMS

In the near future, we can expect that Object-oriented DBMS (OODBMS) will get a higher share in the market and become implemented not only for research and development purposes.

"*Object Oriented DBMS is an integration of database capabilities with object programming language capabilities, the result is an ODBMS. An ODBMS makes database objects appear as programming language objects in one or more existing programming languages. The ODBMS extends the language with transparently persistent data, concurrency control, data recovery, associative queries, and other database capabilities.*" [OMG, 2001].

The goal is to enable the representation of entities, modelled in standard entity-relational diagrams, directly in OO databases [Pokorny, 1999], with adding structured data types and functions for their manipulation, including features as encapsulations, methods and inheritance, as this is the case in OO programming languages. This would enable a faster development time, custom data types and non need of table normalisation in the development cycle.

Until now, there were only few attempts in the field of spatial OODBMS, such as Jasmine, O_2 [Scholl, Voisard, 1992] or GeO_2. One of the main disadvantages of pure OODBMS, until now, is a lack of a standardised and powerful query language, with OQL still being in development . Those databases are nice proofs of concept, but are rarely used in normal applications due to very complex functionality and difficult use. The usual approach is a custom made DBMS programmed in an OO programming language using its capabilities for data management.

We can expect that the problem of complex features could be well solved in OODBMS, where we would store primarily whole objects and their geometric elements will be only subclasses of those. This is a fundamentally different approach from ORDBMS, where we have to use a bottom-up building approach.

Another issue that will be addressed by OODBMS is a wider implementation of scale free datasets for rendering customized “map views”, based on the same date, only viewed on different detail levels. Until now, we are storing enormous amounts of redundant data worldwide, with different semantic data description or mapping accuracy levels, but still describing the same entities of the real world. By implementing the highest detail in the database and enabling the DBMS to filter the data relevant to the user/client and its rendering capabilities, we could assure higher consistency between different geographic applications, from cadastral to topographical mapping ones.

4.8 Implementations

First attempts to extend the traditional relational schema (originates from the 1970s from the work of Codd) were done in the DBMS POSTGRES (UCLA, Berkley)[Stonebraker, Rowe, 1986] and IBMs STARBURST. The resulting data model is often called Object-Relational, because it enables object-like embedded relations – all attributes, spatial and non-spatial are stored in the same tables, using approaches similar to well known object oriented programming methods.

The development led to several attempts to enlarge the existing relational data model employed to model reality in the spatial DBMS, an overview is presented in [Lemmen , 2003].

As following to the implemented data model, also the data querying tool had to be extended to meet the needs of analysing, storing, management and retrieval of spatial data. Several query languages were developed and lead to commercial, non portable extensions of the SQL standard. An example can be the PL/SQL query scheme of Oracle – it is an “SQL with Geometry Types” , or also “implementation of spatial feature tables in the OpenGIS ODBC/SQL specification for geospatial features”. However, Oracle extends the standard in several ways – for example, more feature types are implemented, such as arc, circles and arc strings, which can cause problems while trying to interchange data with other DBMS. The implementation of the methods for handling spatial features also have different syntax, even if most of them are very similar with the standard. However, as mentioned in [Lemmen, 2003], none of the market leading vendors implements the SFS purely. Currently, PL/SQL of Oracle became a reference implementation of the OGC Simple features specification for SQL. Being one of the most commonly available implementations with good availability to the commercial and educational sector, it will be used as reference in the examples below. Oracle Spatial 9.1 and 9.2 was also used during the work on this thesis – see chapter 6.

5 System Architecture of LBS

Location based services are a specific type of web based applications. Until now, we were only concerned by the positioning of the MU and a suitable storage of the data, used by the service. By definition, we are accessing data on a remote basis from different sources. This has several drawbacks to the architecture of mobile GIS technologies and their architecture design. It is a client/server based architecture with a variety of possible types of clients (as shown on Figure 2.2), using internet technology and networking for data access, query and update and having a part of the data transmission performed by wireless means.

5.1 Communication Technology for LBS

For a communication link may be established, it is necessary to assure that:

- The client can address the source of the data (the “service”), on a server. This means it knows “where” to find it and how to “ask” for the correct data (knowing the data model).
- The client is able to communicate with the server – communication protocols and query standards.
- The client is able to read the reply from the server and render it on the screen.

In the area of LBS, one more brick can be added to the architecture of the system – the position sensor readout needs to be encrypted in a known “language”.

The communication in networks is performed trough a variety of different protocols – communication standards, or “languages”. Those are TCP/IP based protocols, from lover level protocols dealing with signal processing, to and application protocols, used for data encoding and communication, such as HTTP.

5.1.1 The Thin Client Concept

Servers are highly powerful computers (or programs running on server computers), that store the application’s data and process them on client computer’s/ application’s requests, and send back replies to the queries trough networks. The use of open standards and open architecture allows different kinds of client devices to connect to servers and communicate with them, trough messages encoded in open standard languages, such as the Extensible Markup Language (XML).

The expansion of network applications and internet to the wireless world brought the problem of insufficient computational power on the client side on scene. It was solved by the thin-client architecture - Figure 2.2. The client (PDA, mobile phone) sends only basic requests for information to the server, where it is processed, the correct information is retrieved, formatted, rendered (if necessary) and only the output is sent to the client. So, only very basic operations have to be performed at the client side. Although the term thin client usually referred to software, it is increasingly used for handheld systems in totality, computers and applications.

5.1.2 GSM Network Architecture

As shown on Figure 5.1, the GSM network is a distributed, tree-like hierarchy, where area called Cells are served by a single BTS. Several BTS are managed by a Base Switching Centre (BSC and those are relayed to the network centre. This is a central processing unit of the whole network, with a system of registries ("databases"), managing user account, authentication, billing and other network tasks. More details on GSM architecture can be found in the web page on www.gsmworld.com. GSM networks were primarily designed for voice communication, in the times when the internet was still in the developing phase and nobody thought about wireless applications and data transmission trough wireless networks.

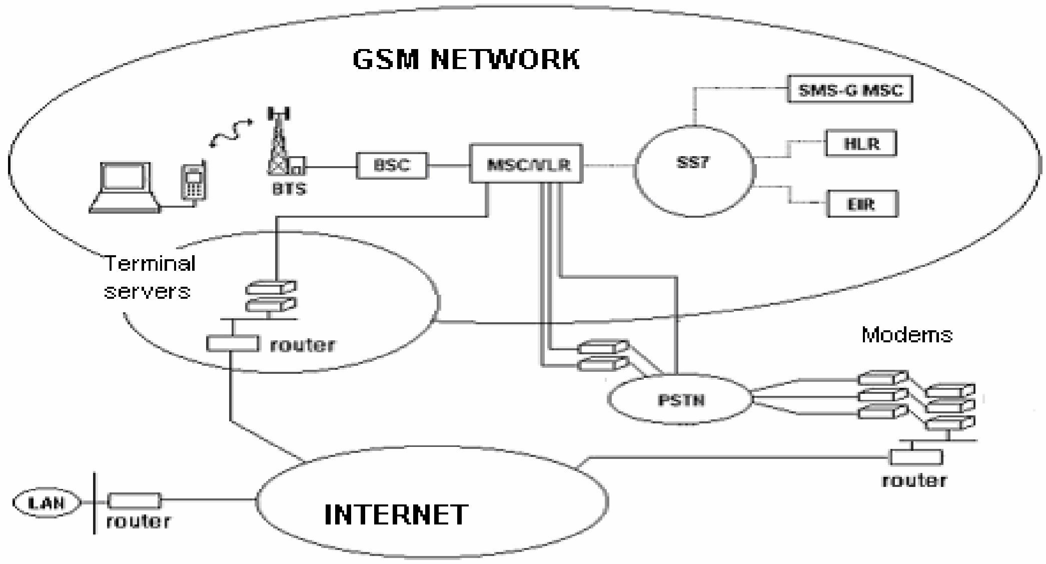


Figure 5.1 – GSM network architecture

Therefore, the next generations of GSM networks are enhanced with Internet gateways that enable seamless connection to the Internet. But this improvement is not sufficient for more complex wireless internet applications, as far a the standard GSM data transfer speed is only around 9.6kb/s.

