Technische Universität München
Lehrstuhl für Kommunikationsnetze

A Distributed Community-Based Location Service Architecture

Maximilian Zündt

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik
der Technischen Universität München
zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr. sc. nat. Christoph-Georg Günther
Prüfer der Dissertation:	1. Univ.-Prof. Dr.-Ing. Jörg Eberspächer
	2. Univ.-Prof. Bernd Brügge, Ph.D.

Die Dissertation wurde am 24.05.2007 bei der Technischen Universität München
eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am
21.11.2007 angenommen.

Foreword

This work has been developed while being a research assistant at the Institute of Communication Networks (LKN) and Center for Digital Technology and Management (CDTM) at the Technische Universität München. During my time at these institutes, I was not only able to gain deep understanding of the field of location technologies and services but also a broad interdisciplinary knowledge by conducting various research projects, carrying out courses and performing organizational tasks at the CDTM. Without the support of many individuals this work would have never been completed.

It has been a great pleasure to work with all my former CDTM colleagues and staff members of the LKN. Being part of the first generation of CDTM assistants, I learnt that a small group of highly motivated people can achieve great things when working together, even with little resources available. Many special thanks go out to Peter Dornbusch, Martin Huber, Jonathan Landgrebe and Matthias Möller.

The uncomplaining support of Mrs. Uta Weber, her skills to smoothly handle all challenges and specialties of CDTM administration were always an invaluable resource. She contributed a lot to my work by providing me creative and friendly support in many times.

Many thanks go to my doctorial studies supervisor Prof. Dr. Jörg Eberspächer who always engaged me in fruitful discussions and guided me successfully towards my research goals.

I would also like to express many thanks to Prof. Bernd Brügge of the Chair for Applied Software Engineering for being my second examiner.

Apart from all the regular staff members, many students have provided extremely valuable work to the success of this thesis. Among them are Peter Ippy, Hermann Anton Karl, Bernd Laqua and Manuel Puerta.

My family and friends have always backed me up throughout the ups and downs of research. I thank you all.

Last but not least, I would like to thank my parents Heidelinde and Siegfried Zündt for supporting me heartily at all times with their wisdom and generosity. To them I owe everything.

Munich, March 2007 *Maximilian Zündt*

Abstract

Location-based services (LBS) to date offer information and specific services tailored to the current geographic usage environment. The "location service" delivering the required location information in turn is provided by a location platform using specific positioning technologies and geodetic data (e.g. maps). Currently no single positioning technology or location platform can provide a location service as well as geodetic data for all known LBS usage environments and required location QoS (i.e. accuracy, delay, etc.). Hence scientific research agrees that a truly ubiquitous location service can only be obtained by use of heterogeneous positioning technologies in our daily living environment, that is, by use of dedicated systems such as GPS and technologies capable of also being used for positioning e.g. the abundance of heterogeneous networks all around us. The difficulty which remains is how to effectively "capture" this diversified technology data, generating meaningful location information and providing a homogeneous ubiquitous location service.

In this thesis we present a novel distributed location service architecture which focuses on the collaborative generation and centralized provisioning of location information derived from heterogeneous networks and readily available positioning systems using distributed heterogeneous location platforms. The architecture is specified using an abstract system model. It contains functional components structured by three major core processes representing its initialization, positioning, and self-optimization properties. Client-based sensor mechanisms paired with centralized knowledge databases and community user models have been combined to provide a self-optimizing and domain independent location service framework.

Within the service architecture a flexible Location Enabling Server (LES) framework has been developed, representing the initialization and sustainability processes of a LES for various dynamic usage environments. The initialization of a community-based LES with focus on collaborative, community-based network environment data acquisition for positioning purposes has been analyzed and successfully simulated for an outdoor inner-city usage scenario. On the data processing side, suitable initialization and positioning models have been evaluated and empirically tested using captured WLAN hotspot information, showing the localization performance in relation to GPS within inner-cities.

An important requirement for providing a ubiquitous location service is the discovery of distributed location platform entities with different geographic scopes of responsibility. Using the heterogeneous network and/or GPS sensor information discovered by the mobile client, we have developed a network independent, DHT-based peer-to-peer location platform lookup mechanism using XPath in a hierarchical Chord ring structure. A demonstrator has been implemented showing the effectiveness of this approach.

Finally, a working location service architecture prototype has been implemented. As a proof of concept, a seamless indoor/outdoor navigation application has been implemented using two fully functional location platform entities.

Kurzfassung

Ortsbezogene Dienste stellen Informationen und Dienste passend zu der jeweiligen Lokalität eines Benutzers oder Gegenstandes zur Verfügung. Der Dienst, welcher die dazu nötigen Ortsinformationen bereitstellt, verwendet geeignete Lokalisierungssysteme und Geo-Informations-Datenbanken über eine „Location Plattform“. Jedoch sind die gegenwärtig verfügbaren Lokalisierungslösungen nicht in der Lage, allgegenwärtig und für jeden Lebensraum brauchbare (d.h. Innen- wie Außenbereich) und dem ortsbezogenen Dienst anforderungsgerechte Ortsinformationen zu liefern. Durch die effiziente Nutzung von verteilten, heterogenen Lokalisierungslösungen und Bereitstellung von Ortsinformationen über standardisierte Schnittstellen ist das Ziel der Forschung die Entwicklung von adaptiven, allgegenwärtigen ortsbezogenen Diensten in verschiedenen Umgebungen. Hierzu werden unterschiedlichste dedizierte und kommunikationsnetze–abhängige Lokalisierungslösungen verwendet. Die Schwierigkeit besteht darin, die dadurch vorhandene Heterogenität der Technologien und Lokalisierungsverfahren effizient und kostengünstig im Griff zu bekommen, um einheitlich Ortinformationen zu generieren und bereitzustellen.

Diese Arbeit beschreibt eine neuartige Dienstarchitektur für die Bereitstellung von verteilten Ortsinformationsdiensten. Der Schwerpunkt liegt auf kollaborativer Generierung und zentralisierter Bereitstellung von Ortsinformationen, welche aus heterogenen Netzen und mit herkömmlichen Lokalisierungsverfahren (z.B. GPS) gewonnen und verfügbar gemacht werden können. Die Beschreibung und Modellierung der Dienstarchitektur erfolgt aus verschiednen Sichtweisen. Sie umfasst die Struktur und das Verhalten von Steuerkomponenten für die Bereitstellung eines homogenen Ortsinformationsdienstes, sowie die dazu gehörenden Prozesse für die Initialisierung, Lokalisierung und Selbstoptimierung. Hierzu wurden endgeräte-basierte Sensormechanismen, zentralisierte Wissensdatenbanken und community-basierte Operations- und Verhaltensmodelle zu einem durchgängigen Servicekonzept vereinigt.

Die Bereitstellung des homogenen Ortsinformationsdienstes wurde durch eine getrennte Teilnehmer-, Ortsinformationsdienst- und ortsbezogene Dienststeuerung über die dafür verantwortliche Dienstbereitstellungseinheit realisiert. Für die Initialisierung, Lokalisierung und Selbstoptimierung eines verteilten Ortsinformationsdienstes wurde ein universelles „Location Enabling Server“ (LES). Rahmenwerk entwickelt, welches die zugehörigen Prozesse in den jeweiligen Lebensräumen über die zentrale Dienstbereitstellungsentität unterstützt. Das Rahmenwerk und die Prozesse wurden erfolgreich anhand einer community-basierten LES-Simulation und -Implementierung untersucht.

Eine weitere wichtige Anforderung in der Bereitstellung eines allgegenwärtigen Ortsinformationsdienstes unter der Verwendung von verteilten heterogenen Dienstquellen ist die Auffindung des geeigneten Dienstes in dem aktuellen Lebensraum. Hierzu wurde ein DHT-basierter Peer-to-Peer-Service-Discovery-Mechanismus entwickelt, der eine netzunabhängige Dienstauffindung anhand von endgeräte-generierten Sensorinformationen ermöglicht. Die Funktionalität des gewählten Ansatzes

wurde erfolgreich in einem Prototypen bewiesen.

Wesentliche Teile der Dienstarchitektur wurden prototypisch realisiert. Darauf aufbauend wurde zusätzlich ein Navigationsdienst prototypisch umgesetzt, welcher eine durchgängige Navigation zwischen Innen- und Außenbereichen unter der Verwendung vorhandener Lokalisierungssystemen ermöglicht.

Abbreviations

2dRMS	*Two-dimensional Root Mean Sqaure*
AAA	*Authentication, Authorisation and Accounting*
ACL	*Access Control List*
AGPS	*Assisted GPS*
AOA	*Angle of Arrival*
AP	*Access Point*
API	*Application Programming Interface*
BT	*Bluetooth*
CSCF	*Call Session Control Function*
CEP	*Circular Error Probability*
CDTM	*Center for Digital Technology and Management*
COO	*Cell of Origin*
CORBA	*Common Object Request Broker Architecture*
CoBrA	*Context Broker Architecture*
CORBA	*Common Object Request Broker Architecture*
CSD	*Circuit Switched Data*
DCOM	*Distributed Component Object Model*
DECT	*Digital Enhanced Cordless Telecommunications*
DGPS	*Differential-GPS*
DHCP	*Dynamic Host Configuration Protocol*
DNS	*Domain Name Service*
DHT	*Distributed Hash Table*
EGPS	*Enhanced-GPS*
EMI	*Electro Magnetic Interference*
EOTD	*Enhanced Observed Time Difference*
ERN	*External Resource Network*
ESSID	*Enhanced Service Set Identifier*
EGNOS	*European Geostationary Navigation Overlay Service*
FM	*Frequency Modulated*
GMLC	*Gateway Mobile Location Centre*
GTB	*Geo Tool Box*
GIS	*Geographic-Information System*
GPRS	*General Packet Radio Service*
GNSS	*Global Navigation Satellite System*
GPS	*Global Positioning System*
GSM	*Global System for Telecommunications*
GLONASS	*Globalnaya Navigationnaya Sputnikovaya Sistema*
HLR	*Home Location Register*
HSS	*Home Subscriber Subsystem*
HTTP	*Hypertext Transfer Protocol*
IM	*Initialization Module*
IMS	*IP Multimedia Subsystem*
IP	*Internet Protocol (RFC 791)*
JNI	*Java Native Interface*

LPR	*Local Positioning Radar*
LACBA	*Location-Aware Community-Based Architecture*
LASAP	*Location-Aware Service and Application Platform*
LAM	*LACBA Access Manager*
LBS	*Location-Based Service*
LC	*LACBA Client*
LCD	*LACBA Client Database*
LCM	*LACBA Client Manager*
LCP	*LACBA Client Proxy*
LCS	*Location Core Server*
LDAP	*Lightweight Directory Access Protocol*
LDB	*Location Platform database*
LES	*Location Enabling Server*
LiF	*Location Interoperability Forum*
LIN	*Location Identification Number*
LMU	*Location Measurement Unit*
LKN	*Lehrstuhl für Kommunikationsnetze*
LLA	*Latitude, Longitude, Altitude*
LLD	*LES Location Service Database*
LLM	*LES Location Service Manager*
LLP	*LES Location Service Proxy*
LOS	*Line of sight*
LP	*Location Platform*
LPA	*Location Platform Agent*
LPP	*Location Platform Proxy*
LPM	*Location Platform Manager*
LRD	*Location Reference Database*
LS	*Location Service*
LSM	*Location Service Manager*
LSP	*Location Service Proxy*
LT	*Location Trader*
MAC	*Media Access Control*
MEA	*Means Ends Analysis*
MLP	*Mobile Location Protocol, by OMA*
MM	*Module Manager*
MPP	*Mobile Positioning Protocol, by Ericcson*
MSC	*Mobile Switching Center*
MSC	*Mobile Switching Center*
MR	*Module Repository*
MTSAT	*Multifunctional Transport Satellite System*
NMEA	*National Marine Electronics Association*
OMA	*Open Mobile Alliance*
OOD	*Object-Oriented Design*
OSDE	*Open Services Delivery Environment*
OSGi	*Open Services Gateway Initiative*
P2P	*Peer to Peer*
PAU	*Profile Authentication Unit*
PDA	*Personal Digital Assistant*
PDB	*Policy Database*
PM	*Positioning Module*
PPS	*Precise Positioning Service*

P-CSCF	*Proxy CSCF*
QoS	*Quality of Service*
RDS	*Radio Data System*
RFID	*Radio Frequency Identification*
RNC	*Radio Network Controller*
RSS	*Received Signal Strength*
RPA	*Redundant Positioning Architecture*
SA	*Selective Availability*
SB	*Shared Buffer*
SDB	*Service database*
SMLC	*Serving Mobile Location Centre*
SIP	*Session Initiation Protocol*
SLP	*Service Location Protocol*
SOA	*Service-oriented Architecture*
S-CSCF	*Serving CSCF*
SGSN	*Serving GPRS Support Node*
SRNC	*Serving RNC*
SOAP	*Simple Object Access Protocol*
SSID	*Service Set Identifier*
SPS	*Standard Positioning Service*
TDOA	*Time Difference of Arrival*
TMC	*Traffic Message Channel*
TOA	*Time of Arrival*
TUM	*Technische Universität München*
UDDI	*Universal Description, Discovery and Integration*
UMTS	*Universal Mobile Telecommunications System*
UTRAN	*UMTS Terrestrial Radio Access Network*
UDB	*User database*
UP	*User Proxy*
UTDOA	*Upload TDOA*
UTM	*Universal Transverse Mercator*
UWB	*Ultra Wideband*
VoIP	*Voice over IP*
VPS	*Virtual Positioning System*
WGS	*World Geodetic System*
WAAS	*Wide Area Augmentation System*
WGS84	*World Geodetic System 1984*
WLAN	*Wireless Local Area Network*
WSDL	*Web Services Description Language*
XML	*Extensible Markup Language*

Contents

Chapter 1 Introduction

"Navigation is the art of choosing between futures"
The Plan to Singularity, by Eliezer S. Yudkowsky

1.1. Motivation

There are many different LBS solutions on them market today. Navigation and routing services based on on-board in-car and mobile (off-board) PDA solutions using standalone GPS devices and locally stored static map data, sometimes using also dynamic information such as car-traffic information via the FM radio broadcast channel dominate the consumer market. Cellular network operator LBS are slowly starting to gain potential as better end-devices with higher usability and more cost effective and accurate positioning technologies become available, but still suffer regulatory and privacy issues for an efficient global deployment. Indoor-tracking solutions using specialized radar equipment or well established short-range network WLAN or Bluetooth radio signals for location-based services are either too costly, complex or simply lacking the concurrent added value and/or market demand for such in-house services. Last but not least, internet-based providers such as Google and community platforms such as Flickr[1] – a service for publishing pictures on the internet including their geographic locations of where they have been taken – are considering teaming up with cellular operators pushing their services onto the mobile market. Looking at this spectrum of LBS market solutions, there is a clear trend towards a convergence of stand-alone, cellular operator, in-house and internet-based LBS providing network, domain independent ubiquitous LBS solutions.

Despite the diverse nature of LBS, underlying location platforms (LP) with respective positioning technologies have similar approaches on providing the best possible location information on entities such as people or other target objects such as buildings, parcels etc. Every positioning system consist of a Location Enabling Server (LES) and Geographic Information System (GIS) entity comprising the LP, in order to make use of and provide the location information from its designated usage environment. In the case of satellite-based positioning systems, GPS and the future Galileo system (will) provide good global coverage of accurate outdoor location information, yet certain weather conditions, urban canyons, inner-cities or indoor building conditions will still set limitations on accuracy and availability [ETZ05]. The abundance and almost ubiquitous

[1] Flickr: http://www.flickr.com

coverage of heterogeneous wireless networks in our daily living environment (e.g. GSM/UMTS, WLAN, or Bluetooth networks especially for indoors) and positioning systems realized using them [OM04b][DH03][DZ02] will hence be a key component for future ubiquitous location information access. Many solutions exist, but currently no LES and corresponding GIS is capable of covering and providing location information for all usage environments. Thus multiple inter-working LES and GIS entities with respective heterogeneous positioning technologies will be needed for ubiquitous location information and content access.

The "multi-facetted" wireless provider and technology landscape, as well as lack of standardization for inter-working and "easy access" to this heterogeneous technology information necessary for location determination systems, makes it difficult to effectively "capture" the diversity and distributed nature of positioning systems for ubiquitous, homogeneous location-information provisioning for LBS. Cellular operator-based LPs specify interfaces for accessing location information of clients within their networks for 3rd party application providers [OM04a][OM04b] having considerable privacy, performance and pricing implications [ZD05b]. Furthermore, the LBS inter-working and roaming between cellular operators is still an ongoing debate. The situation is even more extreme considering the numerous WLAN operators and providers i.e. for WLAN-Hotspot based positioning [ZTB03][LaM05]. There is a continuous evolution especially in the case of short range wireless net-works (e.g. WLAN, Bluetooth), hence making it even more difficult to provide a reliable and up-to-date positioning system based on such technologies. Short-range indoor based solutions as mentioned earlier and even commercial products such that of Ekahau [Eka06] often have proprietary provisioning means of location information and location content e.g. maps of local buildings. There is yet little movement but a lot of work to be done towards harmonized standards for homogenous location information and content provisioning from heterogeneous positioning systems and data formats i.e. relating to format transformation/translation issues.

As a result from the heterogeneity of communication networks and due to the slow evolvement towards 4G architectures for seamless heterogeneous network inter-working, a paradigm shift has evolved towards autonomous, multimode mobile clients being able to discover and roam between potential access networks by themselves [ZTB03][Zue04]. Extending on this principle, such "mobile sensors" can even collect certain wireless data for maintenance and site-mapping purposes, or even distribute local information such as accidents, weather warnings and other public/private event data to potential user communities via centralized knowledge databases [Zha05]. Based on such principles, community-, operator independent- or often called "open-oasis"-based approaches to next-generation network design and service provisioning platforms provide a potential competitive alternative solution [DFH04][ZTB03].

Last but not least, there is a growing notion that valuable as well as up-to-date location information and content can only be provided effectively by local sources e.g. communities living in the specific usage environment of the LBS. This is also true for capturing the dynamic nature and changes in our daily living environment (e.g. WLAN hotspot evolution in inner-cities, local events/traffic information etc.). The whole area of GIS data capture and retrieval tries to deal with this concept, yet is far too slow to discover such volatile location events. Internet communities and their innovative collaboration and communication means (e.g. blogging, geocaching, war-driving, etc.)

evolving into mobile communities are seen as a potential valuable resource in the provisioning of location content for future LBS.

In developing future ubiquitous LBS, all these issues need to be considered. One of the most important factors is to handle the heterogeneity of *dedicated* positioning technologies and respective location information derived from *location-inferring* systems using communication networks. Apart from this, community models and collaborative information generation and provisioning concepts will play an important role in not only providing value added location information and content but also help in capturing and controlling the heterogeneity of existing and future systems at hand. A novel location service relying on a distributed network of LPs will have to be independent of technology, usage domain and service environment. In this thesis a new solution to such a ubiquitous location service architecture will be presented.

1.2. Goal, Approach and Architecture Features

The goal of this work is to develop a location service which brings together heterogeneous dedicated and location-inferring positioning systems relying on the abundance of wireless communication networks in our daily living environment. The challenge here is to capture and control this dynamic and heterogeneous technology environment to provide a competitive *domain* and *network independent* location service. As a solution a novel location service architecture will be developed, which focuses on the *capturing* (initialization), location information *generating* and *homogeneous provisioning* (positioning) from heterogeneous network information. The dynamic and evolutionary aspect of wireless networks in the environment requires *self-learning* and *optimization* of the provided location information. Client-based data acquisition and provisioning via centralized knowledge databases concepts will be used for a network and operator independent location service approach. Furthermore, community models will not only help developing the *collaborative data acquisition* and location information generation processes, but also define the degree of interaction and location content provisioning of community members for potential new LBS concepts. Finally, the architecture should be *positioning technology independent* supporting readily available and future positioning technologies.

Apart from the provisioning of location information form heterogeneous networks the architecture should support the *discovery* and *provisioning* of heterogeneous, independent LP systems for optimal LBS support in varying usage environments and respective LBS location QoS requirements.