Hence, the GSM 2.0 generation networks decreased the amount of time slots reserved for checksums and using High Speed Circuit Switched Data (HSCSD) transfers at the speed around 56kb/s [www.mobil.cz].

Thus, GSM 2.5 generation networks, after adding several new components to the network structure [www.mobil.cz], have implemented a technology based on packet data transmission. The client is not occupying the whole time slot for its data transfer, but instead is sending the data in packets of fixed size trough the best fitting path in the network. The technology is called general Packet Radio Service (GPRS) and enables asymmetric transmission (at lower speed for upload and faster for data download) at speeds up to 171.2kb/s. Another major improvement is that the application can be online non-stop and the user is not paying per second spent on data transmission, but per transmitted amount of data.

The architecture of GSM 2.5 generation networks is totally sufficient for LBS with MU based positioning. The location description (coordinates) are embedded in the query as a query parameter and sent to the mobile GIS application on the network, where it is processed as a standard spatial query by the target services and databases and a reply is generated. In fact, it is a standard web service without direct impact on the GSM network. This is also the case of the Miracle Pilot application, shown in chapter 6.

5.1.3 Adaptations of GSM Networks for MT-LR Services

For LBS profiting of network based positioning, or “dispatching” mobile GIS applications, where the network transmits the location information to a desktop computer, where it is used for emergency, monitoring or fleet management LBS, further modification of the wireless networks are necessary.

In the original GSM network design there is no place neither for any processing unit and location registry for calculating and storing positions of located MU, nor gateway providing this information to external service. This is solved, depending on the LDT, by several basic elements – LMUs, assisting the location determination, additional functionality in the Serving Mobile Location Centre (SMLC), closely cooperating with the central computing power of the network, the Mobile Switching Centre (MSC). The locations of the MU determined in the SMLC are then stored and made available to other GSM networks or LBS providers trough the Gateway Mobile Location Client (GMLC). This configuration is shown on Figure 5.2 that originates in the ETSI technical specification, [ETSI, 2002].

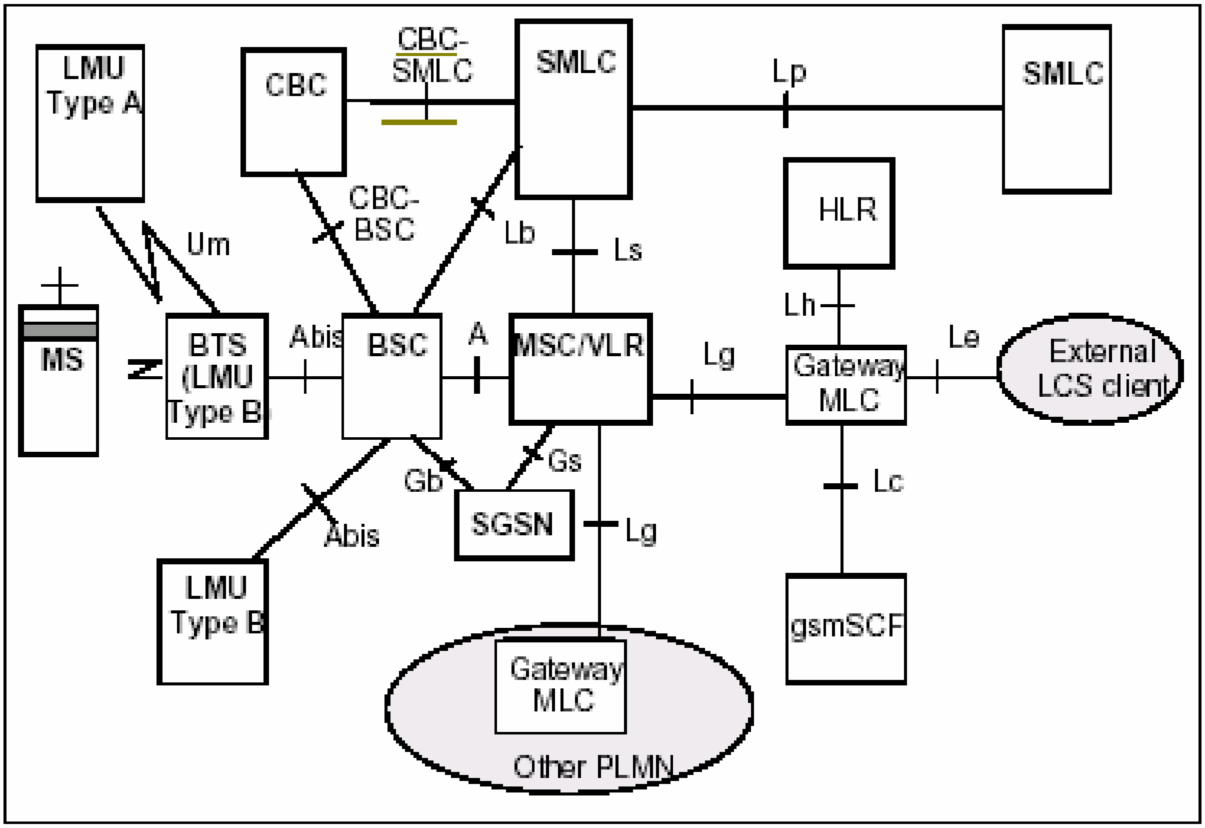


Figure 5.2 – GSM architecture for Location Based Services [ETSI, 2002]

5.1.4 Standards for Location Information Interchange

An implication arising from the architectures mentioned, some standards for position encoding and communication are necessary. The standardization effort have begun and several bodies are concerned with different parts of the problem. The full description of the standards is out of the extent of this thesis, meanwhile, here are the most important ones:

The ETSI standards are taking care of standardizing the internal protocols, network architectures and configuration and basic LDT in the GSM networks. These standards are closely cooperating with the OpenGIS Consortium's OpenLS working group, aiming to standardize the components of the so called "Geomobility Server", or, the server, providing spatially enabled services needed for LBS. These are standardized as "Core Services" and are represented by the forward and reverse geocoder, routing, directory, presentation and gateway services. OpenLS is specifying the necessary attributes of the messages exchanged between those services, as well as the the output from the gateway service out to third parties. The standards consists of specifications of message specification (request and reply), message elements, encoding up to the Abstract Data Types (ADT) necessary for XML messages, and error handling [OGC, 2003], [Niedzwiadek, 2002].