In order to realize these requirements in a new location service architecture solution, the following approach has been taken (see Figure 1.1). In the analysis phase we have examined readily available positioning systems and technologies with their respective LP solutions in their common usage environment. By analyzing existing community models and examples relating to the collaborative value generation we will then generate an architecture model. To account for the complexity of LPs we can describe the system in distinct system processes. Furthermore, we will consider existing architecture concepts and apply them to our specific model.

From this architecture model we will then look at the collaborative location information generation from heterogeneous network technologies in various usage environments more closely. We will define the boundary conditions and critical factors e.g. critical mass of users for initialization, types of networks and coverage, availability, evolution rate, etc. Following this analysis, we then will consider the process of capturing, generating location information and performing localization on the generated location data for various usage environments. The goal from this analysis is to define our novel architecture model in more detail and determine the possible positioning performance relating to existing and readily available positioning systems.

Having discovered the possibilities of collaborative location information from usage environments, we will look at the location service provisioning aspects (i.e. distributed location services) and how to combine them for ubiquitous location service provisioning. This will involve looking at the signaling and inter-working between the different defined architecture components and how specific tasks can be allocates to different control areas in order to keep complexity low e.g. user access, location service discovery, handover and centralized location service provisioning for LBS.

We will prove the functions of various system architecture components by means of iterative prototype implementations. Rapid prototyping methods and part system behavior simulation concerning the collaborative location information generation and user community behavior should become useful here.

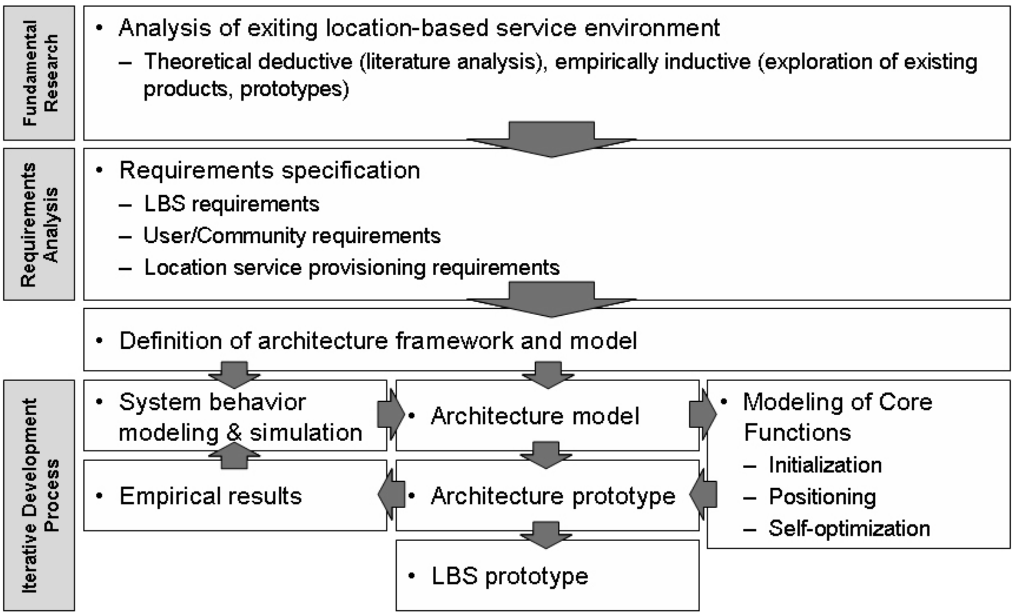


Figure 1.1 The formal approach for developing the new location service architecture concept

We will perform statistical analysis using readily available data gathering tools on capturing real life data from various usage environments. Together with the simulation results we will develop our prototypes further and test them in selected usage environments as well. The implementation of a LBS using out novel location service from our prototype architecture will be used as prove of our concept. With this approach we will develop the new location service architecture with the following features:

- **Seamless** provisioning of a **centralized** and **homogenous location service** using multiple, distributed location sources (i.e. LP entities)

- The centralized location service needs to be **globally** available, **scalable** and independent of access networks and service domains.
- Provisioning occurs via a **generic location service API** to be modeled after existing standards ensuring inter-operability with existing commercial LP solutions [OM04c][LES04].
- Differences in **location QoS** (LQoS) provided by respective LPs, and matching to LBS requirements is realized by means of **translation**, **adaptation** as well as potential **feedback** mechanisms to a LQoS-aware LBS.
- A **LQoS model** is specified on handling and matching the LBS requirements to available LP sources.
- Support of **heterogeneous positioning technologies** and respective LP implementations (i.e. of both readily available dedicated GPS-based and communication network location-inferring solutions). The framework should support current and future location technologies (i.e. be **positioning technology** and **LP independent**).
- The control of location services from LPs in respective usage domains will be via existing, standardized APIs. Hence the architecture can be realized without modifications to these existing location service domains.
- The architecture will be separated into three separately managed parts: user, LP and centralized location service control.
- The user control manages user access and profiles independent of the LP and centralized location service control logic. It considers security and privacy aspects of accessing user location information by LBS and other users.
- The centralized location service control registers and manages LBS requirements (e.g. session setup, LQoS negotiation and adaptation) depending on the location sources currently available on a user.
- The LP control handles the discovery and inter-working (e.g. handover) between LP entities. It determines the relevant entities for the users current usage environment considering user and LBS access rights.
- A **communication network** and **location service domain independent** location service discovery mechanism will be provided.
- The architecture will resemble three core processes: the **initialization** (i.e. sensor data capturing and location information generation), provisioning (i.e. **positioning** on generated data), and **self-optimization** (i.e. learning about changes in the environment and adapting the generated data).
- The initialization and self-optimization processes need to perform **integrity checking**, **authentication,** and **sanitation** on captured sensor data provided by community members for environment specific **reference location data generation** i.e. to account for hardware differences and user trust levels.
- A **flexible framework** will be provided enabling the realization of LP solutions for various usage environments and positioning technologies used (e.g. from fairly static indoor to highly dynamic outdoor community-based application scenarios). In each case initialization strategies, initialization and localization methods, as well as self-optimization features will be supported.

- The **client** middleware should be kept “light weight” due to legacy device support. It should be **modular** and platform independent to support many potential devices and future positioning systems e.g. by using a Java platform and object oriented programming.

The essential parts of this architecture will be implemented in a prototype. The seamless location service and inter-working between multiple LP entities will be demonstrated by an example LBS implementation. The data capturing, collaborative location information generation and positioning (i.e. respective positioning methods) will be validated by simulations and statistical analysis of real life environment data.

1.3. Related Research Areas

At this point we will introduce the research areas which are closely related to the architecture presented by this work. A detailed discussion on related work regarding heterogeneous positioning technologies, LPs and location service architecture concepts will be presented in chapters 2 and 3 later on.

Location Service Provisioning
The major research focus of this work deals with the provisioning of a location service from multiple distributed location sources. This requires a close look at the “location service delivery chain” i.e. positioning technologies and localization methods, as well as standardized and proprietary solutions on location service delivery [OM04b][OGISa] [Eri04]. This work uses readily available positioning technologies and location service solutions, focussing more on the initialization aspects (i.e. methods, efforts and costs involved in provisioning of localization). Initialization and positioning heuristics will be evaluated and applied relating to the community-based scenarios envisioned in this work.

Another main research area presented in this work lies in the provisioning of a ubiquitous location service using multiple distributed and heterogeneous LES and GIS sources along the lines of Nimbus [Rot05] and RPA [Pfe05]. Location fusion aspects of heterogeneous location sources as covered in [HIG03] are discussed and considered.

Other research deals with optimized transmission and availability of measured sensor data and location information according to the LBS needs e.g. speed of target, entering a certain area, etc. [KTL06][Leo03]. This work does not cover the signalling aspects and rather sees such functionality as future work on optimizing the concept proposed here.

The privacy and access control of user location information is very crucial and greatly discussed in this research field e.g. centralized or decentralized storage and access control [ZD05b][Leo03][Rot05], location information “obscuring” techniques [Rot05][SHG03], etc. Privacy and security on location service provisioning will be discussed and adapted in this work.

Context-Awareness and Frameworks
Context-awareness is a concept that enables electronic systems to adapt their actions or responses to a given situation defined by the context of the device and/or its user. Ontology languages [W3C04] are used to model and interpret different kinds of context information. Context-aware systems such as CoBrA [Che04] support the representation

of contextual knowledge, and provide knowledge sharing as well as context reasoning on different kinds of raw context information enabling context-aware applications. The capturing and modeling of heterogeneous context modeling is very complex and still an open debate, hence architectures more or less at a conceptual state. Hence, we will focus on dealing with heterogeneous location context information in our work.

Service-oriented Architectures (SoA)
The SoA concept has gained great momentum in the Enterprise IT-services environment. It provides a framework for registering, discovering and providing distributed services which can be loosely coupled and interchanged realizing complex business or product processes. The work presented here will adapt SoA principles and especially rely on service architecture modeling principles using RM-ODP [X.901].

Communities and Network Effects
Online communities and the Web 2.0 phenomenon have started to revolutionize the way use and communicate on the internet. There are many community types, interest topics e.g. dating, local events, etc. Technological advances allowed them to evolve into new living domains and become more context aware e.g. from internet or "virtual-communities" to "mobile communities" accessing information or services on-the-go, or location-aware mobile communities using explicitly location context in collaborating e.g. "geocaching". Despite the many forms each follow common laws regarding the community development, network effects, collaboration and user behavior or experiences the same user types found within a community. These principles will also be applied in developing our community principle.

Sensor Networks and Peer-to-Peer Systems
Sensor networks are self-organizing and usually composed of a large number of densely deployed spatially distributed, sensor nodes monitoring conditions at different locations and distributing their sensed information and services throughout the network [ASS02][LMS04]. Typical application scenarios are emergency situations (e.g. accidents) or health monitoring of patients (e.g. body area networks). There are location frameworks which provide their LBS without any central server entities using a peer-to-peer approach to provide their services [ZD05b]. Although we use distributed clients measuring the heterogeneous network environment and have them "share" the information with the entire community, we provide centralized entities and knowledge databases generating location information form the sensed data, as well as server side location service provisioning means. However peer-to-peer systems and network independent LES discovery mechanisms will be discussed.

1.4. Contribution of this Work

The presented work in this thesis and developed architecture manages to contribute to important fields in the area of location service development and provisioning:

- **Location service architectures:** This work provides a novel location service provisioning concept incorporating the location service creation, provisioning and self-optimization for respective usage environments. We have shown an alternative approach on providing a location service using heterogeneous sources in the user environment, solving the lack of standardization and inter-working between existing solutions by means of multimodal and autonomous clients and centralized knowledge databases. The community concept has been introduced showing its potential regarding access to locally relevant location information and provisioning for LBS. A working architecture prototype illustrating the major features, as well as a proof-of-concept LBS have been implemented to demonstrate the novel concept of this work.

- **Ubiquitous and adaptive location service provisioning:** This work has addressed the seamless location service provisioning using multiple distributed location services considering the LQoS requirements of certain LBS types. Furthermore, the new concept of LQoS-aware LBS has been introduced providing additional LQoS matching, negotiation and adaptation functions within the service architecture. This is seen as an important feature in future ubiquitous distributed location service provisioning for LBS.
- **Initialization and self-optimization of location services:** We have analyzed the initialization and provisioning of location information from various location service solutions. From this we have developed a flexible LES framework for the deployment and sustainability of positioning systems for various usage environments.
- **Location service discovery:** A new location service discovery mechanism based on a DHT-based peer-to-peer system approach using a hierarchical Chord ring LP structure has been adapted and developed from [ZET06]. A demonstrator application has been created implementing the location service discovery approach.

1.5. Structure of this Thesis

The remaining thesis structure is as follows (see also the illustration in Figure 1.2):

Chapter 2 will set the foundations of our work by describing the LBS environment, giving necessary definitions as well as an analysis of existing positioning technologies and LP approaches for certain usage environments. We will introduce the basic principles of communities and collaborative value generation, showing how they can be applied for location service provisioning. From this analysis, we will develop the requirements of our global location service architecture concept.

Chapter 3 will give an overview of related location service architectures and frameworks. They will be evaluated against our requirements as set out in chapter 2. In particular, we will elaborate on distributed location service architectures closely related to our approach.

In chapter 4 we will introduce our novel architecture concept, showing the various processing planes and control instances in delivering the location information, as well as the vertical distribution of the three core processes among different components regarding the initialization, localization and self-optimization. The architecture model will be presented using different abstract modeling views to illustrate the various functional requirements as specified in chapter 2. We will show the business view defining the different actors and interfaces for interaction with the centralized location service. The data model and component views will show the structuring of the various architecture components relating to the various interfaces. The session view will show the interactions of users, location platform providers and LBS with the architecture. Last

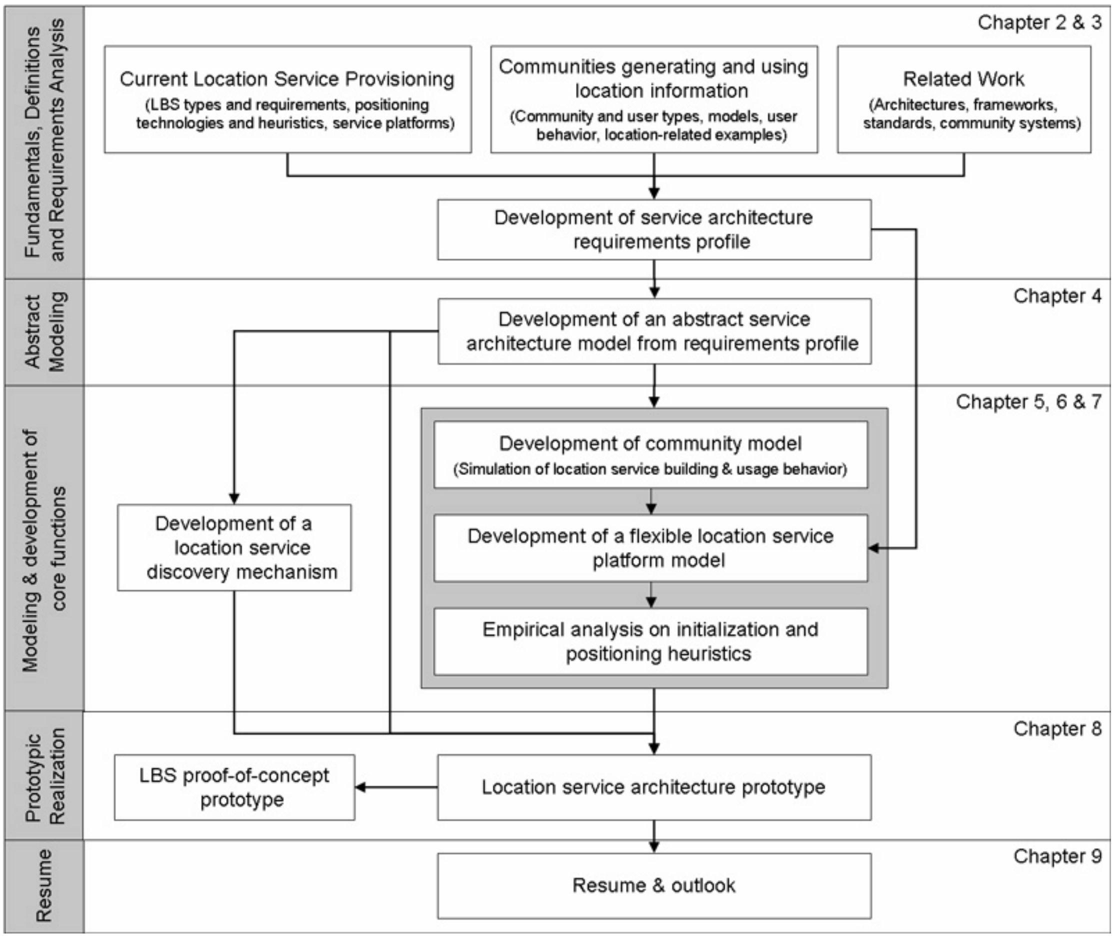


Figure 1.2 Structure of thesis and topics covered in respective chapters

but not least, the communication view will illustrate the signaling requirements between the different components. Example signaling scenarios, we show the functionality of the architecture and interplay between the various components.

Chapters 5 and 6 will focus on the initialization, localization and self-optimization core processes and respective architecture components. A community model representing the collaborative location service building and usage behavior will be developed in chapter 5. Technology, community, user and environmental factors will be discussed and rated. On choosing suitable community example scenarios, the most influential factors will be evaluated and simulated using the developed community model i.e. showing the user behavior and community development on varying critical factors. Using additional empirical results from test drives within the inner-city of Munich, we will determine the boundary conditions for successful location service deployment in our selected scenarios.

Chapter 6 will look at the initialization and positioning processes in more detail i.e. focus on the processing of collaboratively captured environment data, reference location data model generation, and localization performance on the generated data. First of all, a flexible LES framework fulfilling the requirements on initialization and location service provisioning for various usage environments and application scenarios will be presented. Using the challenging requirements of a community-based LES and the scenario used in chapter 5, we will evaluate respective initialization and localization methods for collaborative location information generation and positioning. Empirical results from test drives of the inner-city of Munich and a community-based LES prototype implementation will be used to illustrate a competitive localization performance in relation to the readily available technology GPS.

Based on the architecture concept and model developed in chapter 4, we will look at the location service discovery issues more closely in chapter 7. Various location service discovery mechanisms will be evaluated against the architecture requirements. A hierarchical LP model and a novel location service discovery approach based on a DHT-based peer-to-peer system hierarchical Chord ring structure will be presented, illustrating location service registration and discovery procedures.

Chapter 8 will present an overview of the specification and prototypic realization of our novel architecture concept, as well as a proof-of-concept seamless navigation application using multiple fully functioning LP entities.

Chapter 9 will close this work with a summary and outlook for potential future research items.

Chapter 2 Fundamentals and new Perspectives on Location Service Provisioning

The new location service approach described in this thesis bears reference to research areas dealing with location service provisioning concepts and architectures, as well as currently developing community-based approaches. This chapter will give an overview on existing state-of-the-art research and available solutions in respective fields.

The first part of this chapter will focus on the location service provisioning and respective positioning systems. Hereby, the fundamentals of location information generation and provisioning via positioning systems will be introduced. The following analysis will focus on currently available positioning system technologies and approaches, their generated location data characteristics as well as positioning system provisioning costs (i.e. initialization issues).The second part will deal with the actual location service provisioning via various LP approaches. On establishing the common principles of LPs, current approaches will be evaluated and features of a "silver-bullet" solution drafted. Having summarized the limitations of current location service solutions, we will introduce the concepts governing community-based systems and user generated co-value production, showing the opportunities which will set the foundations for an alternative approach on realizing a location service. Finally, the requirements for a global, "seamless" location service architecture will be outlined, forming a new paradigm on establishing and providing a location service independent of positioning and communication network technologies from our daily living environment.