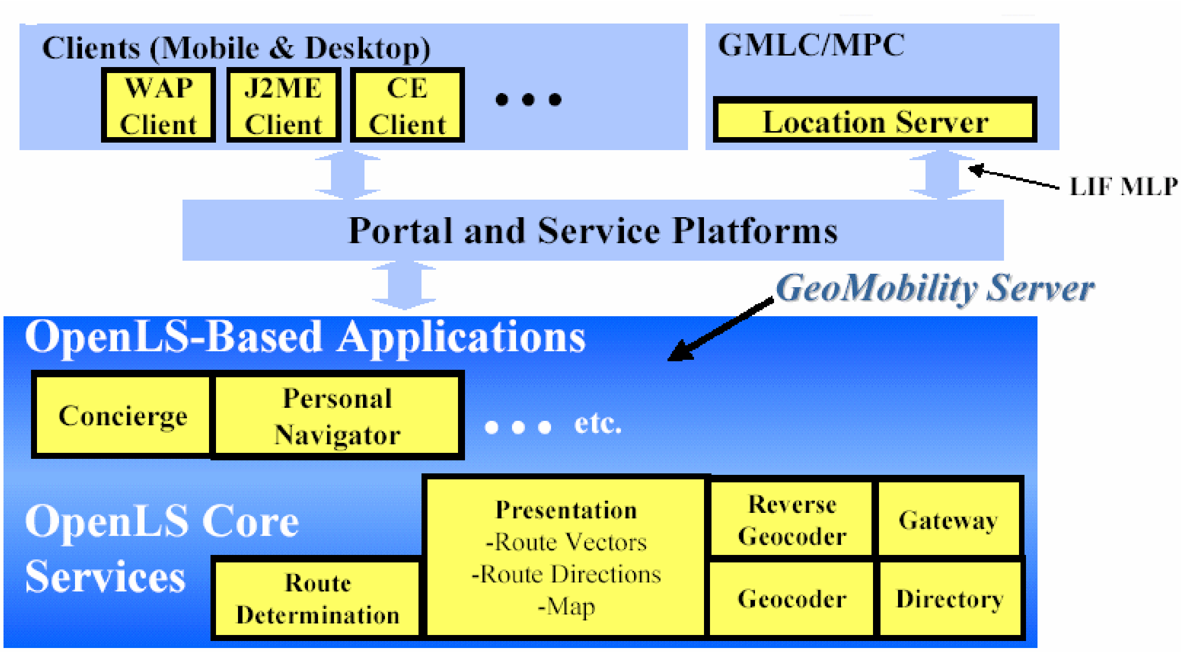


Figure 5.3 – OpenLS Geomobility Server and LIF MLP integration [LIF, 2001]

On the other end of the chain, between the network and the OpenLS specified Geomobility Server, stands the standards specified by the Location Interoperability Forum (LIF) – the Mobile Location Platform, aiming to standardize the location request and reply from GMLSs of the wireless carriers, as well as the LDTs that should be implemented for various services - Figure 5.3. There the situation is more complex, as far as numerous vendors, such as Ericsson, Siemens and others begun to implement proprietary solutions beforehand [LIF, 2001].

It is important to mention additional standards of the OpenGIS consortium, mainly the OpenGIS Web Map Server Interface Implementation Specification [OGC, 2000], that states how to implement web based mapservers to access spatial data and render them to the client. Amon other, this specification also defines the encoding of a URL request to the mapserver service. This is the kind of queries that the LBS client is sending to the web service (for example the one that could be implemented in the MIRACLE project later on). Here a brief example:

```
http://jolly-
jumper.com/wms/sclmapserver.exe?wmtver=1.1.0&layers=uk_level1_0a,uk_level1_0b,
uk_level1_1Ob,uk_level1_1pb,uk_level1_1q,fleetonline&srs=12345&bbox=-
1061841.9579,-4460164.4999,874804.1769,-
3236624.4999&width=516&height=326&format=png&transparent=true&exceptions=inimage
&t=317&request=map&sclvars=Databasename:father,Username:Myname,Password:****,Tab
leName:map_provider_data_transform,IconColumn:accuracy,TransActionID:'B'
```

The example combines the access to raster maps and Oracle stored vector layers (fleetonline), where the parameters of the query can be easily identified, as well as the corners of the query window, the output window (screen) parameters and the used SRS.

Finally, the IETF, along with its member companies, has also proposed another standard called Spatial Location Protocol (SLoP). The aim of the SLoP is to address the following question: How can an application acquire the spatial location of an identifiable resource over/represented on the Internet, in a reliable, secure, and scalable manner? This protocol will specify an absolute location on the Earth and will use WGS84 geodetic datum as default reference system. The format of location information is ideally composed of the following data items (provided capabilities for certain items are available) [Srivatsa, 2002]:

- User location type (for example, absolute/descriptive location)
- Framework (for example. WGS84, UTM)
- Syntax/format (for example, longitude, latitude, altitude in degrees)
- Geocentric position
- Accuracy
- Time stamp (date, time, time zone)
- Time-to-live
- Others (direction, velocity, orientation, etc.)

6 Pilot Mobile GIS Application Design

The area of LBS is incredibly rich and allows the development of simple services, as well as sophisticated applications for a high number of heterogeneous users. As it was shown, the use of spatial databases can simplify the whole system architecture by taking care of a large part of the analytic and data processing tasks. If existing standards are used, and the system profits of a modular architecture with standalone, autonomous modules, designed to take care of an elementary problem, the deployment of such a system can be achieved in a relatively short time.

The capacities of integrated spatial databases, implemented in a mobile GIS application, will be demonstrated on two examples – a demonstration, pilot application, designed the author in a very short time as a “proof of concept” and experiment, with MO - LR, and a real, fleet management LBS, based on OpenGIS standards, with MT-LR.

6.1 MIRACLE

The ideal synergy of the wireless internet, LDTs and spatial databases enables relatively easy and fast development of simple LBS. This will demonstrated on the example of the model architecture of Mobile Internet Real-estate’s Agent Cadastral Location Explorer (MIRACLE) application, a simple LBS that was designed thanks to the cooperation with the GISt section of the Department of Geodesy, TU Delft. The goal was to study and design the architecture of a simple, MO-LR, MU LDT based LBS service that will exploit the capacities of a spatial database (Oracle 8 Spatial) with a sample Cadastral dataset of the Netherlands. For this only a specific subset of data was created with no administrative data, to conform with legal requirements. The attempt to achieve a real functionality was planned, and a successful result would be the MIRACLE.

6.2 MIRACLE Use Case

The use case for this simple LBS was defined as follows:

There is a need to get the administrative data about the specific parcel, within the borders of which the user is standing. The user only wants to automatically input the position as a single point inside of the parcel boundaries, and the system should return at least the parcel identification. There is no need to format the reply or to render a graphical representation of the parcel, as it would be in a real-life system. Furthermore, the problematic of positioning accuracy, and its enhancement trough consecutive positioning averaging methods was not treated in this pilot.

6.3 MIRACLE System Requirements

The architecture design needed to fulfil several requirements:

- Be a modular system with modules performing standalone tasks and performing as embedded entities for easy modifications and extensions in the future.
- Exploit the capabilities of a Spatial database (Oracle 8 Spatial) and the specified dataset and avoid using middleware GIS packages.
- The modules should communicate trough open or documented standards, usual for the industry.
- The implementation should be tried in Java to enable system portability to embedded systems.

Therefore, the basic tasks of the application were identified as:

- get the coordinates from the GPS receiver,
- construct the query for the database,
- send the query to the HTTP server trough the network and process it to the correct application server (servlet),
- catch the query by the servlet, retrieve the query parameter (coordinates in WGS84), modify the query for database query (coordinate transformation to Royal Dutch SRS (RD)),
- reconstruct the topology and fetch the data of the parcel from the dataset,
- send a reply to the client.

These requirements were transformed in the implementation design of tasks of every component, to fit as much as possible the concept of thin-client architecture and were traduced to task specifications for each of the components:

6.4 MIRACLE Architecture

After the components design process was completed, an implementation design of the architecture, where all the component were given their specific place in the chain, was created. The schema of the architecture is shown on Figure 6.1 and precedes the description of the components only for easier understanding of the system. In the development process, the workflow was inverse.