2.1. Location-Awareness and Location-Based Services

Location information is a form of context information which can be classified as follows [DA99]:

Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

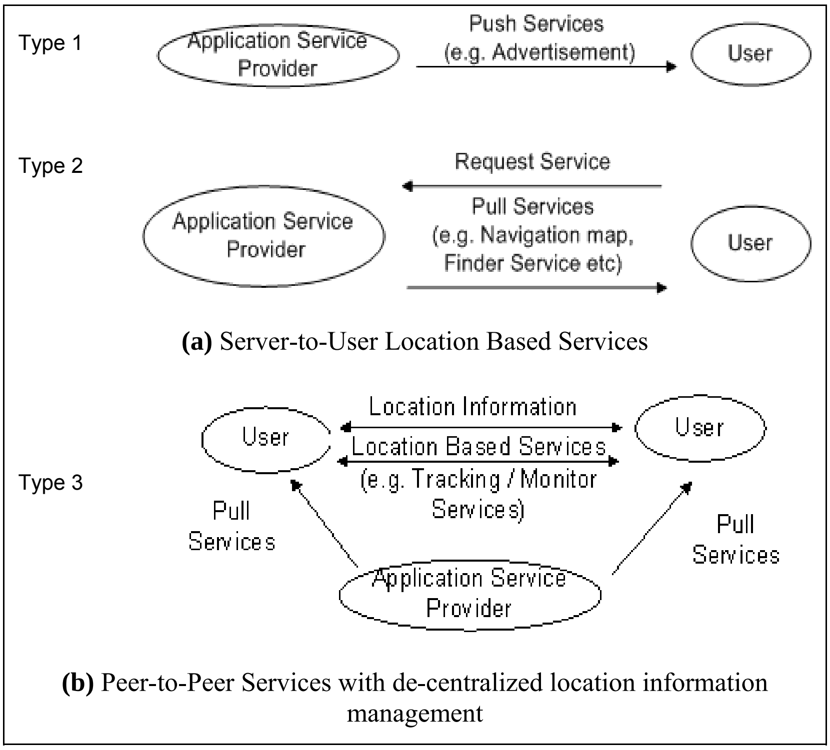
Hence context information can relate to the time, place, and state or presence status of an entity, influencing an application's execution. Relating to the place i.e. location context an application is refereed to as "location-aware" or "location-based".

Corresponding services are called ***location-based services*** (LBS). The term “service” is commonly used in network domains. According to the World Wide Web Consortium[2] (W3C) the definition of a web service is that:

Web services provide a standard means of interoperating between different software applications running on a variety of platforms and/or frameworks.

The key point in this definition is that web services enable heterogeneous application inter-working. The term ***location-aware application*** describes an application which works *offline* such as a car navigation system. Due to the common understanding of the term LBS and the trend towards ubiquitous service access we use the term LBS relating to both types of services and/or applications in this thesis.

The kinds of LBS services offered can be classified into two main types [3GP04] as illustrated in Figure 2.1a: Services which are triggered by an application service provider when a user arrives or enters a certain geographic location, e.g. push advertisements in a shopping mall or localized weather warnings, are classified as type 1 services (also called “push” or “triggered services”). Services which are typically information services requested by a user from an application service provider, e.g. finding nearby restaurant/gas-stations or navigation services, are classified as type 2 LBS (so called “user-requested” or “pull services”).



(a) Server-to-User Location Based Services

(b) Peer-to-Peer Services with de-centralized location information management

Figure 2.1 Location-based service classifications

The two types illustrated in Figure 2.1(a) are server-to-user LBS where location information is provided and managed by a ***central location platform*** provider, i.e. an approach followed by mobile operators primarily [OM04b]. Access to location information i.e. to 3rd party ***application providers***, is only being made available through mobile operator network entities. A third type of peer-to-peer LBS (Figure 2.1b)

[2] W3C: http://www.w3.org

represents a different approach to type 1 and 2 LBS, in which location information is directly exchanged by communicating peers, but LBS can be still provided by a 3rd party. Stand-alone solutions such as in-car- or mobile PDA-based navigation using GPS and static geo-information data stored locally on the device for outdoor navigation can be seen as a ***decentralized location platform*** approach. The issue of (de-)centralized location information provisioning will be discussed in later sections of this chapter in more detail.

The LBS classifications represented in Figure 2.1 give a short overview on recognized LBS types. A classification representing next generation LBS is presented in [KTL06]. Here, the classical LBS types are extended to include the different approaches and possibilities on providing a location service in different usage domains which will help define the requirements on future seamless location service architectures later on in this thesis. The extended LBS classifications are as follows:

1) ***Reactive*** and ***proactive*** LBSs: This corresponds to the type 2 interactive LBS (reactive) and type 1 triggered LBS on the occurrence of predefined events (proactive).
2) ***Self-referencing*** and ***cross-referencing*** LBSs: This distinguishes whether the user is processing his own location information (self-referencing) or that of another entity (cross-referencing). The latter case must consider privacy aspects of that entity, respectively.
3) ***Single-target*** and ***multi-target*** LBSs: The single target LBS considers location information in reference to an entity, whereas the multi-target LBS interrelates the location information of multiple entities, e.g. they are close-by, within the same building etc.
4) ***Central*** and ***peer-to-peer*** LBSs: This distinguishes between the centralized (Type 1 & 2) and decentralized (Type 3) approach on managing location information.
5) ***Outdoor*** and ***indoor*** LBSs: This refers to availability and relevance of the LBS, i.e. global (outdoor) availability and relevance, or only local relevance to a "closed" (indoor) environment.

Items 1) to 3) help to distinguish between LBS type requirements on location information and user interaction mechanisms. Apart from user privacy and availability considerations on LBS, items 4) and 5) relate much more to location service architecture requirements. While item 4) has implications on the general location service architecture strategy, item 5) addresses the "seamless" location service availability question to be addressed in our later architecture discussion. Table 2.1 summarizes the discussed LBS classifications with corresponding LBS examples. Instead of classifying LBS as outdoor or indoor, it is more meaningful to talk about global or local relevance in terms of availability. Item 4) has been omitted since it relates more to the infrastructure rather than to the LBS requirements, although peer-to-peer benefits user privacy control in some of the listed cases [ZD05b]. An accuracy indicator has been included in each case.

What becomes clear when looking at the various LBS types and requirements on location information delivery as well as accuracy, a single system capable of providing a location service meeting all requirements efficiently, at all times and at a reasonable cost is not very feasible. Since a majority of LBS involve location changes of both user and target(s), the complexity of such a system even increases if the location service

Category	Reactive vs. Proactive	Self. vs. Cross Reference	Single vs. multiple targets	Global vs. local relevance	Required Accuracy
LBS Information Services	Reactive	Self	Single	Local	<1km
Point of Interest	Reactive	Self	Single	Local	<1km
Friend Finder	Reactive	Cross	Both e.g. date finder	Global	<200m
Tracking Services	Reactive	Cross	Single	Global	<200m (outdoors) <5m, room accuracy (indoors)
Emergency, Assistance Services	Proactive	Cross	Single	Global	<50m [FCC06]
Announcement, Location-Based Advertisements	Proactive	Self	Single	Local	<1km
Location-Based Billing	Proactive (towards infrastructure)	Self	Single	Local	<500m, Cell Accuracy
Navigation	Reactive	Self	Single	Global	<30m (outdoor) <5m, room accuracy (indoor)
Fleet management	Reactive	Cross	Multiple	Global	50m

Table 2.1 LBS Classifications according to [KTL06]

needs to seamlessly adapt to changing environments, e.g. moving between indoors and outdoors. This most certainly results in a tradeoff between location information quality and availability. A further critical look at available positioning systems and respective location service platforms will help to understand this dilemma better.

2.2. Positioning Systems and Technologies

A positioning system is the heart of every LBS provisioning process. There are many positioning methods possible, and each positioning system has its advantages and disadvantages, but all have the main task of providing the location information of an entity in reference to some coordinate system. The goal of this section is to outline the fundamental principles of most commonly available as well as state-of-the-art positioning systems. The evaluation will briefly cover the positioning methods and technologies, but try to convey the respective provisioning cost as a whole.

2.2.1 Location Models and Location Data Characteristics

The location information determined and provided by a positioning system can be described more precisely in a ***location*** or ***spatial model*** [Rot05]. A ***location model*** structures and organizes the location space, describing the relation between target and reference locations to LBS. There are several forms in which a location model can be conveyed and interpreted. Current literature defines three possible ways [Rou02][Rot05]:

- ***Physical*** or ***geographic*** location models are the most common approach in providing location information in a meaningful form to LBS. They express location information in a common unique coordinate formats and metrics (two or three dimensions, or mapping from one or the other), e.g. WGS84 [WGS], NMEA [NME], Cartesian [DO01], Gauss-Krüger etc.

- ***Semantic*** location models give semantic meaning to location information in a form more meaningful to users and/or LBS. They are often in textual format and can be a powerful mean of representing location information.
- The last location model description possibility refers to the ***proximity relation*** of location information, i.e. the user is close to target or another user, often used in fleet management applications. Most literature does not classify this as a separate model but considers this as something in common to both semantic and physical locations, i.e. a relative compared to an absolute location model

A physical location model often refers to a single point in space whereas a semantic location model to a whole area. For example, instead of using the physical coordinates: latitude 48.152576 longitude 11.577410, one could convey more meaning to a user by the semantic description "Technische Universität München, Building 5, 2nd floor, Room 2509", or just "Max's office". There are many research fields dedicated to the context interpretation, semantic modeling and mapping between physical and semantic models [DHN04]. Although semantic modeling is a very powerful tool in interpreting location context, it is up till now a very difficult task trying to identify common places/objects or even fusing between different sources due to the many representation forms and languages available. Since this is not the scope of this work, hence we will restrict our view to geographic location models.

The location model representing the location information for a LBS provided by a positioning system specifically provides useful information about the location source, i.e. the LP constituting the positioning system (we will elaborate on this in section 2.3.1), its provided location information, and delivery aspects, e.g. location information update interval. For meeting the requirements of LBS, every positioning system fulfils the following location data characteristics in different ways:

- A ***coordinate system*:** the output of a positioning system i.e. location information value generated by the sensor, reference data and localization method, can be mapped to a respective globally unique or only locally relevant proprietary coordinate system [ZDS04]. Standardized coordinate systems usually describe a unique location in three-dimensional space, e.g. LLA, ECEF or WGS84 [Rot05].
- ***Accuracy***: This value gives a measure on how accurate the calculated location information value is with regards to the real physical location. This is usually determined by the accuracy of the reference measurement signal used during initialization (reference data generation of a positioning system), quality of resulting reference location, and localization method used in providing the final location information value estimate.
- ***Precision***: This gives an indication on how reliably or precisely the location information value was determined, i.e. out of how many localization events this value can be obtained with a certain error probability (or deviation of location measurements). There are two useful indicators for this parameter. The *Circular Error Probability* (CEP) and the *2-distance Root Mean Square Error* (2dRMS) [Har01]. The CEP defines the radius of a circle around the true location which contains 50% of the measured locations, whereas the 2dRMS defines a (usually smaller) circle, where 95% of the measured locations reside.
- The ***initiation time*** is the time taken for a first position fix or location information value being obtained. This value can vary during the lifetime of positioning source. For instance: upon cold start, GPS can take from a few

seconds to several minutes until it obtains a first location information value. But afterwards location information is almost instantaneously available. In our case, the term initiation time is extended incorporating the time taken from a positioning request until the location information is obtained. This may include transmission delays in the network if the position source is only remotely available. Since this value is in most of the cases in the order of a few milliseconds, it is miniscule compared to the required location update intervals.

- The ***location update interval*** is defined by the time between location information requests relevant to the type of LBS e.g. tracking service. The value is usually optimized to the travelling speed of the moving target. A faster moving target such as a car on the motorway may require a shorter location update interval than a tourist walking through a museum.
- ***Freshness***: This indicator tells the age of the location information value generated and potentially cached by a location service since the last location update. The value can be determined by the timestamp of the last location update and can be represented in seconds.
- The ***resolution*** factor is dependent on the type of localization method and respective location model used, i.e. point (high resolution), area (low resolution) or a probabilistic model, as we will discuss later on.
- The ***scope*** defines the area using a suitable coordinate system for which a location service can provide location information. These coordinates can be globally unique or only valid within a closed system i.e. indoor area, using a special building coordinate system. This parameter is closely related to the availability factor as explained next.
- ***Availability***: This factor in effect specifies whether a location service is available for positioning or not. Restrictions on LPs occur based on the environment (locally indoors or globally outdoors), timely availability (i.e. only at certain hours), or access rights.

The *coordinate system*, *initiation time*, *scope* and *availability* parameters are usually associated with the location service provided by a LP. Concerning the location information delivery aspects to LBS, the corresponding parameters are *location update interval* and *freshness*. The parameters actually determining the “quality” (i.e. location QoS) of the location information value generated are *accuracy*, *precision* as well as *resolution* which relate to the localization methods used. Depending to the latter, this has an effect on how the location model output is presented to the LBS.

The most logical representation of a physical location is a point in space given respective measurement error with regards to the “true” physical position, e.g. latitude 48.152576 longitude 11.577410 with an accuracy of 15m 2dRMS using GPS. This ***point model*** definition is the most desirable location model output, but it does not work for all localization methods: COO-based systems, for instance, define the service area of the serving GSM/UMTS cell as possible user locations, hence following an ***area model***. Other systems may use a ***probabilistic model*** giving a two-dimensional probability distribution for a measurement expressing the probability that a user could reside on a reference grid mapping to a point in space. The location model issues will become more relevant later on when discussing the location model for the generic, centralized location service.

2.2.2 Principles of Localization

Localization is a mechanism for discovering spatial relationships between entities, i.e. location context [Gir05]. In other words, the principle of localization is to be able to determine the position of a point or region in relation to some other point(s)/region within the same space. The location context of the entity or entities of interest (i.e. ***target(s)***) is determined by the positioning system. Hence the location context is either in relation to the positioning system or other located entities, whereby the role of the locating system can also be carried out by an entity (e.g. sensor networks). Entities can either be stationary or moving. ***Tracking*** is simply following the line, orbit, path, rough road or series of marks of an object by successively locating it, i.e. ***periodical localizations***.

According to [Rou02] there are four fundamental localization mechanisms which can be classified as follows: ***geometric***, ***statistical***, ***scene analysis*** and ***proximity-based*** localization. Virtually all state-of the-art positioning systems are based on and combinations of these mechanisms to provide best possible positioning accuracy or precision. A detailed explanation on the various mechanisms and their mathematical concepts can be found here [HB01]. A summary of the localization methods, their advantages and disadvantages, as well as example positioning technologies is given in Table A.1 in appendix A. A fundamental understanding of the presented localization principles will be relevant for further development and application of the concept represented by this work.

Geometric localization refers to distance (i.e. lateration) or angle measurements (i.e. angulation). Lateration uses the calculated distances from multiple known reference point locations. The distance can be obtained by *time-of-flight* measurements between sender and receiver on a direct line-of-sight (LOS) signal, as well as very precise time stamping and clock synchronizations at both ends for high localization accuracy [Zha02]. Reflected signals from obstacles (e.g. buildings, walls) can induce errors in the measurement, also known as the *multipath reflections*. An alternative to the time-based measurement is to use signal *attenuation* or the *received-signal strength* (RSS) at the target relative to its original intensity at the respective reference point i.e. sender. A distance measurement can then be obtained by using the so called path-loss model [DZ02]. The accuracy of the resulting distance value depends on the precision of the signal propagation model and *multipath fading* [DZ02] which can be caused by moving obstacles, changes in atmospheric conditions. Finally, a*ngulation* is a geometric technique were triangulation is performed between sending and receiving entities by the use of angular measurements. Expensive phased antenna arrays and adaptive antenna transceivers are needed. The accuracy is affected by *multipath reflection* problems and degrades with increasing distance to the target [HM04].

Localization by means of *scene analysis* uses a suitable description of an observed area and images of various viewpoints identifying features of a scene. The analysis of the scene determines the location of the respective viewpoint in the observed area e.g. visual/geometric scene analysis, signal-strength or temperature maps with appropriate reference points for associating particular locations with corresponding signature profiles. Signal-strength profiling (also known as “fingerprinting”) is the most common method used [DZ02][Con05]. However, the accuracy and precision of localization is greatly determined on the number of distinct reference signal-strength profiles/fingerprints available and multipath fading. The provisioning/initialization cost of such an approach is directly proportional to the size of the area to be considered and respective reasonably fingerprint density. Therefore such an approach is best suited for closed, fairly static and small to medium sized usage environments (e.g. office, warehouse etc.)

Probabilistic approaches to localization which relate to the problem of machine learning fall in the *statistical* category. Relating to the previous fingerprinting example, the closeness of a currently received signal strength fingerprint to the pre-recorded reference profiles is not given by a distinct value (e.g. Euclidian distance [DZ02]), but as a probability estimate instead e.g. by use of probability density functions (PDF) [Rou02]. The higher the reference point density, the higher the possible localization accuracy/precision will be. The bigger and more dynamic the observed area for localization, the less effective and computationally expensive this technique will be [LaM05]. Additional problems arise due to the fact that such probability-based localization mechanisms often assume a 2D normal probability distribution of reference locations (e.g. Kalman filter [Kal60], Ekahau [Eka06], [YAS03], [WNY05]), which is often untrue in some usage environment cases. Particle filters [PHU06] for instance, can have weighted reference location probabilities i.e. arbitrary probability distributions. More sophisticated probabilistic methods consider previous location probabilities of a target in order to determine the future locations (e.g. Hidden Markov Model, HMM [SKS03]).

Proximity-based localization is where the target is “close” to a known reference point (i.e. within a specified range) and the target’s location is associated with that reference point. Usually, this range refers to an area around the reference point where the target has been identified. The location accuracy is specified by this area.

In general, the geometric techniques tend to be more suitable in larger, outdoor environments compared to the scene analysis and statistical methods presented. The problem with the latter two is that the cost in both post-infrastructure initialization/deployment and computational overhead increases proportionally to the to-be served locating area. On the contrary scene analysis and statistical methods are less susceptible to erratic signal-strength or other sensor data fluctuations compared to the geometric methods, provided the reference location data integrity is still provided. Hence they are often highly customized to the environment at much higher initialization effort compared to geometric means which mainly require the reference locations of the infrastructure components (e.g. base stations) and certain end-device/base-station capabilities, i.e. hardware cost by means of special antenna arrays, timing resolution etc.

Concerning accuracy and precision both geometric, scene analysis and statistical methods provide competitive localization performance in their respective usage environments. The proximity-based solutions, be it tag-based for very close ranges or COO-based for various usage environments, provide the most cost effective and readily available approaches, on the downside, however being the worst concerning location accuracy and precision. The trade-off concerning accuracy, availability and precision has to be evaluated for respective LBS requirements.

A common approach used in positioning systems is to use multiple localization types at once to compensate for individual localization type limitations for more accurate localizations e.g. using geometric RSS-based method followed by additional fingerprinting scene analysis evaluation [ZDS04]. Another approach quite commonly found in research to-date is the use of ***sensor fusion***, i.e. using heterogeneous positioning systems whose sum of resulting localization accuracies is greater than each of the individual, in itself inaccurate sensor types [HIG03] e.g. using an temperature-based sensor in conjunction with a video camera. The challenging part is on "interpreting" the sensor information and combining the individual outputs in a meaningful form suitable for respective LBS. The statistical approach is the most obvious choice for localization and can be virtually used for all location sensor types. As already mentioned the probability distribution is not normal for every positioning system case and would have to be considered and communicated in every fusion case causing a lot of overhead, hence not very practical when dealing with heterogeneous positioning systems [Rot05]. It can be however feasible within a positioning systems using complementary sensors [Hig03].

2.2.3 Description and Classification of Positioning Systems

The fundamental principles of generating location information from the usage environment via appropriate positioning technology sensors are in essence common to all ***positioning systems,*** e.g. GPS, WLAN-based, GSM COO-based etc. Figure 2.2 exemplifies this in a respective ***positioning system model***. The positioning system model defines two phases of providing location information using sensors in the environment: the *initialization* and the *positioning* phase.

The ***initialization phase*** identifies reference locations in space (i.e. the usage environment) suitable for the localization method(s) to be used. This process involves a *feature extraction* of signals already present in the usage environment (e.g. communication networks signal strength, heat signatures or visual images of certain entities etc.) and *mapping* of these features to geo-locations defined by a respective coordinate system. This task is performed by ***initialization methods*** and the generated reference locations stored in a common positioning system database.