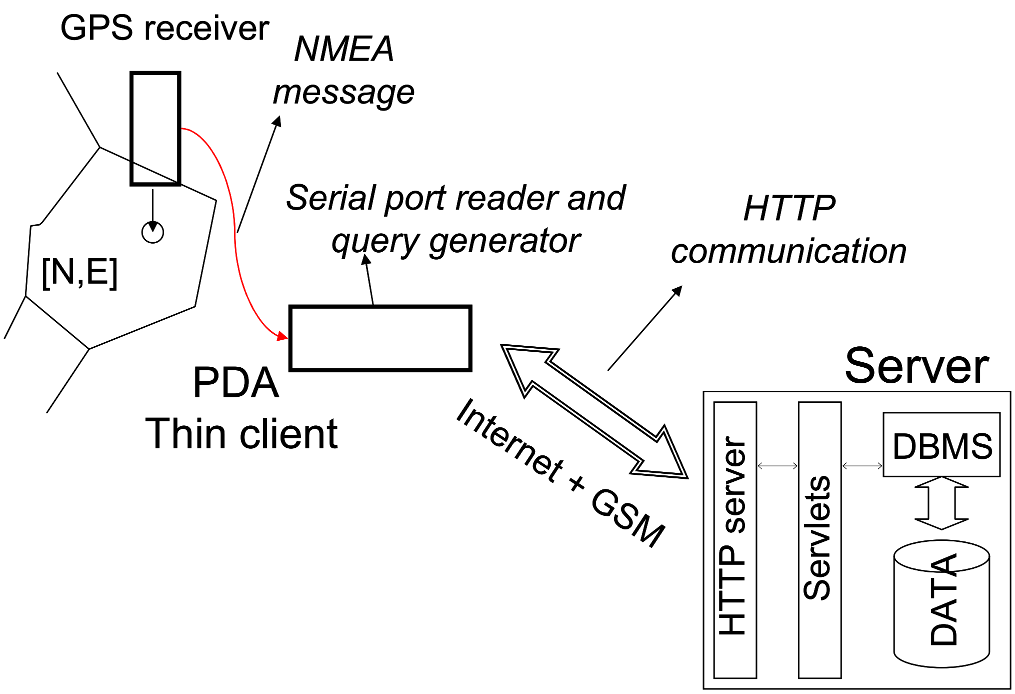


Figure 6.1 – MIRACLE implementation architecture

The components were identified in the order following the data flow in the system. Following chapters provide an overview of all the components (modules).

6.4.1 Location Retrieval

NMEA reader: A module is needed to read the GPS receiver output. As far as we exploit a MU based LDT (GPS), the message will have to be read at the client side. The possibility to use a well documented, even tough not open, standard for sensor message encoding in navigational devices, build by the National Marine Electronics Association (NMEA), dubbed NMEA 0183. It was created as a standard electrical interface and data protocol for communications between marine navigational instrumentation and later spread to the majority of navigational instruments both naval or land and airborne [Benett, 2000], [www4.coastalnet.com/nmea]. This standard was selected for easy deciphering (pure ASCII encoding enables reading of the message by any programming language on any platform), RS232 (serial port) transmission, available on all handheld devices and the support of this format by all the handheld GPS receivers - Table 6.1.

```
$GPRMC,122930,V,5159.2069,N,00423.2772,E,10.8,0.0,270302,1.1,W,S*3A
$GPRMB,V,,,,,,,,,,,,A,S*0E
$GPGGA,122930,5159.2069,N,00423.2772,E,8,12,2.0,-44.8,M,46.8,M,,*5E
$GPGSA,A,3,02,03,08,15,17,18,21,22,23,26,27,31,3.6,2.0,3.0*35
$GPGSV,3,1,12,02,56,250,48,03,78,283,50,08,09,336,37,15,45,061,47*70
$GPGSV,3,2,12,17,30,058,44,18,46,087,47,21,45,141,47,22,37,186,45*7B
$GPGSV,3,3,12,23,17,036,40,26,09,036,36,27,15,306,39,31,36,290,45*72
$GPGLL,5159.2069,N,00423.2772,E,122930,V,S*49
$GPBOD,,T,,M,,*47
$PGRME,15.0,M,22.5,M,27.0,M*1A
$PGRMZ,-147,f,3*34
$PGRMM,WGS 84*06
$GPRTE,1,1,c,*37
```

Table 6.1 – Example of a Garmin eTrex single NMEA block output

NMEA 0183 are output every second and therefore enable “real-time” navigation applications to be developed. The output is a block of a variable number of sentences, all of a 37 byte length (non-occupied data fields are filled by NUL bytes), each beginning with a $ sign followed by the sentence identification (“talker ID”) code consisting of 5 characters. The sentence ends by a carriage return/line feed.

Their output is a final information and no further processing or calculations are necessary to extract the position. After reading the NMEA0183 output, the module only needs to extract the Northing and Easting information (WGS84, in the format DDDMM.MMMM) from their respective place in the NMEA block/sentence. This can be done from several NMEA sentences - $GPGGA (Global Positioning System Fix data), $GPGLL (Geographic Position, Latitude and Longitude), $GPRMS (Global Positioning – Recommended Minimum Specific GPS/Transit data), and others. Every manufacturer can implement it own sentence, provided that it fits to the standardised format. An extensive list and description of the most common NMEA sentences can be found in [Bennett, 2000]. The MIRACLE project used the $GPGGA sentence, as far as it is a very common implementation for output encoding in most of the handheld GPS devices of the majority of manufacturers. This assures higher flexibility to the system.

As the Table 6.1Table 6.2 shows, the sentence consists of several, comma delimited, fields interesting for LBS:

$GPGGA,122930,5159.2069,N,00423.2772,E,8,12,2.0,-44.8,M,46.8,M,,*5E	
Field No	**Content**
1	Time fix in UTC, HH:MM:SS
2	Latitude (DDMM.MMMM), N (hemisphere)
3	Longitude (DDDMM.MMMM), E (hemisphere)
4	Fix quality (0-invalid/1-GPS fix/2-dgps fix/8 – not available)
5	Number of tracked satellites
6	HDOP
7	Altitude above see level, M (metres)
8	Height of geoid (mean see level) above WGS84 elipsoid, M (metres)
9	Empty field
10	Checksum

Table 6.2 - $GPGGA sentence structure

Other sentences, such as $GPRMC, provide also additional information – speed over ground, heading, magnetic variation, and others. Those can very well be implemented in more complex navigational applications.

For MIRACLE purposes, the readout of the output of the GPS receiver was performed in Java 1.2 using an additional (but standard) package javax.comm for reading computer ports, downloadable from http://java.sun.com. The demo package BlackBox.jar was used to read the device trough the RS232 serial port of a personal computer and a sample of the output is in Table 6.1.

6.4.2 Data Storage and Queries

The DBMS used for the MIRACLE project was Oracle 8i Spatial. Spatial supports the object-relational model for representing geometries. The object-relational model uses a table with a single column of MDSYS.SDO_GEOMETRY and a single row per geometry instance.

From the OGC data model, the Oracle implementation differs adding the notion of layer. Layers are composed of geometries, which in turn are made up of elements. A layer is a collection of geometries having the same attribute set. They are grouped into layers by defining the set (combination of attributes – rows in the implementation of a database) of characteristic attributes.

Point data consists of one coordinate. Line data consists of two coordinates representing a line segment of the element. There are always two vertices encoded in the line definition, given in an order first – last vertex, when the topology is reconstructed. Polygon data consists of coordinate pair values, one vertex pair for each line segment of the polygon. In polygon rings, the lines/polygons are defined in order around the polygon (counter clockwise for an exterior polygon ring, clockwise for an interior polygon ring).

The data used for the project were stored in a single table "mylki" and were created on purpose as a subset of the area of Delft, The Netherlands. It was a subset of cadastral data, without administrative data except of the Parcel ID. The geometry has been reconstructed by add on topological function in the original database and was stored in the mylki table - Table 6.3.

The geometric data model is based on the winged edge topology. This means, that no complete parcels are stored, but only one parcel boundary is assigned to the parcel number. The reconstruction of the whole polygon that is the boundary of the parcel is than realised by finding the adjacent boundaries in a counter clockwise manner. When the algorithm comes to a fork, it selects the most left connection (right hand) and continues clockwise. So, the algorithm is able to reconstruct the whole polygon – parcel boundary and display it. This algorithm is also able to handle holes – they have the clockwise rotation if compared to the outside polygon. This assures that the inside of the area will be again the right face. This is a very good way how to omit redundancy and disproportions in geometric data. There is no possibility of having gaps, free space, and different boundaries. The topological purity is achieved [Tijssen, Quak, 2001] - Figure 4.4.

```
SQL> describe mylki
 Name                         Null?    Typ
 ------------ -------- ----------------------------
 OGROUP                       NOT NULL NUMBER(11)
 OBJECT_ID                    NOT NULL NUMBER(11)
 SLC                          NOT NULL NUMBER(11)
 CLASSIF                      NOT NULL NUMBER(11)
 LOCATION                     MDSYS.SDO_GEOMETRY
 GEO_COLOR                    NUMBER
 Z                            NOT NULL NUMBER(11)
 D_LOCATION                   MDSYS.SDO_GEOMETRY
 ROTANGLE                     NOT NULL NUMBER(11)
 ACCU_CD                      NOT NULL NUMBER(11)
 OAREA                        NOT NULL FLOAT(126)
 GEO_BBOX                     MDSYS.SDO_GEOMETRY
 OBJECT_DT                    NOT NULL NUMBER(11)
 TMIN                         NOT NULL NUMBER(11)
 TMAX                         NOT NULL NUMBER(11)
 SEL_CD                       VARCHAR2(3)
 SOURCE                       VARCHAR2(5)
 QUALITY                      VARCHAR2(2)
 VIS_CD                       VARCHAR2(1)
 AKR_AREA                     NOT NULL FLOAT(126)
 MUNICIP                      NOT NULL VARCHAR2(5)
 OSECTION                     NOT NULL VARCHAR2(2)
 SHEET                        NOT NULL VARCHAR2(4)
 PARCEL                       NOT NULL VARCHAR2(5)
 PP_I_LTR                     NOT NULL VARCHAR2(1)
 PP_I_NR                      NOT NULL VARCHAR2(4)
 L_NUM                        NOT NULL NUMBER(11)
 LINE_ID1                     NOT NULL NUMBER(11)
 LINE_ID2                     NOT NULL NUMBER(11)
 X_AKR_OBJECTNUMMER           NOT NULL VARCHAR2(17)
 GEOMETRY                     MDSYS.SDO_GEOMETRY
```

Table 6.3 – mylki table

The reconstruction of the geometry for the need of the mylki table was performed by means of the following statement:

```
drop table mylki;
create table mylki
as
   select m.*,createpolygonfast(m.object_id) as geometry
   from lki_parcel m
;
```

where “createpolygonfast” is a stored method, calling java classes for execution of the geometry reconstruction. This was created by Dr. Wilko Quak at TU Delft for previous projects.

The retrieving of a parcel can be done in several manners:

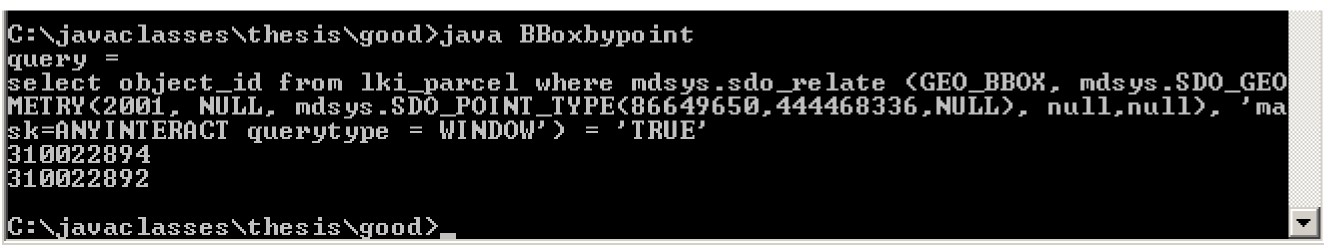
- by giving the precise parcel identification (name),
- providing the minimum bounding box of the parcel and searching the best matching case,

- by providing a point, that lies in the parcel and reconstructing its boundaries, after that the correct parcel can be identified.

The second case is a typical query for primary – approximate search that can produce multiple records as output. Every record in the mylki table has a Geo_Bbox attribute. This is the minimum bounding rectangle of the parcel, created for the Oracle 8i Spatial R-tree index. This can help us to create a candidate subset of parcels while performing window queries, but also for primary search when creating a parcel subset for faster parcel retrieving for the MIRACLE project. This is a case of BBoxes of parcels with concave boundaries, where smaller parcels can fall in the same BBox:

```
Query: SELECT object_id FROM mylki WHERE inside
(point('(242360803,512476153)'), geo_bbox) = 1
```

A variation of this query can also look as follows (used on the lki_parcel table, the original dataset) - . This call was performed trough Java Database connectivity (JDBC) connection from Java, to show that external programs can easily access data stored in Oracle. This JDBC query () can be applied for real implementation on embedded devices. As we can see, the query generate two parcel IDs as output.



```
C:\javaclasses\thesis\good>java BBoxbypoint
query =
select object_id from lki_parcel where mdsys.sdo_relate (GEO_BBOX, mdsys.SDO_GEO
METRY(2001, NULL, mdsys.SDO_POINT_TYPE(86649650,444468336,NULL), null,null), 'ma
sk=ANYINTERACT querytype = WINDOW') = 'TRUE'
310022894
310022892

C:\javaclasses\thesis\good>_
```

Picture 6.1 – Console output of JDBC query to Oracle.

If we use a similar query, using the newly generated geometry, that stores reconstructed geometries in the mylki table (the example also shows how the PL/SQL query is encoded in the Java class) - :

```
import java.sql.*;
import oracle.jdbc.*;
class BBoxbypoint2
{
  public static void main (String args [])
       throws SQLException
  {
    // Load the Oracle JDBC driver
```

```
    DriverManager.registerDriver(new oracle.jdbc.OracleDriver());

    // Connect to the database
    // You must put a database name after the @ sign in the connection URL.
    // You can use either the fully specified SQL*net syntax or a short cut
    // syntax as <host>:<port>:<sid>.  The example uses the short cut syntax.
    String url = "jdbc:oracle:thin:@www.gdmc.nl:1521:geobase";
    String userName = "tomko";
    String password = "martin";

    if (args.length > 0) url = args[0];
    if (args.length > 1) userName = args[1];
    if (args.length > 2) password = args[2];

    Connection conn =
      DriverManager.getConnection (url, userName, password);

    // Create a Statement
    Statement stmt = conn.createStatement ();

    // Select the nazov column from the test table
    String s = "select object_id " +
                        "from mylki " +
                              "where mdsys.sdo_relate (GEOMETRY, " +
                        "mdsys.SDO_GEOMETRY(2001,                          NULL,
mdsys.SDO_POINT_TYPE(86360615,444736790,NULL), null,null), " +
                        "'mask=ANYINTERACT querytype = WINDOW') = 'TRUE' ";

    System.err.println("query =\n" + s);

    ResultSet rset = stmt.executeQuery (s);
    // Iterate through the result and print the employee names
    while (rset.next ())
      System.out.println (rset.getString (1));
  }
}
```

Table 6.4 – JDBC query for reconstructed geometry

the output will be different:

```
C:\javaclasses\thesis\good>java BBoxbypoint2
query =
select object_id from mylki where mdsys.sdo_relate (GEOMETRY, mdsys.SDO_GEOMETRY
(2001, NULL, mdsys.SDO_POINT_TYPE(86649650,444468336,NULL), null,null), 'mask=AN
YINTERACT querytype = WINDOW') = 'TRUE'
310022894

C:\javaclasses\thesis\good>
```

Table 6.5 – Bboxbypoint2 console output.

As mentioned before, the query input is a coordinate pair. For mylki, coordinates needed to be in RD, but the source from the GPS was in WGS84. Therefore, a Java class was used to transform the coordinates. More details can be found in chapter 6.4.3.

6.4.3 Java Implementation

The two basic components of the MIRACLE architecture, as shown beforehand, realise the most important tasks of the application – location retrieval and database querying. The example shown that no special middleware GIS application is needed to perform spatial analysis and that integrated databases are perfectly apt to fulfil a majority of needs of LBS.

However, a connection between the server and the client must me implemented somehow. MIRACLE uses Java for its purposes, as it has already be shown in some of the previous examples. The choice was made according to several characteristics – the language used for LBS should portable (compile one, run anywhere – on any OS platform), have low resource expectation, be OO for the possibility to use modular architecture. And finally, it should be supported by numerous embedded devices (Phones, PDAs, Palmtops) for easy implementation. All this is assured by Java and it different modifications.