In geographic and proximity-based localization, the initialization phase determines the exact geo-coded locations of the base stations/tag readers in reference to some coordinate system. In scene analysis and statistical location means the exact locations of the base stations do not need to be known, but sufficient fingerprint location profiles and distributed grid reference locations for calculating position probabilities need to be respectively generated to ensure the required localization accuracy and precision. Thus, the ***positioning phase*** of a positioning system determines the target location of an entity using the current sensor data from the environment and reference locations stored in the positioning system database. In an "ideal world" the localization method should be able to determine the exact reference location. But there are three potential sources of error diminishing this fact: the highly dynamic multipath fading and reflection problems in the environment are the biggest error sources which essentially make it virtually impossible to determine the exact location references during positioning from the initialization phases. Secondly, an error is usually induced in the mapping process of environment features extracted to the coordinate system to be used, i.e. due to inaccuracies of the referencing source used, e.g. manual location referencing, by a user on a digital map [Eka06] or using a commercial GPS sensor with an inherent in-accuracy of several meters in urban canyons. Hence the quality of the possible location information accuracy and precision not only depends on the localization method itself, but more importantly on the quality of the generated reference locations and environment conditions itself. Therefore, the location model output always represents the quantification of the closeness of the generated location information with regards to "world truth" given the respective positioning system model.

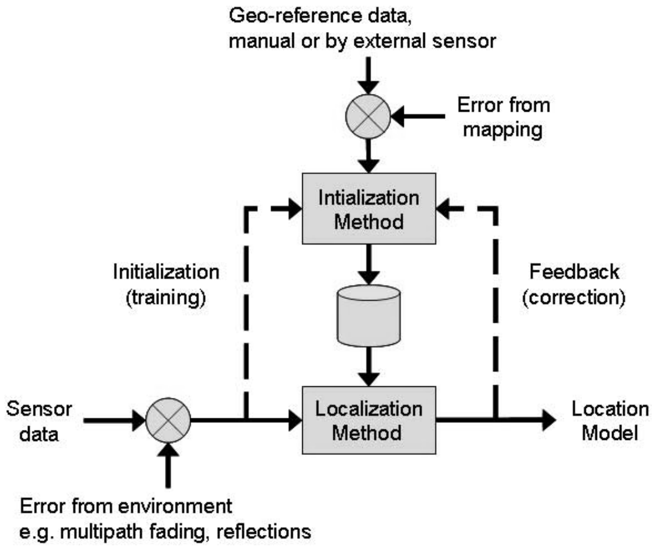


Figure 2.2 A model of a positioning system

The overall ***initialization cost*** can be defined as the sum of respective *infrastructure cost* (both hardware and software) and *training cost* on the respective usage environment. The latter refers to the reference location generation process. The weighting of these cost factors is greatly determined by the localization method and technology used in the positioning system. Thus, geometric and proximity-based localization methods have miniscule to none training costs compared to the potential infrastructure costs, whereas with statistical and scene analysis-based methods it is the opposite case, since positioning systems of these kinds usually rely on existing infrastructure components. To make matters worse, they need to be potentially adapted to changes in the environment (i.e. re-trained), as we will illustrate further on. Generally speaking for all localization types, the initialization cost grows in direct proportion to the to-be served area in the usage environment i.e. ***service area*** of a positioning system. Ideally, the service area covers the entire usage environment which in effect is rarely practically possible due to building walls, furniture or radio shadows/blind spots.

The ***feedback*** loop indicated in figure 2.2 represents a desired feature sought in positioning systems relating to "self-improvement" and "self-learning" capabilities. One of the main tasks is to reduce overall initialization and potential reoccurring initialization costs during the lifetime of a positioning system.

The Relationship between the Initialization and Positioning Phases

Once a positioning system has been initialized, that is, the infrastructure has been setup and/or reference locations have been sufficiently generated for the usage environment, the positioning system can be seen in a "ready state", i.e. the positioning system provides the localization quality intended. Over time the infrastructure in the usage environment may change (e.g. furniture moved, access points relocated or just renamed etc.), hence the positioning system may not be able to provide its initial localization quality due to more frequent reference location mismatches, and the average localization quality for that usage environment will drop. This effect is more drastic depending on the type of localization method used or general dynamics of the usage environment. Unless base stations or readers (for tag-based methods) are not relocated, geometric and proximity-based localization methods are not much affected. Statistical or scene-analysis-based methods, however, are considerably affected if not completely disabled in some parts of the usage environment. In order to ensure a continuous, "acceptable" localization quality level, the positioning system needs to be re-trained i.e. re-initialized accordingly at respective cost [ZDS04]. Figure 2.3 exemplifies this initialization and positioning phase relationship.

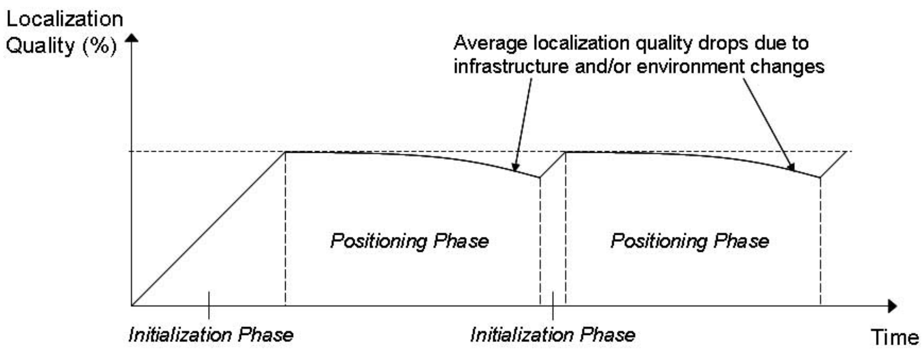


Figure 2.3 Simplified illustration on initialization and positioning phases of positioning system

An early detection of a drop in localization quality is desirable to ensure a certain location service quality for LBS. Ideally, the positioning system could detect changes in the environment and infrastructure on its own and react on it respectively, e.g. by requesting a re-initialization early, or correct its reference data itself accordingly. The former case can be achieved by so called "nearest neighbour" calculation or cluster-analysis tests using the reference data stored in the database and currently detected sensor data during localization in the positioning phase [BB03]. The latter case of "self-learning" is more complex and would require more sophisticated ***feedback*** methods and procedures commonly found in neural network applications. Self-learning represents the silver bullet approach where an initially trained positioning system is capable to detect environmental changes to a certain degree autonomously, i.e. the system is constantly trained while positioning occurs.

Classification of Positioning Systems

Having described the aspects and desired features of positioning systems, we can summarize them into two main types for our following analysis: ***dedicated*** and ***location inferring*** positioning systems. *Dedicated positioning systems* and technology have the sole purpose of providing localization. *Location inferring positioning systems* is where the location can be inferred as a byproduct of a system usually serving another major purpose. These can be subdivided into two main groups: *communication-network dependent* and *identity-based* positioning systems. In the former case, the location information is added value, a byproduct of current wireless or wire line communication systems. An *identity-based system* usually solves the purpose of authenticating/identifying users or objects e.g. using RFID tags swipe-cards. Since the location of the respective reader is usually known, a location can be inferred on an identification trigger event respectively.

Although dedicated and location inferring positioning systems are two different approaches in providing location information, respective solutions can be put into the following categories for defining the roles between infrastructure and mobile parts accordingly [Nok01]:

- ***Handset-based*** positioning: The mobile device performs the location calculation to determine the user's current position alone, e.g. GPS.
- ***Network-*** or ***infrastructure-based*** positioning: The network or infrastructure component calculates the users' location alone, e.g. GSM Cell ID.
- ***Handset-assisted*** positioning: The mobile device performs measurements which the network or infrastructure component uses for location determination, e.g. GSM TDOA and EOTD, as well as OTDOA in UMTS/FDD Systems.
- ***Network- or infrastructure-assisted*** positioning is where the network or infrastructure component provides information to the mobile device which performs the localization, e.g. AGPS

The following sections will use these classifications for analyzing existing and readily available positioning systems and technologies accordingly. The analysis will focus on the initialization cost, localization accuracies as well as service area issues including the location information parameter and method discussion points of sections 2.2.1 and 2.2.2. A summary of the presented positioning technologies, rough localization performance values, as well as advantages and disadvantages, etc. is given in Table A.2 in appendix A. Detailed explanations on each technology presented can be found here [Kue05].

2.2.4 Dedicated Positioning Systems

The most commonly known technologies in this category are satellite, radar and laser-based positioning systems. Satellite-based systems are widely available and used in various consumers, industry and military applications providing global outdoor localization capability. Radar-based systems originated from military and the civil aviation industry but have found their way into short range, high-end consumer-based and industrial products e.g. car distance cruise control and steering of robots etc. Laser-based systems have yet to make this transition into the consumer domain and are not very practicable and/or affordable for potential applications, thus won't be considered in this analysis. Our focus will be on satellite-based systems, but we will include a radar-based system example in Table A.2.

Satellite-Based Positioning Technology

Satellite-based localization is the most common and readily used handset-based technology to-date e.g. GPS, GLONASS and the upcoming Galileo. All systems are based on geometric lateration localization methods using synchronized timing and orbital data signals sent by satellites to the receiver device. The point location of a target is provided in LLA and WGS84 geographic coordinate format along with accuracy, speed and heading information. They provide localization accuracies in the region of a few meters on a global scale [ZI06a][ETZ05]. Different localization service accuracies are provided via publicly open and military restricted frequency bands (e.g. SPS and PPS). Apart from the system-based accuracy degradation method (i.e. SA mechanism [ETZ05]), natural accuracy degradation is caused by the urban canyon problem, i.e. lack of visible satellites to the receiver unit in dense inner-city scenarios. Satellite-based localization principally does not work in indoor environments although some manufacturers claim that their receivers have a high enough sensitivity to capture strongly damped or reflected satellites signals [ETZ05].

The second drawback of standard GPS is the cold start initiation time of a receiver unit. It can take several seconds, worst case even minutes for a GPS unit to find, synchronize and decode the signal data of all satellites within viewing range, hence making it unsuitable for instant LBS availability. There are no training costs and receiver hardware costs are miniscule, thus allowing broad end device range application scenarios, e.g. smartphones, PDA, notebooks, onboard car navigation etc. Still, high power consumption of GPS receivers is a problem, especially during prolonged time of position fix.

There are several enhancements improving the accuracy in obstructed usage environments, initiation time and service availability of standard GPS in the form of network/infrastructure-assisted positioning system approaches e.g. ground station-assisted DGPS [LKD03], the mobile operator equivalent AGPS using the GSM network [Sna06], the even further accuracy-improved EGPS using a combination of AGPS and the GSM Timing Advance signal [DH05], as well as the satellite-assisted WAAS, MTSAT and EGNOS systems. Apart from the AGPS most of these systems require costly usage licenses and/or hardware/software modifications to the handset.

The future of satellite-based positioning systems and availability to a broad spectrum of LBS applications is very promising. New modern satellite systems are being rolled out improving the overall location service, accuracy and global availability. However, the performance of satellite-based systems in urban canyon and indoor environments will still be problematic. Current research focuses here follows the lines of hybrid solutions, e.g. AGPS, indoor DGPS, etc. [LKD03]. GPS is widely distributed and will be followed by its improved predecessor GPS 2R-M in the near future. Galileo might have a difficult start due to the GPS end-device market penetration, yet the EU and other countries will eventually market-push this new, independent standard.

2.2.5 Communication Network Dependent Positioning Systems

Positioning systems available in current wireless networks are potentially available in almost any populated living environments to-date. The main function of wireless networks and technologies is to provide resource efficient and high quality data communications and not localization. But as a value-added byproduct, the radio resource base station ID, timing or RSS communication link information can be used for virtually all localization methods presented in section 2.2.2.

Mobile Operator Network-Based Positioning Technology

The currently available and standardized localization technologies for mobile operator-based GSM and UMTS networks use several of the previously described localization methods.

Network-based *COO* positioning is the most common and readily implemented mobile operator proximity-based localization method using the GSM base station Cell ID (or respective Service Area Identifier (SAI) in UMTS) where mobile devices are currently registered. As mentioned earlier, the base station location represents the location of the mobile user with an accuracy corresponding to the service cell size. Due to the huge cell size differences, the location error can range from a few hundred meters to several kilometres. Since the operators exactly know the geographic locations of their base stations, this type of location method is readily available at virtually no initialization cost or special mobile device/infrastructure requirements. The COO method is less suitable in UMTS due to "breathing cells" (i.e. dynamic service areas) and multi-cellular services (i.e. simultaneously registered at multiple cells with different cell sizes).

Several time-of-flight-based, geometric lateration localization methods have been standardized and partially implemented in mobile operator networks using synchronized signal runtime measurements from the mobile device to three other base stations e.g. the handset-assisted TOA, UL-TOA, TDOA, UTDOA, as well as EOTD localization methods [Sar02][Kue05]. All work very similar but differ in timing method (i.e. absolute vs. difference time values), special hardware and software requirements on either infrastructure or handset, or use hybrid localization forms (e.g. a combination of lateration and angulation methods).

AOA-based methods have also been specified and partially made available in operator networks. As explained in section 2.2.2, this network-based localization method requires costly multi-phase antenna array upgrades at the base stations. The AOA technology is so far implemented in only very few base stations throughout Germany (only the bigger providers T-Mobile and Vodafone) and by U.S. mobile network.

The localization accuracies range from several meters to several kilometres in mobile operator networks which are sufficient for many LBS applications (see section 2.1), but the potential costs are too great for mobile operators to roll-out these technologies on a global scale without too much impact on their existing networks or handsets. Furthermore, when taking into account that mobile devices need to provide more sophisticated technologies to enable rich multimedia content and games, battery power optimized solutions are needed contrary to power consuming localization mechanisms. The current choice of mobile operators is on the AGPS technology since it has very little to none impact to existing network infrastructure and has been proven feasible for even small electronic mobile devices.

The IEEE802.11 Wireless LAN Standard (a/b/g/n)

The deployment of the IEEE802.11 Wireless LAN (WLAN) networks or *hotspots* has drastically increased in the past years due to the high comfort of WLAN usage and the steadily improving performance of WLAN hardware and standards. Localization in WLAN networks and available commercial products in this field rely mainly on RSS-based and handset-assisted lateration [DZ02], scene analysis [DZ02][Eka06], probability-based [GJK04][SKS03][WNY05], or combinations [YAS03][DZ02] of these localization methods. These are usually deployed in closed, fairly static indoor environments (e.g. offices) with existing WLAN infrastructure.

The biggest downside of these approaches is the extensive training cost in order to achieve acceptable localization accuracies. Furthermore, highly dynamic environments render an RSS-based system more or less useless, often having to re-initialize the positioning system periodically. Nevertheless, possible indoor location accuracies are in the range of only a few meters so that "room accuracy" can most certainly be achieved on a regular basis [ZDS04]. Some WLAN positioning system prototypes exist relying on time-of-flight localization [Aer06], hence do not require a training period or suffer as much from changes in the environment than their RSS-based counterparts. Here localization accuracies are in the range of a few meters which is comparable to the WLAN RSS-based lateration and fingerprinting methods [DZ02]. However, these systems require costly modified WLAN hardware (client and/or APs) to provide sufficiently high timestamp resolution and they suffer from multipath reflections especially in indoor environments.

WLAN hotspots have much smaller cell sizes compared to cellular GSM/UMTS networks. Hence the current trend to effectively provide location specific services for hotspot locations such as hotels, coffee shops or supermarkets, is to rely on short range networks such as WLAN. Due to the enormous growth of WLAN hotspot locations in densely populated areas (almost 70-80% coverage in inner-cities [Wig06]), hotspots with their fairly small cell sizes of several meters can provide a very good COO-based localization outdoors by using the unique AP MAC addresses. This is only feasible if the geographic locations of the hotspots are known which is difficult to achieve in a multiple provider landscape. Furthermore, although WLAN hotspot APs in offices and homes usually remain in a fixed location for a long time, some are switched off when not used, and the evolution rate of APs within inner-cities follows half a year to a year cycles. We will discuss these issues in later chapters in more detail.

Bluetooth

Virtually any device can be supplied with Bluetooth, covering a service area of several centimetres to several meters. Similar to WLAN, network COO-based localization can easily be implemented with Bluetooth using the unique 48bit device address as an identifier resulting in location accuracies corresponding to the service area. Neither signal propagation time nor signal fingerprint positioning methods are feasible since the Bluetooth v1.1 standard does not specify clear RSS measurements, i.e. only three categories of “acceptable”, “too weak” and “too strong”. This is supposed to be different in the upcoming v2.0 standard. Some experimental research has been done in actually modifying the Bluetooth hardware itself, resulting in timing-based positioning capability with an accuracy of less than a meter [FDW04].

Ultra Wideband (UWB)

Unlike other wireless systems which utilise spectrum in discrete narrow frequency bands, UWB operates by transmitting signals over a very wide spectrum of frequencies making it useful for short range, large data transfers. Commercially available UWB localization systems using TDOA-based methods achieve location accuracies down to a few centimetres LOS in a service range of a couple of meters indoors and outdoors [TSH05]. However the ranging accuracy heavily depends on the signal-to-noise ratio and multipath problems incurred in the usage environment, currently still requiring quite costly hardware configurations [DNB05].

2.2.6 Identity-Based Positioning Systems

Another kind of location-inferring system type can be defined relating to identification systems. The so called *identity-based* systems use the trigger event of an identified target, e.g. by means of swipe-card, barcode or RFID sensor identification tags. Information stored on the tag is usually static and moves along with the respective target. The detection usually occurs on the proximity (usually a few centimeters) of a tag to a respective reader device. Stand-alone, one-way, passive solutions require a special receiver device. In passive variants, the energy needed to transmit or read out the information form the tag is provided by the reader (i.e. transponder device). In active variants, the tag itself has its own power source and can potentially transmit its information over larger distances.

The most common and promising technology in this field is RFID [Fin00]. This term includes all technologies that use radio waves to uniquely identify individual items. There are *active* and *passive* tags. Low frequency, passive tag RFID systems have short transmission ranges, typically only a few centimetres and up to two meters. Active, battery-powered RFID solutions usually use high frequencies and offer long transmission ranges of several meters. Passive tags are a lot cheaper than active tags and can be printed onto flexible surfaces such as product labels. The lack of standardization has the effect that momentarily, each RFID tag vendor has their own standard for frequency, communication and design which makes interoperability almost impossible. Furthermore, RFID is still too costly for cheap consumer products e.g. supermarkets. Privacy issues and security mechanisms protecting the data stored on the RFID device are still ongoing research and industrial debates.

2.2.7 Summary Discussion

Having looked at the currently available and most promising localization methods, respective positioning systems and technologies, we can summarize the following key points:

Concerning accuracy, precision and service coverage the ideal case is to provide a fairly high accuracy (on average below 10m) sufficient for all LBS types on a global scale. Clearly, GPS and Galileo (will) set the current and future standard in providing this level of location quality. For commercial LBS in densely populated, obstructed or even indoor environments this level of location quality can only possibly be provided by means of hybrid/assisted positioning systems, e.g. AGPS, DGPS, EGNOS etc. at much higher cost. Here, political implications (due to terrorism and user privacy), infrastructure and licensing costs, as well as end-device battery power implications remain the critical factors. In the long run, the still existing problem will be the lack of availability in indoor environments and even more so the provisioning of appropriate locally accurate location content (e.g. maps), as the next section on LPs will elaborate on further. This requirement is usually fulfilled by locally available, short range positioning systems, which are customized to the local usage environment and can provide the respective location accuracy e.g. WLAN, UWB, Bluetooth etc. Current cellular technologies used by GSM and UMTS (e.g. AOA, TOA EOTD) cannot provide the necessary accuracies (now and in the near future) and it is expected that the sure bet for mobile operators – the current driving force of LBS – will be in the lines of AGPS.

The initialization cost is a twofold issue: higher infrastructure and device costs (usually hardware) are usually incurred by geometric timing-based positioning systems resulting in proprietary non-standard conforming solutions, whereas training costs are virtually non-existent. On the other hand, scene analysis, proximity-based and statistical positioning system solutions usually only incur much lower software-based infrastructure costs “on-top” of an existing infrastructure, yet with training costs proportional to the required service area. Current figures of deploying fingerprinting or probability-based positioning systems specify an effort ranging between 500-750m²/h in order to achieve location accuracies between 1-5m for standard office environments [Eka05][ZDS04][DZM04]. The training cost varies depending on the environment dynamics, infrastructure, manpower provided (i.e. users or service technicians performing the initialization process), type of initialization methods and strategy used [ZDS04]. Hence these kind of positioning systems provide a competitive alternative in small usage environments, e.g. indoors.

The timing-based positioning approaches in short range standards such as WLAN and Bluetooth contradict the current standards developments and can be seen as only proprietary solutions now and in the near future. UWB has problems concerning regulatory, distribution, technical and market acceptance issues. Compared to Bluetooth and WLAN solutions they clearly lack the “huge head start” concerning market penetration and availability. However, UWB could be interesting for radio access critical environments such as hospitals. RFID is seen a very promising technology for product chain improvements, warehouse management and logistics applications [Gar05].

2.3. Location Service Provisioning

The LBS ecosystem identifies three distinct participants: the *end user* who uses a LBS to obtain a service or information relating to his current physical location or remote location and/or entity (user or object) of interest; the *location service provider* who serves this location information on an entity and additional *location services* via a *location platform* for a designated user environment; and the *application provider* using the location information and services to offer the respective LBS to the *end user*. There are various advantages and disadvantages on how LPs are structured. These will be presented and evaluated on various LP approaches in the following sections.

2.3.1 The Location Platform Principle

LPs enabling any kind of LBS all share a common structure as depicted in Figure 2.4. The main difference in LPs is how location information is generated from respective *positioning systems* (i.e. from the device and/or infrastructure, e.g. network) as well as how/wherefrom additional location content is provided (e.g. map data, navigation and Point-Of-Interest (POI) services, POI data, etc.). Hence the LP itself can be broken down into two major components: a *Geo-Information System* (GIS) holding appropriate map as well as POI data, and a *Location Enabling Server* (LES) providing the location information of a positioned target in some meaningful format for the GIS system. The LES mainly comprises the positioning system part of a LP. A LP can have multiple LES components to support multiple positioning technologies (e.g. for hybrid solutions, see chapter 8). As we have shown in section 2.2, the positioning system parts can be distributed across both client and infrastructure components. On the other hand, the LES is associated with the component part holding the “majority” of the positioning system by providing the single point of contact for location information retrieval on positioned entities (i.e. the location service) privacy and access control, a common information storage (e.g. positioning reference data), as well as potentially computational intensive data processing functions causing too much overhead in other related components.

Nevertheless, the distribution of the positioning system parts has advantages and disadvantages giving rise to different LP approaches, as we will show next.

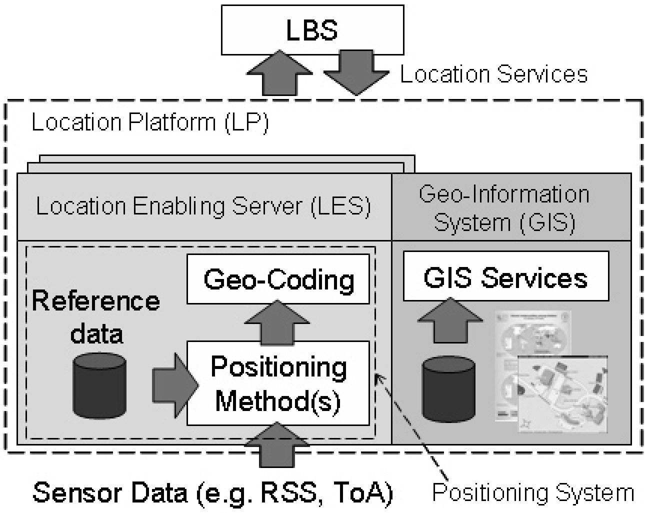


Figure 2.4 A generic location platform model

2.3.2 Evaluation Criteria of the Location Platform Concepts

The to-be presented LP approaches show different motives on providing a location service in a best possible way. A potential "silver bullet" LP solution can be outlined by the following features which we will use to discuss current LP approaches:

- ***Location service availability:*** a location service provided by a LP should be available 24/7, especially in emergency situations. Furthermore, it is of importance whether the location service is restricted to the current local environment or globally accessible. This also relates to the coverage/scope aspect of supported positioning system(s) by the LP (see section 2.2.1).
- ***Heterogeneous positioning system support:*** multiple positioning systems need to be supported and controlled by the LP to ensure positioning flexibility, availability and higher accuracies since no single solution can ensure 100% coverage.
- ***Interoperability:*** this feature relates to the fact that no LP can provide location information for all usage environments (e.g. specific buildings). Hence interoperability is required between various LP operators for "seamless" availability of a location service.
- ***Scalability:*** a LP solution should be able to support all required location request within its service area at all times. Due to the nature of some LP, this especially relates to the signaling and processing overhead incurred due to location update requests and potential handovers between LPs
- ***Cost:*** as shown in section 2.2.1, the cost factor constitutes of several aspects relating to both user and infrastructure in terms of hardware, software and communication costs. In the LP case a higher weighting lies on the communication/signaling overhead endured costs e.g. sensor data transmission. Ideally a LP has little to none impact on the user in terms of additional software/hardware on his device, and data transmission costs incurred in providing his data to the location service.

- ***Privacy and Access Control:*** a LP should enable the user to have full transparency and control over his location information in a non-intrusive way (e.g. constant user queries). Transparency relates to which other entity requests the user's location information. Furthermore, the user should be able to specify privacy settings easily, and a centralized control entity should enforce the access rights on user's location information to 3rd party application providers or other users, respectively.
- ***Standardized interfaces and location formats:*** the LP should provide a standardized interface for location service access (e.g. OpenLS [OGISa]). Concerning local indoor LP, mapping functions should be provided converting from a proprietary indoor to global location formats supporting interoperability.

2.3.3 Current Location Platform Approaches

We can currently associate three distinct types of LP approaches: *device centric*, *decentralized middleware-based* and *centralized* LPs. Each type is associated with a client (i.e. the tracked entity) and/or infrastructure components residing in the local network or some remote location, as illustrated in Figure 2.5. In each case the LP parts are localized or distributed among the components and there exists communication/interaction relations between them (indicated by the arrows).

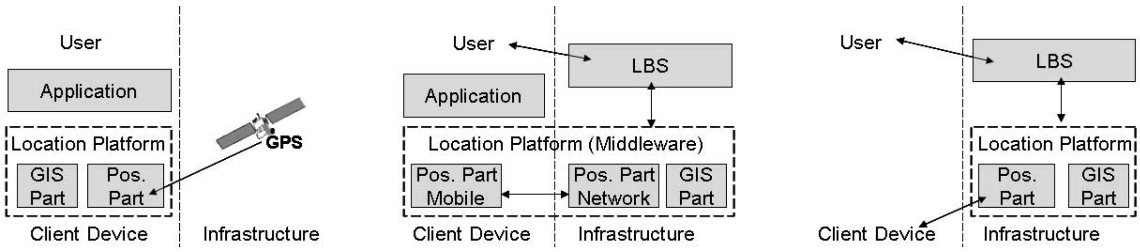


Figure 2.5 Different location platform classifications

To be exact, the client and/or infrastructure components are always involved with the positioning system in some form. Similar to [Nok01] the distinction, however, is made whether the respective component has an "active" role in the positioning process, i.e. is actively manipulating and providing sensor data, or if it is "passively" providing relevant sensor data at no additional costs to the user i.e. additional transmission or processing efforts, e.g. obtaining the current Cell ID the client is registered with from the network radio resource management infrastructure, rather than from the mobile client directly and transmitting it to a requesting entity via TCP/IP connection.

Device Centric and Peer-to-Peer-Based Location Platforms

The *device centric* LP type (Figure 2.5, leftmost image) can be typically referred to stand-alone onboard car-navigation, off-board mobile PDA or classical dedicated navigation device-based solutions. Here the user accesses the location-aware application on the device which has direct access to locally stored GIS content and functions (e.g. maps, routing services etc.) and positioning system information (e.g. GPS/Galileo sensor). This also means that the positioning and geo-coding is entirely performed on the device and additional data for positioning calculation is obtained from the "infrastructure", i.e. transmitted by GPS satellites, broadcasted via radio etc. as indicated by the arrow. It is also possible for the *device-centric* LP to obtain remote GIS information from a web-service (e.g. Google maps) via a device communication interface, but it still represents a device centric solution where privacy and access control on location information is in the hands of the user.

The *device centric* LP type is also the basis for *peer-to-peer*-based LP approach in which location information is directly exchanged between peers [ZD05a], i.e. another *device centric* LP device or remote application in a peer-to-peer network, as shown in Figure 2.6. This LP extension can be referred to as *direct-device centric* [KTL06]. Due to signaling overhead these kind of LPs work well for single event LBS, e.g. friend finder or LBS information services, but don't scale well for tracking or navigation services. Furthermore, these LPs rely on *self-positioning* which limits the type positioning systems capable of being supported to satellite-based ones.

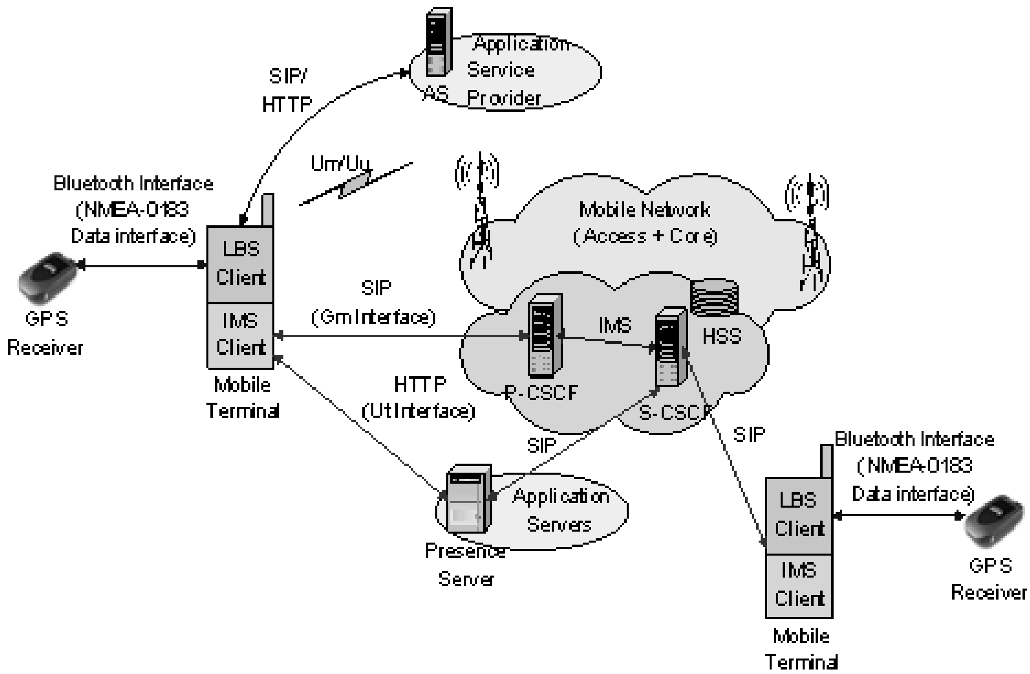


Figure 2.6 A peer-to-peer-based LP example [ZD05a]

Centralized Location Platforms

The other most common LP form available on the market today is a *centralized* LP approach (Figure 2.5, rightmost image). In this case, the client device of the user takes an indirect "passive role" in the positioning process, i.e. the infrastructure is able to obtain sensor data from the infrastructure component the client is currently associated with, e.g. GSM Cell ID COO-based positioning. The GIS component is also stored in a database online which helps keeping GIS data up-to-date. Since the entire LP resides on the infrastructure side, location information provisioning occurs at no additional costs to the user and scales better for multiple users and frequent location update requests of tracking or navigation type LBS. The only "real cost" generated for the user is by accessing the LBS online via an active communication link. Furthermore, access and privacy control on location information to 3rd party application providers is centralized at the infrastructure, i.e. operator's side, but positioning carried out and access granted only on the user's consent.

For current mobile operator networks the Open Mobile Alliance (OMA) standardisation group specifies a *centralized* LP framework providing the functionalities required supporting LBS via standardised interfaces from the operator's core network (see Figure 2.7). The location services from the LES and GIS (i.e. Geo-Toolbox, GTB) component of the LP are provided to the LBS application provider via a standardized XML API using the MLP protocol [OM04c]. This API is the only interface to the LBS application provider providing the necessary location information. The position estimates as seen in step 4 in Figure 2.7 are performed in the access network by the corresponding network entity, handset or hybrid positioning method (e.g. AGPS), then forwarded back to the LP via the GMLC *Le* interface.

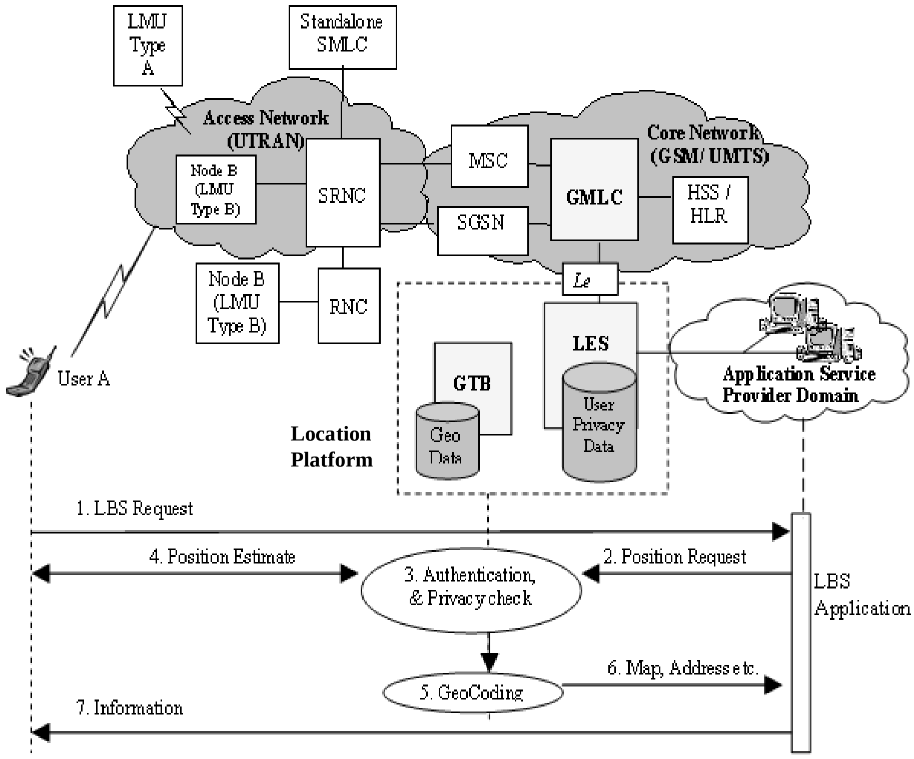


Figure 2.7 Mobile operator-based location platform

One of the additional primary tasks of the LES is to perform privacy and access control including temporary storage of user data, provide configurable user privacy profiles for each subscriber and handle access rights quotas of external application providers. The Secure User Plane Location (SUPL) framework has recently been specified by the OMA [OM04a] with emphasis on location privacy, security, LBS roaming support and generation of appropriate LBS billing records. Furthermore, advanced mobile operator-based positioning is supported (i.e. Enhanced Cell ID and AGPS). As we have seen from section 2.2.5, the mobile operators have favored network-based localization means, but soon have realized the impeding infrastructure upgrade costs. Hence AGPS has become the most viable approach next to the GSM Cell ID mechanisms ensuring backward compatibility to legacy devices. Although AGPS is network-assisted based on *self-positioning*, the signalling and transmission of generated location information to the infrastructure occurs "silently" at no costs to the user. Hence, access and privacy control remains centralized at the operator [Hul04]. The mobile operator's intend is to keep control over location information access, thus being able to generate revenue from LBS. Privacy control is implemented by querying the mobile user whether he wants to let himself be positioned before he can carry out the LBS. This legally upholds the user's privacy rights, but does not give any indication to which 3rd party this location information is handed to, and has considerable usability drawbacks.

Mobile operators with their LP implementations are limited to their own networks and can only serve their own mobile customers – who have agreed to be tracked – with LBS, unless roaming mobile users from other operators are willing to pay higher rates to receive those services. The OMA standard and EU regulation foresee LP roaming capabilities in the near future, but political and lack of agreement on OMA standards implementation are prolonging this fact, hindering an efficient global deployment and slow down the introduction of AGPS overall. Furthermore, the OMA standards currently limit LP roaming to mobile operator networks, thus positioning capabilities are bound to those available within the target network, lacking consistency in availability and global accuracy.

Decentralized Middleware-Based Location Platforms

An in-between approach combining both *centralized* and *device centric* LP concepts is the *decentralized middleware-based* LP type (Figure 2.5, middle image). Here, both client and infrastructure components have active positioning system parts. The GIS part still resides on the infrastructure side although in some sensor network approaches [LMS04] the sensor data is distributed across all peers. There is an active communication link between the mobile and infrastructure positioning parts which is hidden from the respective LBS or application part via the middleware. But this is also the biggest drawback of this approach since all data exchanged between the mobile and infrastructure is sent via active data communication links which usually incurs additional traffic costs to the user e.g. by using GPRS or UMTS. Concerning hardware/software costs, the decentralized LP middleware would require a software client to be adapted to each new client system. This again could also have an effect on increase mobile device power consumption.

So far there is no standard defining *decentralized middleware-based* LPs, hence current solutions are proprietary and in a research state. This also applies to respective privacy and access control concepts. Since the sensor transmission involves active client part and communication link, the user can always opt-out on whether to provide his location information or not. There are a few concepts relating to instant messaging, SIP-based and IP Multimedia Subsystem (IMS) solutions trying to deal with the privacy and access control issue on location information [ZD05b].

Indoor positioning systems using WLAN RSS-based localization are based upon this LP concept [Eka05][ZD02]. The client positioning part has the responsibility of gathering visible access point MAC Addresses/ESSIDs and RSS information, transmitting this sensor data to the corresponding infrastructure positioning part. The latter performs the actual positioning using this- and pre-registered location reference data. This principle can also be applied to Bluetooth network identifier, UWB, GSM Cell ID, RFID tag or similar technologies, respectively. Since client devices become increasingly *multimodal,* i.e. have several communication and data interface extensions, multiple sensor data information can be provided simultaneously enabling *heterogeneous positioning*. Thus the decentralized middleware-based concept is often described as an *operator* or *positioning system independent* LP approach having the flexibility and extensibility for future LP requirements.

2.3.4 Summary Discussion

In this section we have presented different LP approaches, each having advantages and disadvantages which have to be weighted against each other in order to form a global, "silver bullet" solution. A summary of the discussed LP approaches is given in Table A.3 of the appendix section A:

A centralized LP approach as pursued by the mobile operators presents the ideal solution concerning location service availability to the application provider (i.e. standardized interface and roaming support to other operators, but at higher cost to external network customers) as well as scalability. However, positioning technology support is limited to the respective mobile operator network domain and roaming to other non-mobile operator location service providers is not specified and supported. A decentralized middleware-based LP approach is the most flexible in supporting heterogeneous positioning systems and location service provisioning (i.e. centralized or decentralized via the mobile device a.k.a. device-centric LP approach) working independently in heterogeneous network environments. On the downside, scalability is a problem due to the higher signaling cost involved. Means of optimizing and reducing the signaling overhead is by intelligent signaling protocols which use dead-reckoning, data compression and estimation/interpolation algorithms [Leo03][TK05]. In both centralized and decentralized approaches, the need for open standards concerning heterogeneous positioning technology and domain inter-working is a must to ensure the often talked about and requested ubiquitous location service availability [ZDS04]. Last but not least, privacy is a two-fold issue: location information should only be provided to a (trusted) known requesting peer or application service provider of the user's choice, thus offering him full control over his location information and hence over his privacy. A decentralized middleware-based or device-centric LP implicitly allows this kind of control which increases user awareness and trust for LBS [ZD05b]. Mobile operators should exhibit a similar trust on controlling access to and privacy on user's location information relating to 3rd party application providers. However, this is not handled very transparently and user friendly in respective LBS applications.

2.4. Location Information Generation and Provisioning by User Communities

Communities are currently seen as a revolution for mobile and internet-based applications. New innovative community platforms based on Web 2.0 services fulfilling specific community needs are being deployed every day. In particular, the interesting aspect of communities is on how they generate and share information as well as services between fellow community members hereby collaboratively generating value [Hu04]. The goal of this section is to introduce the community principles and practices to show how they can be used as an alternative location service provisioning means and potential improvement to existing approaches.