Java enables several types of implementations – applications, applets (running inside web browsers), servers and server pages (running on servers), and middlets (running on embedded platforms). Mostly everywhere the core of the code is the same.

For the examples used above, Java classes used as standard "applications" were used. However, attempts to run these classes on the PDA available – Nokia communicator 9210 with Symbian Crystal OS were not successful. The preparation of applications for embedded platforms requires more experience and understanding of different OS and programming approaches, and this was out of the extent of this Thesis.

For the implementation of the architecture, several Java components will be needed.
On the client side, this will be:

1. The NMEA reader/parser, to extract the coordinates pair from the NMEA message.
2. A JDBC query class that would send the query trough internet to the "application server" and read the reply.

On the server, the implementation will consist of:

1. a servlet, installed in a servlet container joint with a http server or native Oracle AS, further called "application servers", transforming the WGS84 coordinate pair to RD and processing the query atop the database, as shown in the JDBC example.

2. Optional, a DB reply transformer, creating XML replies.

6.5 Further Improvement for Complex LBS

The MIRACLE project is a simple mobile GIS application and doesn't provide all the possibilities that LBS can provide to users. This includes map rendering on screen, real internet based querying and the use of network based LDT. The implementation of such functionality would affect the architecture of the application in several details. The full extent of this problematic cannot be created in this thesis, but some guidelines were given.

Usually, the location in network based applications is retrieved from GMLC gateways. Those are accessed trough XML based queries that can be issued for example by java servlets. An example of the architecture is shown on Figure 6.2.

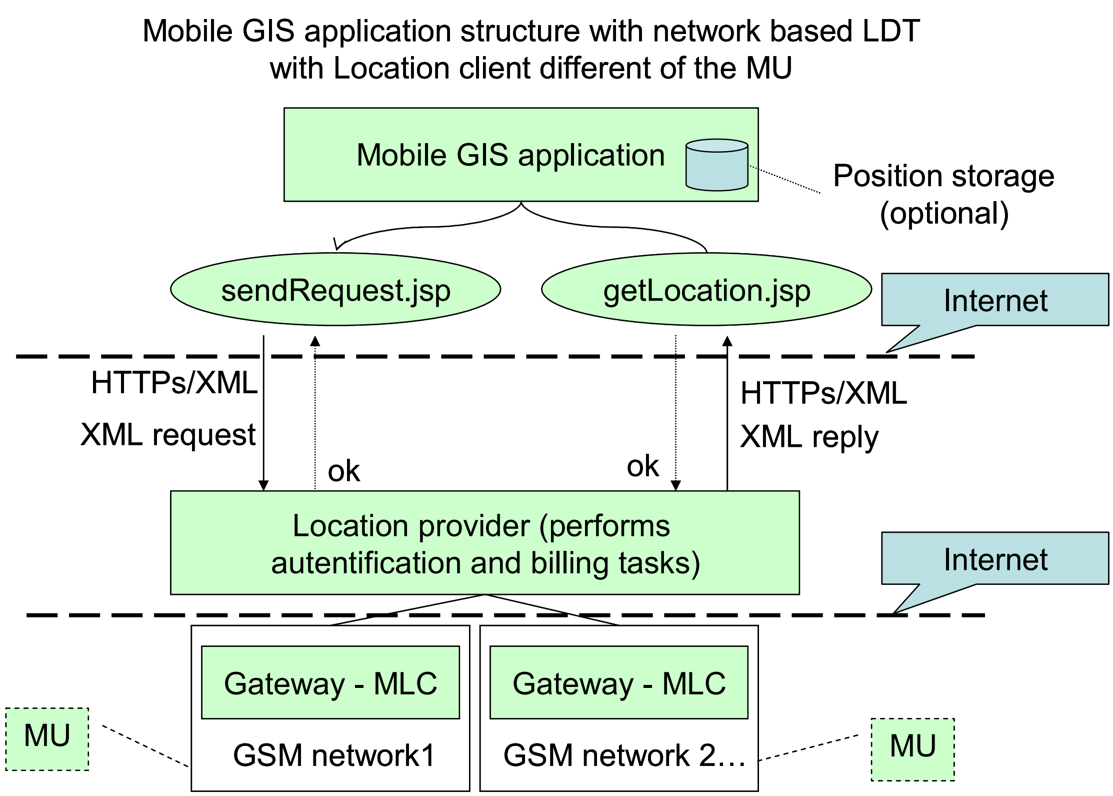


Figure 6.2 - Example of LBS with network based LDT architecture

The standardisation of those queries was mentioned earlier in this paper. Here is an example how such a query can be issued - Table 6.6:

```
<?xml version=""1.0"" ?>
<LBSRequestMessage>
      <LBSAuthentication>
            <Username>myUsername</Username>
            <Password>myPassword</Password>
            <AccountRef>myAccountRef</AccountRef>
            <RequestID>1001</RequestID>
      </LBSAuthentication>
      <LBSMessageData>
            <Operator>TM</Operator>
            <Subscriber>447766554433</Subscriber>
      </LBSMessageData>
      <LBSGateway>
            <Response>00</Response>
            <TransactionID>1243-5423-9630-af39</TransactionID>
            <ErrorText></ErrorText>
      <LBSGateway>
</LBSRequestMessage>
```

Table 6.6 – XML encoding of a location query

Such a query is very easy to parse by any contemporary application of DBMS and therefore the output can be easily inserted to the database. The following shows the full list of steps necessary to create a spatial table in Oracle, including a table that typically stores outputs form a location gateway. This table is created based on a previous spatial table with geometries stored in the WGS84 SRS. The new table stores geometries in Lambert azimuthal SRS. The example shows the capabilities of Integrated databases to transform the coordinates of the stored data. This could also be used in the MIRACLE project and it would lead even proper implementation of the architecture.

The script creates the table MAP_PROVIDER_DATA_TRANSFORM based on MAP_PROVIDER_DATA with one additional column LOCATION_NEW, which is a transformed copy of the LOCATION column with srid = 106496. The Metadata (USER_SDO_GEOM_METADATA) is also filled for both columns, and two spatial indices, one on LOCATION and one on LOCATION_NEW, are created - Table 6.7.

```
Martincall SDO_CS.transform_layer(
         'MAP_PROVIDER_DATA',
         'LOCATION',
         'MAP_PROVIDER_DATA_TEMP',
         106496)
/
create table MAP_PROVIDER_DATA_TRANSFORM as
```

```
   select t.*, u.GEOMETRY as LOCATION_NEW
     from MAP_PROVIDER_DATA t,
       MAP_PROVIDER_DATA_TEMP u
     where u.SDO_ROWID = t.rowid
/
drop table MAP_PROVIDER_DATA_TEMP
/
delete from USER_SDO_GEOM_METADATA
   where table_name = 'MAP_PROVIDER_DATA_TRANSFORM'
/
insert into USER_SDO_GEOM_METADATA(table_name, column_name, diminfo, srid)
   select
     'MAP_PROVIDER_DATA_TRANSFORM',
     'LOCATION',
     diminfo,
     srid
   from USER_SDO_GEOM_METADATA
   where table_name = 'MAP_PROVIDER_DATA'
     and column_name = 'LOCATION'
/
insert into USER_SDO_GEOM_METADATA(table_name, column_name, diminfo, srid)
   values(
     'MAP_PROVIDER_DATA_TRANSFORM',
     'LOCATION_NEW',
     MDSYS.SDO_DIM_ARRAY(
       MDSYS.SDO_DIM_ELEMENT('X',  -450000,   110000, 10),
       MDSYS.SDO_DIM_ELEMENT('Y', -4445000, -4244000, 10)),
     106496
   )
/
commit
/
create index MAP_PROVIDER_DATA_TRFM_IDX_1
   on MAP_PROVIDER_DATA_TRANSFORM(LOCATION)
   indextype is MDSYS.SPATIAL_INDEX parameters('SDO_INDX_DIMS=2')
/ create index MAP_PROVIDER_DATA_TRFM_IDX_2
   on MAP_PROVIDER_DATA_TRANSFORM(LOCATION_NEW)
   indextype is MDSYS.SPATIAL_INDEX parameters('SDO_INDX_DIMS=2')
/
```

Table 6.7 – Creation of Spatial tables in Oracle, with metadata and index entries

6.6 *Miracle Summary*

The Oracle Spatial DBMS proves that the implementation of LBS using this database can be relatively easy and straightforward. The cooperation with external applications, such as Java servlets and other internet based services is very good and additional functionality to the capacities of Oracle can be solved this way.

A real development of a thin client based LBS would require more time and experience in the field of embedded devices, as far as the compile once – run anywhere concept of Java is somehow degraded by the lack of several standard Java libraries in the embedded versions of Java for PDAs.

7 Conclusion

Integrated spatial databases present a modern and efficient way of storing data for various spatial enabled applications. With the spread of distributed, internet based services and wireless communication, it became possible to think about accessing spatial data in the field and personalize the service provided to the user upon spatial context. The mobile GIS technology is profiting of the storage and analytical capabilities provided by integrated spatial databases to simplify the development of interoperable, modular applications for LBS.

The assets of this approach are huge. File based storage of geographical data, used in conjunction with relational databases without geometry data types, produces problems with spatial queries, consistency control, and data management. Attempts to bring traditional file based vector datasets on embedded devices, and develop mobile GIS applications proved, that serious problems arising of data synchronization. Locally stored copies of geographical files were updated in the field and afterwards needed to be synchronized in the office with the original datasets, or with updates performed by colleagues. The lack of tools, typical for DBMS, such as data security, authentication, and mainly transaction management enabling to control the lineage of the changes was reported to be critical.

The main assets may be grouped to the following characteristics:

- Data consistency – when an element is created, at the same time its geometry and non-graphical data are inserted. There is no mismatch between the graphical and semantic content.
- Topographical purity – in a well designed database with topological support no disconnected network, clear spaces or double boundaries may occur.
- Coordinate systems, datum and projection support (dependent on the implementation).
- Spatial indexing and access methods.
- Easy consistency check based on checksums, indexes and topological rules available.
- Versioning support and other database management assets – data merging and splitting, updating.
- Easy interchange of data thanks to XML/GML – spatial data and non-spatial data can be transferred at the same time.
- Easier "intelligence" implementation – automated generalisation/scale support, detail improvement while performing spatial selects, such as zooming on field models.

- Querying and management tools are transparent and well defined, as it is in a the non-spatial database world – SQL can be an example.
- Metadata storage for spatial objects.

The study proved, that even a simple spatial database, combined with widely accessible programming and developing tools are powerful enough to produce a useful LBS, that can be used for useful applications. LBS are not an experiment anymore, and the developers needn't to be afraid of the development process, with the appropriate tools widely available.

7.1 Discussion and Future Research

One of the biggest advantages of storing the spatial data in database tables is the possibility to implement scale-free application with dynamic detail modification, or generalization features. The spatial parameters of each modelled feature are given in real-world coordinate systems. There is no difference in complexity of the details of features, "mapped" for larger scale or smaller scale, it is just the cartographic visualization that is different. This can be very important for LBS, as far as standard datasets used for road map generation or city centres mapping could be integrated and their simplified and generalised output (view), could be rendered on the screens of embedded devices.

Future Research will focus on the possibilities to apply datamining strategies in integrated spatio-temporel databases for navigational and routing purposes, based on ontological interpretation of spatial knowledge into reasonable navigation instructions. More effort will have to be put on the investigation how to handle uncertainty in location information (arising from the kinetic nature of the phenomena and its temporal properties), with regard to spatial queries searching databases with navigational purposes.

LBS are just a part of the family of web-mapping application areas. It uses the asset of web-services approach to use distributed data from different providers and allows to embed them to one service/application. With LBS application design based on modular architecture and open standards, those services could join the growing family of applications profiting of the national and international Spatial Data Infrastructure (SDI) projects and clearinghouses. Those are only possible thanks to integrated spatial databases enables fair pricing, data content definition in metadata archives, standardized access through interoperable interfaces.

This can result in a big launch and much wider spread of the Geo community and the applications of geographic information, based on transparent commercial relations, for the benefit of the whole society.

References

Basch, J., Guibas, L.J., et al. (1997): “Data Structures for Mobile Data”, Stanford University, Stanford.

Beinat, E. (2001): ”Location-Based Services. Market and Business Drivers.” in Geoinformatics, April 2001.

Benett, P. (2000): ‘The NMEA FAQ, version 6.3”, peterbb@interchange.ubc.ca.

Buehler, K. (2002): “OGC and LBS Overview”, GITA presentation.

Dolton, L.M. (2001): “Prospecting Spatial Database Offerings”, www.geoplace.com.

ETSI, (2001): “Digital Cellular Telecommunications System (Phase 2+); Location Services (LCS); Location Services Management (3GPP TS 52.071 version 4.0.0 Release 4)”, ETSI, www.etsi.org.

ETSI, (2002): “Digital cellular telecommunications system (Phase 2+); Location Services (LCS); Functional description; Technical Standard TS 101 724, Version 8.5.0”, ETSI, www.etsi.org.

Fonseca, F., Davis, C. (1999): “Using the Internet to Access Geographic Information: An OpenGis Prototype.” in: M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman (Ed.), Interoperating Geographic Information Systems. Kluwer Academic Publishers, Norwell, MA.

Fonseca, F.T., Egenhofer M.J., et al. (2000). “Ontologies and Knowledge Sharing in Urban GIS”, Computer, Environment and Urban Systems (in press), University of Maine, Orono.

Java Location Services, (2002): http://www.JLocationservices.com.

Lemmen, C. (2003): ”RDBMS Product Survey”, in GIM International, February 2003.

Liu, Z. (2002). “A Java-Based Wireless Framework for Location-Based Services Applications”, MSc Thesis, University of Calgary.

Location Interoperability Forum, (2001): “General Presentation”, www.locationforum.org.

Ludden, B. (2000): “Location Technology”, Vodaphone UK presentation, www.GSMworld.com.

Ludden, B. (2001): “Operators Perspective”, Vodaphone UK presentation, www.GSMworld.com.

Niedzwiadek, H. (2002): “OpenLS Architecture” presentation, Open GIS Consortium, GITA.

Nikolai, D. (2002). “Mobile Positioning Solutions for GSM”, Alcatel R&I, Stuttgart.

Object Management Group, (2003): ”MDA Guide Version 1.0”, http://www.omg.org.

Oosterom, P. van, Lemmen, C. (2001): ”Spatial Data Management on a Very Large Cadastral Database”, Computer, Environment and Urban Systems, Elesevier Science Ltd.

Oosterom, P. van, Quak, W. (2002): “GIS Algorithms and Datastructures.”, lectures, Delft University of Technology, Department of Geodesy, Delft.

Oosterom, P. van, Quak, W., et al (2000). “The Architecture of the Geo-Information Infrastructure”, Delft University of Technology, Department of Geodesy, Delft.

Oosterom, P. van, Verbree, E., et al (2001). “Storing and Manipulating Simple and Complex Features in Database Management Systems”, Delft University of Technology, Department of Geodesy, Delft, (2000), in 3rd AGILE conference on GIS, Helsinki, Finland.

Oosterom, P. van, Vijlbrief, T. (1996): ”The Spatial Location Code”, in 7th International Symposium on SDH, Delft.

Open GIS Consortium, Inc. (1999): ”The OpenGIS Abstract Specification. Topic 5: Features”, version 4, OpenGIS Abstract Specification, OGC.

Open GIS Consortium, Inc. (1999):”OpenGIS Simple Features Specification For SQL, Revision 1.1”, OpenGIS Specification, OGC.

Open GIS Consortium, Inc. (2001): ”The OpenGIS Abstract Specification. Topic 1: Feature Geometry (ISO 19107 Spatial Schema), version 5, OpenGIS Abstract Specification.