2.4.1 Description of a Community

A ***community*** is defined as a group of people having common interests, sharing information between each other, meet online on the internet and/or are living in the same locality under the same government [Koc03][Mer01]. On the internet world, a community is remotely distributed across a potentially large physical space, but brought together virtually, hence often referred to as a *virtual-community*, *online-community* or *cyber-community*. Hereby, members communicate and exchange information using electronic devices, networks and services via a community platform. Virtual communities which have the need of accessing and sharing information or services with potential local relevance while being on the move with other mobile community members are referred to as a *mobile communities*. In this case, the respective location context of each community usually has particular relevance in also finding fellow community members and bringing them together physically. This has led to the development of another phenomenon we can also define as *ad-hoc communities* or *flash mobs* [Rhe03] where members sharing a common purpose/goal, temporarily physically get together for common task or knowledge exchange, and disbanding soon after that common task has been achieved.

The Three-Layered Community Model

Virtual and mobile communities can be represented in a three-layered model as given by [FT02]. The modeling views can be specified as 1) the "technical" web-platform, 2) the hereby resulting communication space, as well as 3) the community of people who more or less use this platform on a regular basis. Figure 2.8 illustrates this model. Layer 1 represents the platform and infrastructure for realizing and visualizing the needed information and communication services for the community (e.g. providing chat, IM, forums, wiki, etc. by means of a web portal, or dedicated systems such as Skype, ICQ, etc.). The community platform can be seen as a *product*, the value of which is represented by the communities' *knowledge base* constituting of both number of participating users and information available.

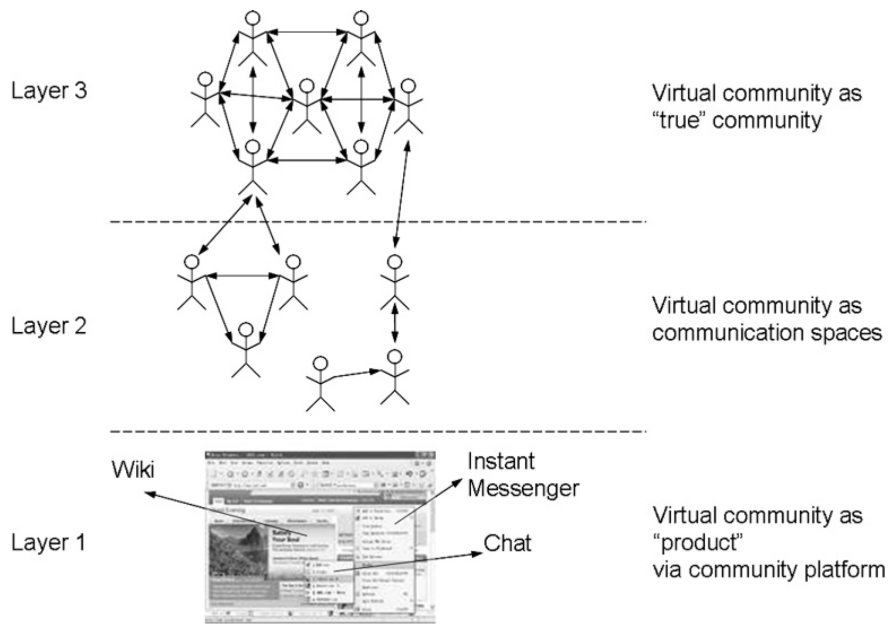


Figure 2.8 Three-layer model of a community [FT02]

The provided communication and interaction services by the platform provide the so-called communication spaces (layer 2). This inherits a variety of communication possibilities depending on the number of addressed community members and frequency of interactions, i.e. synchronous (e.g. chat) or asynchronous (e.g. mailing list) communication, one-to-one or one-to-many addressing of community members. Last but not least, groups of community members interact and develop/grow within the provided communication spaces, i.e. layer 3.

Motives for Participating in a Community

Apart from serving the need of establishing social contacts, finding likeminded or people in similar situations, the basic goal/motives for participating in a community is the need for exchanging and obtaining information or a particular service the community can provide. The information hereby exchanged can have both *explicit* and *implicit* character [Koc03]. *Explicit knowledge* relates to published media accessible to all users whereas *implicit knowledge* is inherited and tightly bound to the individual user/knowledge bearer (e.g. user experience, local events he is currently aware of). Communities support the knowledge exchange between individuals and through this dialogue help externalize implicit knowledge and turn it into explicit knowledge available to all members.

Community Types

Communities can be categorized into four major groups [Car99]: communities of *interest*, *practice*, *purpose* and *passion*. In reality communities possess features of multiple community types at different levels. The community of interest is the largest and most diverse community type constituting of members having different backgrounds, knowledge, interests, motives, and tasks to fulfill. The other communities are much smaller and are more focused to a particular common task, interest, goal, etc. It is often the case that a community of interest contains several different sub communities representing the other community type characteristics. A community can change from one type into another e.g. specialized sub-groups forming out of community of interest [Car99]. In a community of passion for instance, the intensity of the engagement in this community's interest can attract similar minded and interested people. The growing number of members softens the original homogenous community core, hence slowly evolving the community into a community of purpose, or even into a community of interest with increasing popularization of its interests. This can be seen as the most common form on how communities are created and developed around innovative ideas, technologies and interests, which are spawned by so called *lead users* [Rhe03] growing them into the mass market. This also applies to the community location service concept developed by this work.

2.4.2 Growth and Development of a Community

At the time a community is initially formed usually a very small user community of a few people sharing a common interest, practice and/or passion establishes a community platform to exchange their knowledge and ideas. Thereafter, the growth of the community follows certain rules governing the quality of the information and services provided (i.e. attractiveness to new and existing users), interactivity hence contribution of the community members as well as their incurred satisfaction or frustration with what is being offered by the community (i.e. willingness to recruit new members). The community as a product is represented by the quality of the information represented by the explicit knowledge visible and available via the community, the services provided for interaction as well as the size of its existing user base (i.e. implicit knowledge). In order to ensure a certain quality of explicit knowledge generated, the community itself evaluates and corrects the information generated by means of respective trusted experts within the community and by public community consensus. Common community tools to support this process are recommendation and rating systems on generated explicit knowledge [Koc03].

The growth of a community can be described by positive external *network effects* on systems (i.e. *network externalities*) which describe the relationship between a product or system and its respective number of users [Bar02], e.g. mobile phone SMS service. Hence the usage of the explicit knowledge or product and usefulness of a network grows with increasing user base. The increased usage increases the attractiveness of the whole user network. Thus the user base continues to grow which in turn increases the overall usefulness and value for the whole community, i.e. a *positive feedback* occurs and we talk about *positive network externalities*. Similarly *negative feedback* through *negative network externalities* can occur (e.g. mobile network outages, lost SMS, character limitations within SMS etc.) which can cause users to leave the community. We can further sub-classify direct, indirect and two-sided network effects [Bar02] in both positive and negative ways, but this level of detail won't be necessary within the scope of this thesis.

Although the principles, motives and behavior of communities are still ongoing research, a commonly agreed model representing their development undergoing positive feedback can be described as illustrated in Figure 2.9. As mentioned above, the growth and success of any community relates not only to the technical platform issues but also to the quality of explicit knowledge available. At the early launch phase of a community (Phase 1) the user growth is very slow and steady. A *critical mass* of users is needed who collaborate, generate explicit knowledge, exchange information between the community members, and propagate the value of the respective community further (i.e. attract and recruiting new members) thus keeping the whole community alive [Mer01]. Once this critical mass of users within the community (i.e. *tipping point*) is reached, the growth of the community will continue to grow exponentially or even stronger on its own due to positive feedback, i.e. take-off phase [Bar02]. However, this "unlimited" exponential growth is usually limited by a *saturation* effect almost always occurring in community networks which has to be accounted for (e.g. distribution of mobile phones in a respective market). Saturation is the result of the currently addressable/reachable market with what the community i.e. its product can at the time offer, or possibly a more attractive replacement product, technology or community growing in its place can do better [Hen00][Wol03]. This results in the *tipping-curve* behavior or "S"-curve adaptation as illustrated in Figure 2.9 [DR00][SV98].

There are several laws which describe the growth and respective value of networked communities represented by phases 1 and 2 (e.g. law by Metcalfe [DM98], respective improvements by Odlyzko and Tilly [OT05], as well as Reed's law [Ree01]). The difference between [DM98][OT05] and [Ree01] is that the former sees the growth and value of a community network to be purely a factor of the number of connections established within a community as a whole, whereas the latter only considers the relationships within social networks, hence value of a network in accordance to the number of users and aggregated value of individual sub-groups relating to respective users within the whole community, not just that of an individual user. The bottom line is that every law considers an exponential growth relating to network effects, but with different optimistic or rather cautious growth factors as well as stepwise refinements

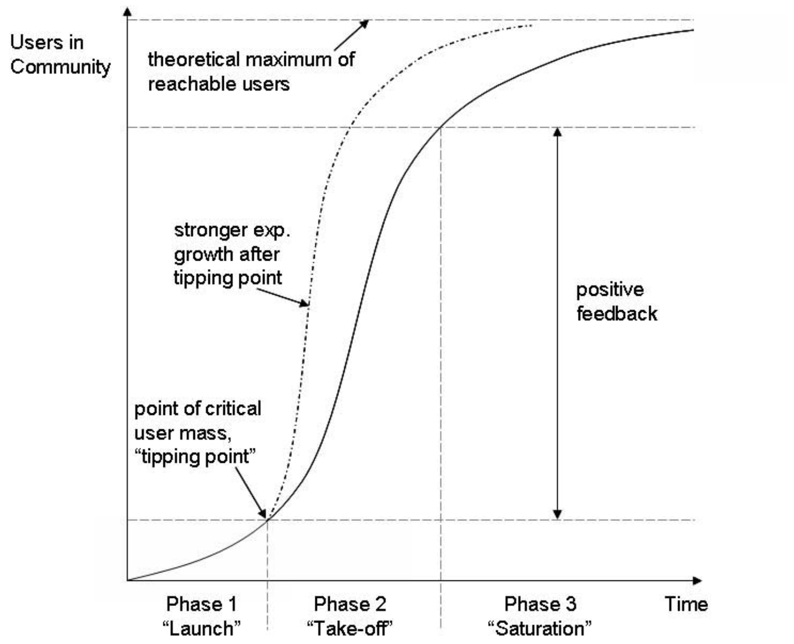


Figure 2.9 The growth of a community following the tipping curve behavior

towards community related and market specific factors. Within the scope of this work a basic common network model will be sufficient for our modeling purposes. As already mentioned, community growth is usually limited by a so- called natural saturation effect occurring in almost every community network, hence allowing us to determine maximum potential size of a community, i.e. n_{max} [SV98][Wol03]. An exact model representing both initial exponential growth and saturation effect relationship is not feasible, yet can be approximated for amount of users *n(t)* at a time *t* by the following formula resulting in the “S”-type tipping curve as illustrated in Figure 2.9:

$$n(t) = \frac{n_{\max}}{1 + e^{-\beta \, n_{\max} \, t}} \tag{2.1}$$

Thus n_{max} represents the theoretical maximum of community members reachable which the tipping-curve exponentially approaches. The β factor determines the growth rate representing the different market effects and growth towards the saturation point determined by n_{max}.

2.4.3 User Integration and Roles within the Community

The most interesting concept about communities is that every user within the community has two inherent roles in an *economical* sense: 1) as a producer of small incremental, small content parts and 2) a consumer of the public good that was produced by the entire community. This is commonly known as *value co-creation* by engaging customers directly and intentionally into the production and distribution of value, e.g. the wikipedia project [KFS99]. Users within a community can also create value non-intentionally. This process is known as *non-intended value creation* which usually serves as a built-in self-improvement during the usage of a system by its users, e.g. Email provider-based spam-filter, customers purchase recommendation engine by Amazon [Hub04].

As we have explained the different community types in the previous section, users with different degrees of knowledge, interest, and involvement exist within communities. Apart from these user attributes the *behavior* and willingness of participation of a user within a community can be classified into three distinct types [Var06]:

- ***Linus***: This user group follows the open source and most idealistic mentality in fulfilling the interests of the community. They usually expect little reward in materialistic or monetary form but rather appreciation of their overall contribution. The expectancy of contributing to the community at higher costs and effort as well as acceptance of incompleteness and flaws within offered services or information is the highest among to the other two groups. Examples here are open source developers or article writers in the wikipedia project [KFS99] etc.
- ***Bill:*** The Bills are prepared to contribute to the system as long as they receive adequate reward and quality information as well as services in return. This user type will usually also pay for appropriate information and services. Bills can be seen as the most neutral user type. Example user types would be people buying MP3s from an iTunes store.
- ***Alien:*** This user group wants to have access to all community information and services at best possible quality with as little contribution needed and costs incurred. Unlike the Linus user group flaws in services and incompleteness of information are rarely tolerated.

As a rule of thumb it can be said that 10% of users in a community contribute 80% of the explicit knowledge available to all community members [Car99]. Hereby, the largest if not the single most contributors here are the Linus type users followed by a much lesser fraction of Bill and next to no contributions by Alien type users. The exact ratios vary depending on the individual community but the given approximation remains more or less the same. However, the distribution of users within a large community is usually Gaussian, e.g. community of interest, where the majority of users are Bills and Linus as well as Alien type members represent a minority.

General risks for communities when integrating users into the value creation process relate mostly to privacy, legal risks and brand reputation problems [Fri01]. Especially the privacy concern has to be addressed when users intend to contribute value to the community, i.e. protect their privacy during this process.

The satisfaction or frustration of the user within the community is closely related to the amount of contribution and costs involved in relation to the reward obtained for the individual user. If one outweighs the other, the user either increases his involvement and contribution to the community, even recruiting new members, or decreases his efforts respectively, leave the community, and in the worst case even propagate mischievous information on the community. Providing *incentives* for contributing can compensate for higher participation efforts and increase overall contributions in a community e.g. member ranking systems, extended privileges, playful contribution by means of games, or simple monetary compensations [Nav06].

Last but not least, every community exhibits an *evolutionary user interaction* process [ZD06]. Since every community usually starts with a small core group of members and little explicit knowledge available within the community the required contributions at the beginning of a community are much higher but will eventually drop as new members may join and more explicit knowledge is produced. This can be also seen by an eventual saturation effect as less and less "new" explicit knowledge is produced hence more and more of the potentially "available" knowledge space is provided, eventually resulting in a mere optimization of the explicit knowledge available within the community. As environments as well as "knowledge" can change or be newly discovered the potentially available knowledge space increases respectively.

The co-creation, user interaction evolution as well as user frustration and satisfaction issues on required contributions and fulfilled needs within the community are important community aspects in our further analysis and development of our set out concept.

2.4.4 Communities Using and Generating Location Information

Mobile communities have a great advantage since they have access to and can share local relevant information or changes in their current living environment almost in real-time provided the right communication tools and distribution methods, e.g. traffic jams and accidents reported by car drivers via mobile phones which are distributed to fellow drivers by local radio stations. Thus, communities using location information can either provide location specific content or information relating to their current living environment or use this information to obtain location specific services (e.g. local events, traffic state, thunderstorm warnings, etc.).

The previously mentioned Flickr community, for example, uses location information from GPS devices and manual address input to geo-reference pictures taken are uploaded and shared online by users via their mobile devices. The picture locations can then be viewed on appropriate online map services. Another community activity called *geocaching* [3] or *geotagging* is a modern form of treasure hunting game. More interestingly a few mobile communities have started mapping the locations of visible WLAN hotspot locations in their local living environments. The so called *wardriving* or *warchalking* has developed into a need for people finding open WLAN networks for free internet access. Already thousands of users are participating worldwide providing up-to-date information on millions of WLAN hotspot locations, and the number is growing stronger every day [WIG06].

Providing Community-Based Location Services

In the early days of LBS, due to the restrictions and high pricing models imposed by mobile operators concerning location information provisioning for LBS, communities[4] have started to map GSM base station Cell IDs in a similar fashion using GPS receivers and special functions on their mobile phones for reading out the current GSM Cell ID. Soon, these passion driven communities developed simple, mobile operator-independent LBS applications using special software clients on their mobile phones. The PlaceLab [LaM05] and Location Trader Project [DFH04] have followed this approach on providing WLAN-hotspot and GSM Cell-ID-based positioning using readily available internet-community databases, e.g. [WIG06].

[3] http://www.geocaching.de
[4] http://www.nobbi.com

In [DH03] first attempts on co-value production by the communities where evaluated by making users manually enter addresses or GPS coordinates of newly discovered GSM cells not yet available in the community database and contribute these during usage of the community-based LBS. In essence, the newly discovered base station represents implicit knowledge inherited by the respective user at that time which becomes explicit knowledge as soon as he contributes its geo-coded location to the community database for other users to use immediately. Hence at the start of the community there are only a few geo-coded base stations available in the community database from the respective communities living environment, thus requiring a lot of initial contributions by the members. The level of contributions drops as the users discover more base stations from their environment (i.e. generate explicit knowledge) resulting in more immediate successful localizations.

Clearly, the number of frustrated users being disappointed with the service and potentially leaving the community will be initially higher than the number of satisfied users. The aim of the community in this case is to have an initial minimum amount of members to compensate for early drop-outs, and to ensure a sufficient amount of contributions for discovering all possible base stations within the given usage environment as quickly as possible for an acceptable location service availability. This may include the use of incentives (see section 2.4.3). The minimum amount of users directly relates to the size of the usage environment area in question.

As mentioned in the previous section the biggest problems relating to user generated location information are user location privacy and quality of the contributed data. Communities usually don't have high privacy and security mechanisms relating to the identity of their users. This is why it is extremely important to handle the contributed location information by users anonymously in order to increase their willingness of contribution and avoid misuse of this data [ZD05b]. A more critical issue, however, is the provided location information quality. Since users have different means of geo-coding hence interpreting one and the same physical location, i.e. by using a GPS device, manual address or other semantic description of the location referencing, there is always an induced (possibly also maliciously intended) geo-coding error which has to be considered when consolidating heterogeneous user contributions. Therefore, there needs to be some restriction and control on the amount of geo-coding methods provided as well as sanitizing, transformation and consolidation mechanisms inherent by the system i.e. community platform. Although multiple users contribute location information on a physical object, this data is often incomplete and only represents partly the true physical position. Thus the provided system heuristics need to always try and generate the best possible location information from the provided data.

Comparing standard indoor and potential community-based location service deployment scenarios in their typical application environment, rough dimensions on initialization efforts and costs can be drafted as shown in Table 2.2. Both communities and positioning systems undergo an initialization/training phase of the knowledge in their respective environment as explained in section 2.2.3, heading towards saturation as more of the available knowledge space is discovered resulting in mere optimization of existing knowledge in the environment, i.e. changes are being adapted to. Also, the more "manpower" i.e. users are contributing in the initialization process, the faster the initialization process proceeds. In both cases the degrees of user interaction, critical user mass and efforts of location information contribution depending on the respective service environment have to be considered. Last but not least, the risks of co-value production and the requirement that a community-based location service has to provide similar levels of location service reliability and availability as existing location service solutions in a particular environment have to be considered.

Type of Location Platform	**Indoor WLAN RSS-based**	**Outdoor community-based**
Localization types	Scene analysis e.g. fingerprinting, Statistical e.g. HMM	Proximity-based e.g. COO
Positioning technologies	WLAN 802.11X	WLAN 802.11X, GSM
Usage environment types	Offices, factories, museum	Inner-cities
Infrastructure state	Failry static	Dynamic
Usage environment size (m^2)/estimated effective usable area (%)	>100m² /70% (per office floor)	>150km² /30% (open accessible spaces to users e.g. streets, public places)
Estimated usual training effort (m^2/h) per person	750-1000 [Eka06]	Depends on user type and mobility
Usual available manpower	1-2	2-100+
Estimated training time given manpower	1-3 hours	Several weeks
Initialization (training) procedure	Structured, with knowledge of the environment	Unstructured, with little to no knowledge of the environment
Optimization and maintenance	Discrete scheduled re- initialization and maintenance events	Continuous self-optimization and learning

Table 2.2 Rough comparison of standard and community-based LP initialization efforts and cost

2.4.5 Summary

In this section we have presented the community principle giving a three layer community model. We have explained the motives for users participating in a community (i.e. community based collaboration) introducing the existing community types. Looking at the critical factors affecting communities (e.g. critical mass of users for positive feedback), the development phases of a community have been illustrated and the laws governing the growth and value of a community explained e.g. laws of Metcalfe and Reed. The user integration and participation methods as well as roles within a community have been discussed which need to be considered when modeling the user behavior. Finally, examples of communities generating and using location information have been given, comparing the community-based initialization processes to common location service provisioning methods, as well as discussing the requirements and precautions of offering a community-based location services.