OpenGIS Consortium, Inc. (2000): “OpenGIS Web Map Server Interface Implementation Specification”, Technical Report reversion 1.0.0.

OpenGIS Consortium, Inc. (2001):”Geography Markup Language (GML) 2.0.” OpenGIS Implementation Specification.

OpenGIS Consortium, Inc. (2003): “OpenGIS Location Services: Core Services”, OGC Implementation Specification, version 0.5.0.

Oracle Corporation (2001):” Oracle Spatial User’s Guide and Reference”, Release 9.0.1, Oracle Corporation.

Pfoser, D. (1995): “Issues in the Management of Moving Point Objects”, presentation, Nykredit Center for Database Research, Department of Computer Science, Aalborg University, Aalborg.

Pokorny, J. (1999): “Objektově-relační databáze”, in “Objekty ‘99”, Praha.

Retscher, G., Winter,S. (2001):”Concepting a Location-Based Service: Guiding visitors of the Technical University”, presentation, www.gi13.geoinfo.tuwien.ac.at/users/winter/ws01/.

Saltenis, S. (1998): “Indexing the Present and Future Positions of Moving Objects”, presentation, Nykredit Center for Database Research, Department of Computer Science, Aalborg University, Aalborg.

Scholl, M., Voisard, A. (1992):”Object – Oriented Database Systems for Geographic Applications. An Experiment with O_2.”, Project Report, Paris.

Srivatsa, H. (2002): “Location-Based Security for Wireless Apps”, IBM developer network.

Stonebraker, M., Rowe, L. (1986): “The Design of POSTGRES”, University of California, Berkeley.

Syrjarinne, J. (2001) : “Wireless-Assisted GPS. Keeping Time with Mobiles”, http://www.geospatial-online.com.

Tijssen, T., Quak, W. et al (2001): ”Spatial DBMS Testing with Data from the Cadastre and TNO NITG”, Delft University of Technology, Department of Geodesy, Delft.

Tomko, M. (2001). “Analýza možností lokalizácie mobilného prístroja a jej využitie v komerčnej ponuke služieb operátora”, University Student Competition “Mobilná komunikácia a kvalita ľudského života, Cena Eurotel 2001”, Bratislava.

Winter, S., Pontikakis, E., Raubal, M. (2001): “LBS for Real-Time Navigation. A Scenario.” In Geoinformatics, May 2001.

Wohlert, R. et al (2000): ”Location Based Services. Service Requirements Document, Revision 1.0.0”, GSM North America, The North American Group of the GSM MoU Association.

Worboys, M.F. (1995): “GIS – A Computing Perspective”, Taylor & Francis, London, UK.

http://www.epsg.org

http://www.gpsinformation.net

http://www.GSMworld.com

http://www.mobil.cz

http://www.w3.org/

http://www4.coastalnet.com/nmea

Abstract

Spatial Databases for Mobile GIS Applications

The need of flexible and real time access of spatial data in the modern society, mainly in business applications, implies a development of Mobile GIS solutions. The ability to provide data about location on site magnifies their value as the information finds its client in the right moment and on the right place.

Approaches enabling IS designers to develop mobile GIS applications without using proprietary GIS software packages, profiting of newly established communication, data encoding and transfer standards, exists. Such tools and approaches are further discussed.

The development of such an application requires good and well-modeled dataset with a suitable spatial functionality, a powerful and reliable positioning system, fast and cheap data transmission and sufficiently powerful portable computer. Data security, application extensibility and the use of heterogeneous browsing devices have to be taken into account while developing such and application.

While modeling a mobile GIS application, several basic problem areas can be identified, each requiring special attention:

- data storage and retrieval, as far as it still is not common to store geographical data in integrated spatial databases, that could reconstruct the topology of the searched objects from simple inputs,
- location determination technology part, that tackles also problems with precision, reliability, availability and coverage,
- Remote data access and querying, the communication technology between different devices in the network, data encryption and communication protocols.

When designing a modular client-server system with open architecture, the use of an integrated spatial DBMS is crucial. Such an approach often enables to “skip” the use of middleware GIS software, performing spatial analysis.

The author presents the modes ofl architecture for mobile GIS applications, exploiting the capabilities of an integrated spatial database for spatial data storage, management and analysis.

An example of designing the architecture of a simple system, enabling remote access to Cadastral data in a mobile environment, queried only by providing information about the position of the user is given. The thesis emphasizes the importance of the use of portable, object-oriented programming and modular approach. The example of the use of the Java programming language in an Internet environment is used.

Attention is given to the data flow and the reduction of the traffic online, that results in a reduction of users' costs, by the use of a Thin-client approach. It enables also to move the processing load to the server side of the system.

Keywords: Mobile GIS, integrated spatial database, system architecture, modules, interoperability

Abstrakt

Využitie priestorových báz údajov v mobilných aplikáciách

Moderná spoločnosť potrebuje čoraz častejšie flexibilne prístupovať k priestorovým dátam, v tzv. “reálnom čase”. Táto potreba, najmä v podnikovom sektore, si vyžaduje nasadenie mobilných GIS riešení. Schopnosť poskytnúť dáta o zvolenej lokalite priamo v teréne zvyšuje ich hodnotu, keďže client dostáva informáciu v správnom čase na správnom mieste.

Dnes už existujú prístupy, ktoré umožňujú dizajnérom informačných systémov vytvoriť mobilnú GIS aplikáciu s využitím najnovších štandardov pre komunikáciu, kódovanie a prenos dát bez použitia komerčných GIS balíkov. Tieto možnosti a nevyhnutné nástroje sú v príspevku hlbšie analyzované.

Vývoj takejto aplikácie vyžaduje kvalitnú a dobre modelovanú bázu údajov, s vhodnou funkčnosťou umožňujúcou priesotorové analýzy, výkonný a spoľahlivý systém na určenie polohy, rýchlu a lacnú infraštruktúru na prenos dát a dostatočne výkonný prenosný počítač. Bezpečnosť dát, možnosť rozšírenia a nasadenie rozličných koncových prístrojov musia tiež byť vzaté do úvahy pri návrhu podobnej aplikácie.

Pri modelovaní mobilnej GIS aplikácie musíme venovať pozornosť niekoľkým základným problémom:

- uloženie a správa dát, keďže používanie integrovaných priestorových databáz, schopných uchovávať geografické dáta a rekonštruovať topológiu ešte stále nie je bežné,
- problém určenia polohy, ktorý zahŕňa oblasti presnosti, spoľahlivosti, dostupnosti a pokrytia,
- vzdialený prístup k dátam a ich dopytovanie, komuikačná technológia medzi prístrojmi a v rámci siete, kódovanie dát a komuikačné protokoly.

Využitie integrovaných priestrových SRBÚ je kľúčové pri návrhu modulárneho systému s otvorenou architektúrou typu klient server. Jeho nasadenie umožňuje pri niektorých druhoch aplikácií úplne obísť nasadenie analytického GIS middlewaru.

Autor predstavuje modely architektúr mobilných GIS aplikácií, využívajúcich integrované priestorové databázy na ukladanie a správcu dát.

Prístup je demonštrovaný na príklade návrhu architektúry jednoduchého systému, umožňujúceho prístup ku katastrálnym dátam, dotazovaným len na základe parametru o polohe, získaného automaticky. Zdôrazňuje dôležitosť nasadenia prenositeľného, objektovo orientovaného programovania a modularného prístupu. Použitý je priklad nasadenia programovania v Jazyku Java v prostredí Internetu.

Pozornosť rovnako venujem redukcii „mobilného“ toku dát pomocou tzv. “Thin client” prístupu, ktorý umožňuje znížiť náklady používateľov, ako aj presmerovať veľkú časť využitia výpočtového výkonu systému na stranu servera.

Kľúčové slová: Mobilný GIS, integrovaná priestorová databáza, systémová architektúra, moduly, interoperabilita