2.5. Features of a Global Location Service Architecture Supporting Community Principles

Future LBS will require ubiquitous location service access meeting their LQoS requirements respectively. None of the existing positioning solutions can yet, and will in the near future be able to provide a single ubiquitous real time location service with a location information quality representing the ground truth physical location of the corresponding target at a reasonable cost. Hence solutions providing localization and location content best suited for a particular usage environment will be used. The resulting landscape of heterogeneous positioning systems, especially by the abundance and use of heterogeneous networks, will be able to provide an almost ubiquitous availability of location information. Hence, this heterogeneity implies a distributed network of LPs providing respective location services.

Using communities for location service provisioning has the great advantage of having access to locally relevant data and discover/adapt to changes in the usage environment quicker than any existing service provider can, e.g. traffic accident reporting example as described in section 2.4.4. Thus the built in self-optimization and learning features of a community can support the initialization and maintenance of location services (i.e. positioning systems). Combining this with suitable LP approaches (i.e. distributed middleware-based LP) and suitable initialization and localization heuristics can provide a universal, flexible LP model independent of underlying positioning systems and usage environment situations.

Therefore, a global service architecture managing the discovery and provisioning of distributed location services will have the following features:

A LP independent and ubiquitous location service: the global location service architecture needs to provide a homogenous, generic and centralized location service to LBS independent of multiple distributed and heterogeneous LPs providing location services for their usage environment. The global location service should be ubiquitous and highly available providing best possible location information and content for the current usage environment fulfilling the LBS needs, respectively. It should not only provide location information on respective targets, but also on the QoS of the location information and local LP providing it. This is especially important for the LBS if LPs are treated independently and respective LQoS guarantees cannot be satisfied in a heterogeneous LP environment whilst moving from one usage environment to another. Thus the global location service should provide means to negotiate and signal LQoS changes to LBS. Furthermore, the global location service API should implement commonly known location provisioning standards e.g. [OGISa]. Finally, the global architecture should support the discovery of LPs with a service area matching the targets location.

Multiple LP and heterogeneous positioning system support: various kinds of standardized or proprietary LP solutions need to be supported by the centralized global location service. Every LP should be supported irrespectively of the formats and coordinate systems for GIS content as well as location information. The global architecture will provide the necessary inputs needed to the corresponding local LP to perform the localization of the target (e.g. AAA, sensor input, target ID,) and use available location services via appropriate interfaces as provided. Respective queries and responses will be adapted to the respective global location service requirements for evaluation on required LQoS.

Provisioning of best possible LQoS for LBS: having a central location service provisioning of location information and content from various LPs has several immediate advantages apart from single-point of contact for LBS providers:

- Caching and estimation functions, e.g. interpolation of location information to reduce signaling between global location service and original local LP [Leo03]. This can improve service continuity in case of LP outages, e.g. a handover case.
- Provide backup connections to multiple LPs potentially providing location services on the target in case of LP outages, e.g. whilst being in the transition from indoors to outdoors.
- Provide simple transformations between location coordinate formats as discussed in section 2.2.1

- Optional: enable sensor fusion [High03] from multiple LPs in order to achieve required accuracies as specified by respective LBS

Sensor fusion can be computationally expensive and complex. Due to the sensor interpretation and selection problems descried in section 2.2.2, and the fact that the provided location information and GIS content are mostly closely tied together, fusion at this level has to consider the end-compatibility and choice of the latter GIS as well. Although this still potentially feasible, fusion responsibility of multiple sensor sources is possibly kept best within the respective local LP solution. Nevertheless, the global location service has to signal the LBS information on how to handle the provided location data from the currently serving LP.

Seamless interoperability and handover between LPs: since most LP solutions don't specify respective LP interoperability methods, a centralized global location service has to provide seamless handovers between multiple, independent, heterogeneous distributed LP systems.

Infrastructure independent location service discovery: the best solution would be the local LP broadcasting its presence for location services within the local network domain e.g. using available network functions such as DNS, Mobile IP, etc. [ZTB03]. Discovering available LPs in the environment without potential access to the local network domain, as well as general network heterogeneity prohibit this capability. Thus other means need to be provided on discovering appropriate location services independent of and with little impact to the existing infrastructure, e.g. having potential LPs register their with a global directory service e.g. UDDI [W3C04].

Ensuring privacy and access rights on location information: the user location privacy and access rights need to be controlled between LBS and LP providers. Users need to have transparent control on which LBS or LP provider, or which other registered users have access on their location information. The central location service needs to enforce these rules on LBS providers as well as perform access control of users to certain LPs, e.g. is the tracked user an employee or traveling guest of the airport.

In addition, a *community-based* location service architecture needs to provide the following features:

Technology independent learning and self-optimization: the local LP responsible for the usage environment should perform the actual collaborative location information generation and self-optimization. The global architecture should support the signaling between mobile clients and the LP of sensor data captured from the living environment required for the (re-)initialization/training processes independently of end-devices, local infrastructure and positioning technologies used.

Collaborative initialization and robustness to varying hardware configurations: the community-based LP needs to be able to process multiple sensor data contributions (i.e. automated from sensor devices and manual user input) for the initialization and self-optimization process of reference location data. In order to support a broad spectrum of mobile devices, the hardware configuration differences of respective technologies potentially causing considerable variations on captured sensor data need to be accounted for and handled appropriately. Thus appropriate sanitation and integrity checking functions need to be provided by the LP.

Chapter 3
Overview and Evaluation of Existing Location Service Architecture Concepts

The goal of this chapter is to introduce the state-of-the-art research in location service architectures and works relating to the established features at the end of the previous chapter. Relating to the fundamental principles laid out in chapter 2, an overview on the research area focus of the related works will be given as illustrated in Figure 3.1. This will be followed by an introduction of the various approaches, giving a brief evaluation on the set out architecture features presented in section 2.5. We will close this chapter summarizing the evaluation and comparing the related architecture approaches.

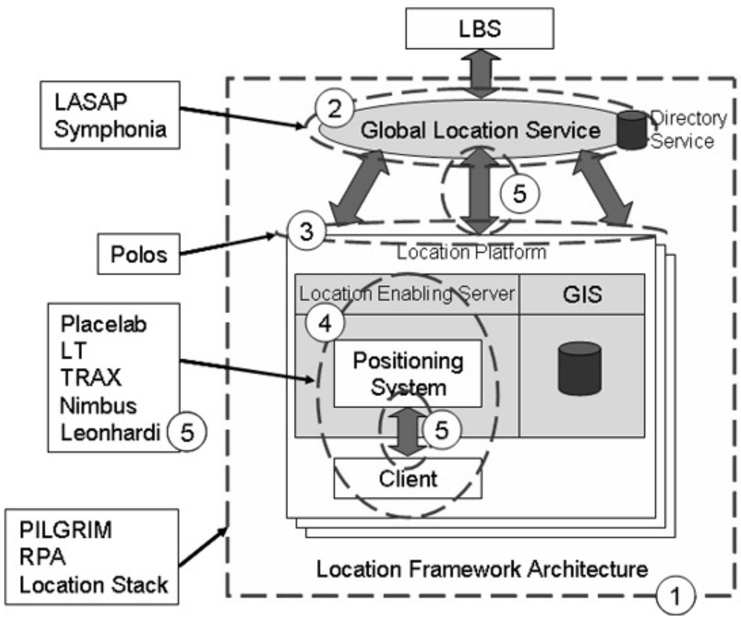


Figure 3.1 Overview on related works and respective focus areas

As shown in Figure 3.1 current architecture approaches dealing with distributed location service provisioning or respective portions of the location information generation and provisioning process can be grouped as follows. The first group (1) represents existing frameworks and architectures dealing with location as well as context-awareness in general. These in effect describe homogeneous context and location information modeling, processing and provisioning from multiple distributed sensor sources for requesting applications. A second group (2) closely associated to the first are location broker models and SoAs [KBS04] having the main purpose of discovering and providing location services for applications, supporting required user, and service control functions. Groups (3) to (5) represent part solutions of potential location service architectures and desired features as described in the previous chapter. The group represented by (3) exemplifies existing standards for location service provisioning and frameworks using and extending these as well as supporting proprietary solutions. Group (4) illustrates LP solutions for location information provisioning as presented in section 2.3.3. Unlike location architectures and broker models represented by groups (1) and (2), examples in this group lack potential user, service and privacy/security management functions, focusing on location information processing. Finally, (5) looks at solutions for intelligent location information signaling of location services and sensor data.

There are several other possible approaches and combinations of location service solutions, but the chosen related work examples best represent various requirements as set out by this work.

3.1. Context-Aware Architecture Frameworks focusing on Location

A context-aware architecture provides a framework for determining this context information from multiple sensory inputs, and providing this value added information to applications or respective services to entities in this context. According to [MK05] viable context-aware systems are applied in limited and closed domains, e.g. smart home or meeting room. This is because the complexity of such a system grows, and interpretation as well as determination of context becomes increasingly difficult the more heterogeneous sensory inputs are considered. On the other hand, providing context information beyond such usage domains for more sophisticated and global applications (e.g. seamless navigation and tourist guide) using context information from multiple potential context-aware systems requires so called context trading or broker systems. Context brokers are intelligent software agents which help to match, supply and demand context information from multiple domains at as little incurred costs to the requesting applications (i.e. the functions are more along the lines of context management)

Many frameworks and architectures with different approaches to context and/or location information processing and provisioning have been developed, e.g. CoBrA [Che04], Nexus [DHN04], RPA [Pfe05], The Location Stack [Hig03], etc. Since context interpretation in the broad sense requires very sophisticated features such as context modeling languages and ontologies for knowledge sharing and reasoning needing to be explained which would go far beyond the scope of this work. Hence we will restrict our analysis to those approaches dealing with location context information primarily.

3.1.1 The Redundant Positioning Architecture (RPA)

The main focus concerning the Redundant Positioning Architecture (RPA) is the positioning determination from multiple sensor sources even only partly designed for this purpose in the first place. Hereby the fusion of different context information sources is performed in order to achieve the highest possible confidence about identifying a user/object and his/its current physical location, i.e. through redundancy of potential location indicators/technologies in the usage environment [Pfe05]. Example location sources include visual, biometric, pressure pad, network IDs, user device type or plain dedicated technologies such as GPS. Figure 3.2 gives an overview on the respective architecture components.

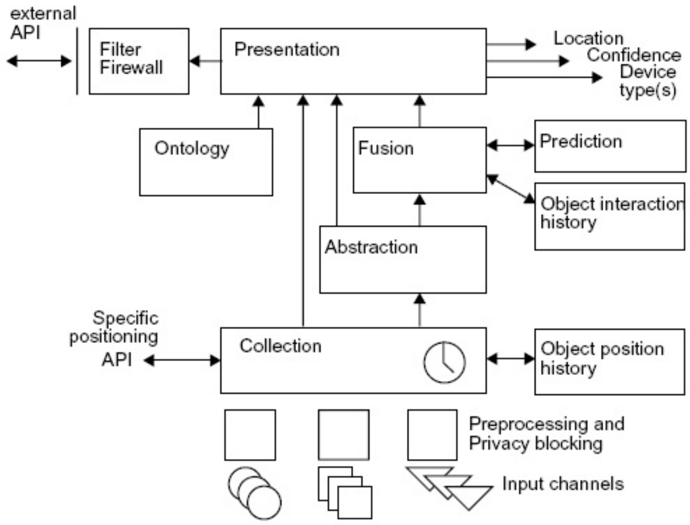


Figure 3.2 The Redundant Positioning Architecture [Pfe05]

Describing the system bottom up, the input channels can utilize any of the described localization technologies. The heterogeneous input data requires preprocessing tailored to different technology. A privacy filter stage enforces privacy and security settings and filters out protected data before processing, e.g. unauthorized scanning of a biometric RFID chip in a passport. The collection module provides plug-in sockets for the input channels and acts as an intermediary buffer for time-stamped location requests, later source comparisons, as well as estimation functions, implementing respective positioning APIs. The abstract module converts location data from different formats into the one required for fusion purposes. The fusion module performs fusion of multiple input channels, considering potential fusion capability and pre-processing of respective input channels level. History input datasets and feedback of fusion results can be used for motion estimation/prediction purposes as well as for self-improving and -healing in case of localization problems. Results from the fusion and other processing modules are provided to the presentation module along with confidence levels which ensures that the application has all relevant information in order to determine the best possible localization result fulfilling its needs. Finally, the ontology module should provide extensibility for alternative context processing other than location information. Implementations of such systems are seen for individual administrative domains and usage environments, whereby inter-domain location information exchange and access control should be realized via the external firewall forming peer-to-peer communication networks.

The RPA provides an interesting theoretical concept for homogenous location information provisioning and localization by means of multiple heterogeneous and redundant sensors in the environment, but leaves many open questions regarding the inter-domain signaling of location information, structuring of respective location service domains, user location privacy, suitable location models for fusion, etc. It is still not clear whether fusion of multiple heterogeneous location sensor types makes sense. A respective system would need to evaluate and consider this, and/or provide all outcomes to the requesting application to choose from.

3.1.2 The Location Stack

One of the first successful attempts to location fusion from multiple heterogeneous, in itself inaccurate, short-range tracking systems was carried out by [Hig03] by the realization called the "Location Stack". Inspired by the OSI network architecture, it defines a layered model of several location processing stages with similar functions as the modules described by the RPA. The bottom layer, i.e. the physical layer in the stack is called *Sensor layer* where the raw data is generated by the hardware performing the timing, temperature, signal-strength, etc. readings of the particular tracking system. This is followed by the *Measurement layer* where the actual localization methods as described in section 2.2.2 are carried out. The following *Fusion layer* merges the provided measurement results into probabilistic representation objects locations irrespective of coordinate systems provided, e.g. Bayesian- and Particle filter techniques [PHU06]. The *Arrangements layer* relates multiple fusion layer objects of different sensors determining the best possible fusion results. Following this stage the *Contextual Fusion layer* combines the resulting location results with other non-location context information, e.g. temperature, light, task schedules etc. Last but not least, the topmost *Activities layer* is application specific and categorizes the resulting context information into semantic states defining an application's interpretation of the world, e.g. a set of rule-based triggers on certain location and task events.

The Location Stack framework is a publicly available Java package implemented as distributed middleware-based service architecture. The respective positioning systems and locating applications communicate via asynchronous XML messages and remote procedure calls and dynamic service discovery capability provided by the middleware.
A lot of different positioning technology types are already supported, e.g. ultra sound, RFID, IrDA, WLAN. So far the Arrangement layer implementation only allows for multi-object proximity tests and object containment within a certain region of a map. The Context and Activity layers are also still a theoretical issue since it in itself represents ongoing research far from plausible realization. Again, the Location Stack focuses on simple sensor fusion producing very good results in localized heterogeneous sensor network installations and represents only another potential part solution of the set out goals of our work.

3.2. Service-oriented Architectures

A service-oriented architecture (SoA) links remotely distributed computational resources on demand for requesting applications, other services or users. According to [KBS04] a SoA can be defined as a paradigm for organizing and utilizing distributed capabilities that may be under the control and ownership of different domains. It provides a uniform means to offer, discover and interact with respective resources in a consistent manner in order to perform and fulfill certain tasks. Thereby these resources are made available independently of prior knowledge of the underlying platform implementation or device type (e.g. mobile end-device or server) [CHT03]. In terms of a software architecture, a SoA defines the use of loosely coupled software services to support the business processes of software users. Especially the powerful service provisioning and discovery mechanisms in distributed environments make SoA successful in many business application areas, the most famous standard being the Open Service Gateway Initiative[5] (OSGi).

The SoA concept has been around for quite some time (e.g. DCOM, CORBA [KBS04]) and supports many service standards e.g. XML, SOAP, WSDL [W3C04]. One of the early SoA concepts linking location services of distributed LPs, database systems and media content can be found in the LASAP project [PPZ99] involving CORBA, an early object-oriented middleware SoA approach. Commercial solutions for LBS in converged mobile and enterprise environments are provided by the OSDE [FL05] and Symphonia [Han05], both of which followed proprietary implementations of the OSGi standard.

3.2.1 The Service Architectures LASAP and Symphonia

The Location-Aware Service and Application Platform (LASAP) was originally developed at the University of Technology of Berlin allowing heterogeneous clients to access heterogeneous resources. Figure 3.3 illustrates the basic concept behind LASAP and more or less any SOA approach. The appropriate middleware platform represented by LASAP enables the deployment, discovery and access to remote and localized LP services, media and other enterprise systems such as user AAA e.g. via an LDAP active directory. It also has a built-in load balancing capability in handling service request. However, the evaluation and handling of the often redundant resources is left to the application. This includes fusion, harmonization, and translation of potential location and content resources, either taken care of by the application itself or needing to be provided by a respective 3rd party service provider.

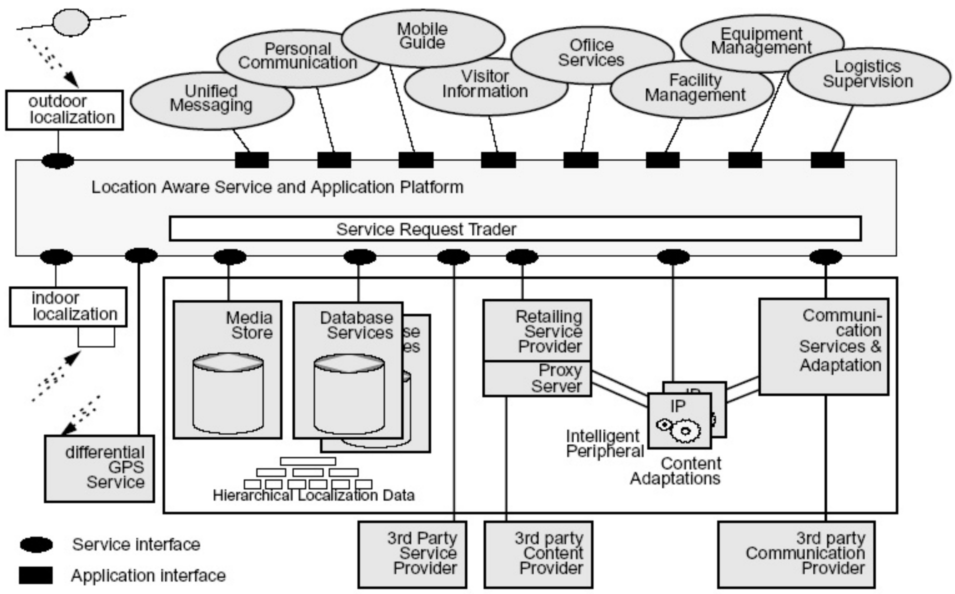


Figure 3.3 The Location-Aware Service and Application Platform (LASAP) [PPZ99]

Symphonia and the more mobile operator focused OSDE are no different with regard to LASAP. Being based on the original OSGi framework Symphonia was originally intended for communication and connectivity solutions within enterprises (e.g. DECT, VoIP over WLAN). The service platform supports various location services provided by respective LPs relying on existing infrastructure components within the enterprise as

[5] OSGi: http://www.osgi.org

well as external mobile operator-based solutions. Furthermore, Symphonia provids basic internal services and libraries for autonomous location information translation by means of “service chaining” to Symphonia-based applications, i.e. location chaining. These properties of forward- and backward-chaining of an architecture originated as methods of planning utilizing means-ends analysis (MEA) [NS63]. By using existing database and location services, the provided translation services can for example map a network MAC address of a mobile device to a layer 2 network switch information which again can be mapped to a Location Identification Number (LIN) identifier required in emergency calls and from there into a street address.

SoAs provide very good service discovery and provisioning capabilities of services implementing the respective OSGi framework reachable via respective network domains. This can be a problem since most companies are reluctant to provide open interfaces to their networks. Furthermore, such frameworks are usually very big and not very lightweight, hence costly to implement. However, SOA concepts are still worth further investigation within the scope of this work.

3.3. Mobile Operator-based and Open Standards

Having introduced various location service provisioning concepts and solutions in chapter 2, there are three dominant standards specifying location service provisioning. For current mobile operator networks the Open Mobile Alliance (OMA) [OM04b] standardization group provides a LP framework providing the functionalities required supporting LBS via standardized interfaces from the operator’s core network as explained in section 2.3.3. The OMA has also another smaller working group dealing with presence and availability wanting to make device status and capabilities context information available to applications, but so far had a fairly slow uptake from industry.

The Parlay group [Par07] has been established earlier than the OMA initiative with the goal of defining open and secured interfaces enabling external 3rd party application providers to access mobile operator network internal resources and services (e.g. mobile operator-based SMS, telephony, location services etc.). This standard was originally designed to be network independent, e.g. support Mobile IP, fixed or IP networks. However, this has resulted in a far too big and complex standard exhibiting only slow development. This is why the former “lighter” and much more mobile operator tailored OMA standard has been founded with the goal of supporting future mobile operator needs more closely.

A more independent and coming out of the geospatial sector approach is represented by the Open GIS consortium [OGISa] which have established the OpenLS initiative specifying the GML language [OGISb] which can be used to describe the output of location sensors in a generic form. The OpenLS is an open LP concept specifying the following core services: a directory service for point of interests; a gateway service for obtaining the location of a mobile device from a network (modelled after the MLP protocol [OM04c], see section 2.3.3); a geo-coding/reverse geo-coding of location names, addresses, etc., into point geometry, e.g. Cartesian coordinates; map presentation as well as routing functionality.

3.3.1 Polos

Since architectures and standards specifying globally distributed LPs for seamless positioning support using heterogeneous positioning systems are not available yet, some work has focused on using and extending readily available location standards, trying to make them more generic, easy-to-use frameworks for enabling location-based billing and privacy. The Polos project [PAM03] has tried to specify a framework for a flexible service provisioning and re-configurability management middleware for 3G systems and beyond. The focus here lies in the initialization and management, user profiling, service discovery, charging and billing of LBS. An overlaying framework (XML) prepares data for interfaces using so called “Context Interpreters”. Compared to the JAIN (Java) standard this approach should be more open and generic. This framework, however, does not consider the heterogeneous distributed positioning system inter-working but takes this as an underlying LP feature “for granted”. Additionally this framework builds upon standards for existing cellular networks and does not support indoor-based positioning networks.

Frameworks for standards extensions of existing standards are fairly limited and rely too much on the underlying functions of the respective supported location service standards. Some well established location service standards descriptions such as the one by the OMA and OpenLS initiative do, however, prove useful when developing an open LP framework which also supports community-based approaches.

3.4. Distributed, Middleware-based and Device-centric Location Service Platforms

We have so far looked at concepts and solutions on processing location information for a homogeneous location service provisioning from multiple distributed location services, means of discovering and providing location services and other information to this cause as well as interface issues for homogenous standardized provisioning of location information. We will now show related work on the described LP approaches from section 2.3.3.

3.4.1 The Device-Centric Location Platform TraX

TraX [KTL06] is a lightweight device-centric LP which focuses end-device-based positioning technologies such as GPS or Galileo, and provides location information independently of respective GIS systems for LBS via the mobile device itself. This LP approach follows a modularized approach supporting different configurations, i.e. it can either provide location information in *direct* or *intermediary*-mode as depicted in Figure 3.4. The direct mode delivers location information directly to requesting LBS providers (e.g. single event localizations), whereas the intermediary-mode uses an optional location provider as buffer for LBS requiring location tracking and for offering more advanced location services. The intermediary allows multiple trusted LBS providers to track a mobile device simultaneously enabling advanced functions for inter-relating the location data of multiple mobile devices (e.g. proximity detection, location triggers, fleet management), or modifying the data for privacy protection (e.g. proposed schemes are anonymization, k-anonymity [KTL06], and an implementation of the Geopriv [IETF] location exchange protocol).

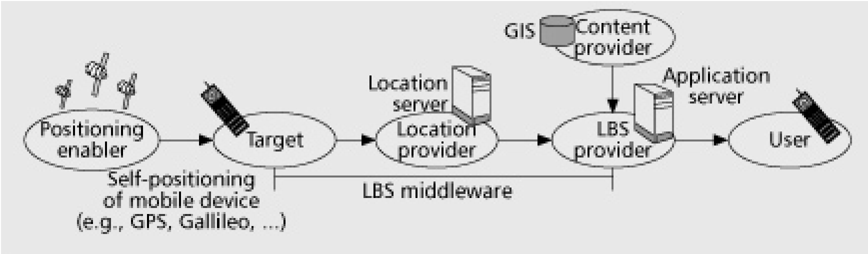


Figure 3.4 The TraX location platform concept [KTL06]

The location service provisioning via the TraX middleware is also defined in three distinct layers. The lowest *positioning layer* deals with the individual positioning technology system returning requested location information immediately (*immediate mode*) or in preset intervals (*continuous mode*), and performing *positioning handover* if one or the other positioning system becomes available. This layer was realized by adapting the Location API provided in J2ME framework [SDN]. The next stage in the framework is the *positioning management layer* which optimizes the location update requests over the air interface. A *positioning deamon* on the client configures the position data exchange by simple *polling*, or more sophisticated means such as *distance* (i.e. dead-reckoning) or *zone*-based location updates [KTL06]. The location data itself can be transferred by means of *piggybacking* i.e. attaching positions to HTTP requests. Finally, the *service layer* should provide necessary location services to LBS via standardized interfaces. The OpenLS standard has been proposed here, extending it with additional sub-services as enabled by the positioning management layer e.g. multi-target location services using zone-based location updates for proximity detection. MLP has been proposed as a transport protocol delivering the location information to respective LBS, although it only so far supports polling-based location updates.

The device-centric LP approach and decoupling of the GIS component limits the amount of positioning systems which can be supported and scalability of TraX. The proposed intelligent location update schemes are thus essential, and could prove useful in LP solutions in general. Although the intermediary is at first considered, it is a required component in efficient location service provisioning relating to multiple-LBS and user tracking support, as well as implementation of some interesting location service privacy concepts, e.g. k-anonymity [KTL06]

3.4.2 Nimbus: A Distributed Middleware-based Location Platform

The Nimbus framework [Rot05] proposes a decentralized location service using distributed location servers. It uses a similar device-centric middleware-based approach as TraX but with more sophisticated concepts on dealing with heterogeneous positioning and GIS systems for homogenous location service provisioning via the mobile device. Figure 3.5 provides a rough overview. The positioning calculation in Nimbus is performed via “positioning drivers” and the fusion as well as provisioning logic of location data for LBS is based on a four stage *Virtual Positioning System* (VPS) framework on the mobile client. The raw location estimates are obtained by the respective LPs in the environment. This information is augmented and converted into a common location format, e.g. from proprietary to global coordinate formats, or a semantic location mapped into physical location, etc. The resulting information is collected from each available location source by the mobile terminal, enriched to provide respective LQoS estimates and converted into a common area location model for selection of the best possible location estimate. The last step in the enriching process ensures that both semantic and physical location information is available for a requesting application to choose from. A semantic resolution algorithm is used for this purpose using LP relationship and hierarchy information, calculating the probability that a user resides within a given service area. Nimbus structures LPs for various usage environments into service domains and hierarchies, defining appropriate relationship (neighbouring nodes in the hierarchy) and association models which are passed between LPs using mobile-server- and inter-server lookup protocols. LP discovery and handover mechanisms are initiated via the local LP in the environment. The discovery is performed in the following ways: the local network or LP itself broadcasts its presence (e.g. LP service beacons, DHCP, SLP, Mobile IP [TW04] etc.), the mobile client broadcasts a service request into the local network (e.g. using DNS), or the mobile client queries the local LP for another LP. As a backup mechanism in Nimbus there is always a cached list of visited and globally available LPs including their relationships and associations stored locally on the mobile client, e.g. globally available root LPs.

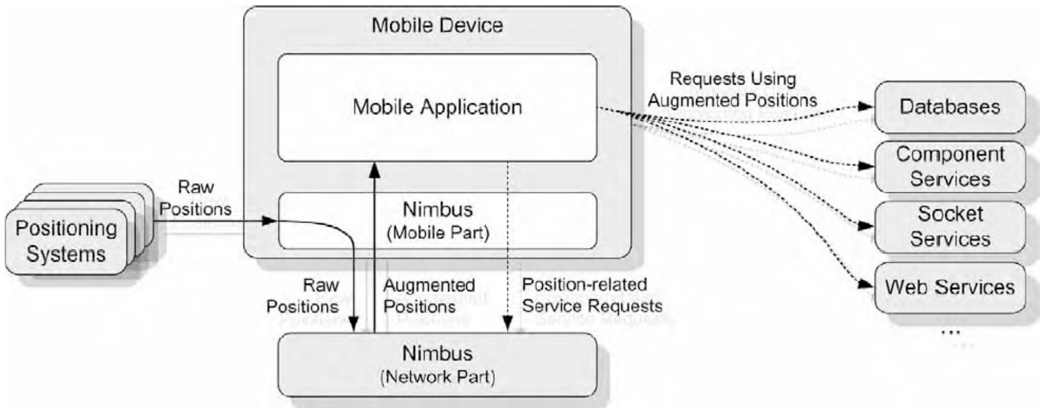


Figure 3.5 The Nimbus location service framework [Rot05]

Although allocating and relying very much on the location information processing and provisioning capabilities on the mobile device Nimbus provides a very promising distributed location service concept of hierarchically modelling LPs for various usage environments. The mapping, harmonization, and selection mechanism of heterogeneous positioning systems compared to the fusion approaches presented in earlier related research projects provides an interesting alternative approach. However the dependency on local network functions and LP discovery mechanisms as well as inter-dependencies between LPs are somewhat of a limiting factor.

3.4.3 A Middleware-Based Approach on Efficient Location Service Signalling

The work developed by Leonhardi [Leo03] also considered a similar approach to the Nimbus and TraX project on providing a globally distributed location service. The main focus here, however, was on developing efficient signaling approaches and protocols, optimizing the location update signaling between remote location sensors (*positioning reporting*) and the respective location service (*positioning requesting*), providing augmented location information to the respective LBS. The sensor or mobile client signal reporting can be optimized on a polling, zone-based, or dead-reckoning basis similar to TraX. The location service on the other hand could optimize the location requests by performing linear-estimations on future locations from history or map-based data hence reduce signaling overhead as needed. Combinations on both positioning reporting and requesting methods have been developed to form coupled-protocol approach adapting to the dynamics of generated location information as well as respective LQoS needs of the LBS. Security and location privacy is provided using a PKI relationship model as well as authentication certificates between sensors/mobile clients, location service, and LBS provider. Means of degrading LQoS, i.e. blurring the location accuracy are also defined on matching security levels between LBS and location service.

The work carried out by Leonhardi provides some interesting solutions on optimizing the location information signaling and provisioning in distributed location service environments and middleware-based LP approaches, which could be regarded as future improvements to our architecture concept.

3.5. Community–based Location Platforms

Current community-based approaches to location service provisioning focus on either collaborative location-based content generation or usage of collaboratively generated location data for location service provisioning. Both will be presented in this section.

3.5.1 Generation of Location-based Content by PILGRIM

PILGRIM is a location broker architecture which uses the location of the user to filter and recommend web information accessed by the user from his mobile device [BB03]. The proposed new middleware-based location broker architecture is depicted in Figure 3.6. The location broker automatically collects information of visited websites and current location data by readily available LP solutions (e.g. GPS) by mobile users, collaboratively cultivating models relating web resources to their spatial usage patterns in a common knowledge database. The recommendation engine evaluates user contributions according to their spatial closeness (e.g. nearest neighbor, Euclidean distance etc.) as well as user interaction and feedback on proposed links (e.g. click rate, retention period on mobile device screen, user action etc.). The recommendations and ranking improve the more data is provided to the recommendation system over time.

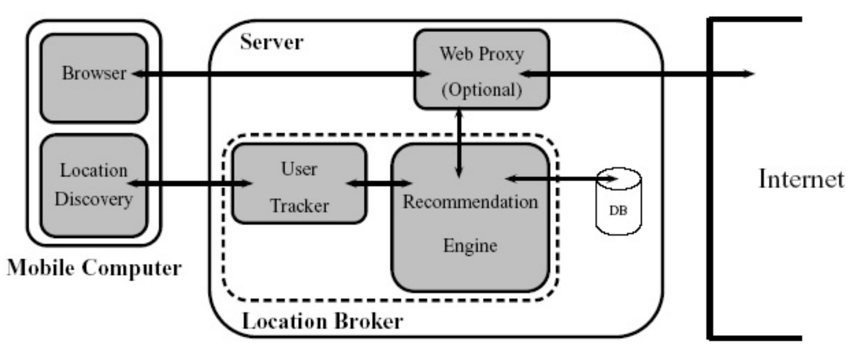


Figure 3.6 The PILGRIM location broker architecture [BB03]

The biggest problem of this architecture is scalability, a lack of input validation and user privacy mechanisms. The growing amount of data provided requires locality-based data structures and a distributed system of local databases possibly available in the respective usage environment. Although very little active user intervention in the validation process of user contributions is wanted and semantic evaluation and validation of user input can be performed automatically, the quality of the contributed content can only be properly validated by the user community itself. Finally, user privacy should be upheld by anonymizing user contributions before they are processed by the location broker.

3.5.2 Operator-Independent Location Service Provisioning

As explained earlier in section 2.4.4, communities have started gathering and geo-referencing of wireless networks in their environment. The concept of using the resulting community database of geo-coded GSM base stations and WLAN hotspot locations for localization independent of service provider or host (i.e. publicly/privately owned) has been investigated in Intel's PlaceLab [LaM05] and the Location Trader (LT) project [DFH04] using a distributed middleware-based LP approach. Both exploit the broadcasting of radio beacons from GSM and 802.11 networks picked up by mobile clients as well as pre-generated beacon reference databases to determine client positions. Users manually contribute their wardriving logs of geo-coded network beacons at once into the community data pool, or as in the case of the LT project, submit their received network beacons automatically during LBS usage (e.g. HTTP piggybacking). In the absence of a GPS device, if a network ID match with the community database was unsuccessful, the user is queried to enter his current location manually thus generating a new entry of the network ID along with this manual location reference in the community database. In order to support the contribution and location lookup process, LT specifies nearest neighbor heuristics [BB03] to be performed on the data pool. Placelab aims to take this concept further and remove the necessity of using GPS or other ground-truth referencing methods once sufficient mapped beacons are available in the location pool. Furthermore, a different system heuristic called *Beacon-Print* [HCL05] was specified in order to "learn" about (common) places users frequently go to. Privacy in both cases is enabled by the user enabling or disabling the passive beacon monitoring by the client in the environment, hence his contributions to the community. Placelab also suggests a partial downloading of some portions of the community database to the client following the device-centric LP approach. Both see anonymization of the war-drive logs and contributions to the community data pool are seen as essential.

The LT and Placelab concepts are also only part solutions in the direction of our global location service approach, lacking essential features on LP discovery and handover, scalability, localization capabilities, etc. The limitation of both approaches is to maximize the coverage and availability of the LP, but considering positioning accuracy only secondarily. Whereas the LT project has focused more on the community development aspects, Placelab has looked more on the system heuristics on processing the contributed data. Experiences from both works will help develop the approach followed by our work.

3.6. Summary Discussion and Conclusion

In order to draw a final conclusion, we will summarize the analysis of the presented concepts and architectures in Table 3.1. All presented architecture, frameworks and LPs support heterogeneous positioning systems in some means or other. A separation and simultaneous support of various LP types and sensors is most commonly found in the architecture as well as framework examples. A decoupling by means of interfaces and adaptors for mapping corresponding location information into a homogenous location format using an appropriate location model for further processing is found in the RPA, Location Stack and Nimbus projects as well as in the work by Leonhardi. This is an essential feature and perquisite for homogeneous best possible as well as global location service provisioning independent of underlying technology and service implementation. Polos is restricted to standardized interfaces and 3G/4G platforms of mobile operators. The SoA-based projects LASAP, OSDE, and Symphonia merely provide support if respective solutions provide services implementing the corresponding SoA framework. Other approaches specify no concrete augmentation of the heterogeneous location data and/or assume a homogeneous form for further processing, thus focusing more on the parallel signaling of each in particular (e.g. PILGRIM, LT, Placelab, TraX).

Following the homogenization/mapping process, we have specified post-processing on the multiple and varying location data sources (in terms of LQoS) in terms of intelligent selection, fusion and/or estimation functions, in order to ensure a relatively stable level of LQoS for requesting LBS. All of these methods are supported by the RPA and Location Stack framework, whereas Leonhardi and Nimbus revert to only providing intelligent selection mechanisms. The question of fusion necessity at the global location service level is brought up again as Location Stack only applies this to local, closed environments using dedicated (in itself inaccurate sensors) and appropriate GIS data, both being handled at the LP level. Due to the common dependence between location data and GIS content of local LPs, an intelligent selection, matching and provisioning of appropriate LQoS at the global location level, seems to be the right choice until appropriate, less complex fusion solutions for both types of location information are feasible. Both the Nimbus and the Nexus project [DHN04] see this also as future work.

The distribution of location information processing across client and infrastructure/server-based components is a critical issue. In order to ensure a broad end-device support as little processing functionality as possible should reside on the mobile client (contrary to the Nimbus approach). The local environment location information processing and provisioning is best kept with the local LP as well as global location service provisioning centralized for LBS support. The architecture should, however, be flexible, lightweight and modular enabling an adaptation of these processes best suited for particular LBS application, environment, infrastructure and device purposes i.e. allocating more complex functions onto the mobile client as in Nimbus. Using a distributed middleware-based LP approach and a centralized global location service requires an optimization of the location data transmission between client and local LP, as well as between local LP and global location service as described by Leonhardi and TraX, thus should be supported in a global location service architecture framework, but won't be covered by this work.

Based on the previous points on mapping and intelligent selection on providing a homogeneous location service from multiple sources, a common location model is needed whereto each source can be potentially mapped (e.g. RPA, Location Stack and Nimbus). Mapping services could be *additionally* provided from external sources to enable format flexibility for LBS (e.g. internal translation services offered by Symphonia, translation from geographic/physical to semantic locations as pursued by Nimbus). Nevertheless, the provisioning and interpretation of semantic location information is still not suitable for practical applications and a very big research state on its own. Hence the focus should be to provide location information in physical coordinate form ensuring broad compatibility to many GIS systems, using common location standards for location service provisioning, e.g. OpenLS, adapting the much developed OMA standards [OM04c] etc. The provisioning of the common global location service and respective post-processing operations should be independent of LP and location technology sources and invisible to LBS, but available location information and corresponding LQoS should be presented via the common location model. Therefore appropriate negotiation and adaptation functions need to be available for LQoS-aware LBS, whereas legacy LBS should be provided set LQoS levels as needed, e.g. best effort, maximum accuracy, maximum availability, etc. Currently all presented approaches leave the mapping, selection and negotiation up to the respective LBS increasing in implementation effort and costs for the application provider. Hence seamless location information in dynamic and heterogeneous LP environments requires these additional functions.

SoA solutions such as Symphonia, LASAP and OSDE provide very powerful location service discovery mechanism in distributed domains, but are costly to implement and support. Furthermore, depending on knowledge and access to the local network environment such as in Nimbus is hard to achieve in heterogeneous network domains. The location service discovery needs to be independent and with little impact on the respective local network domain (e.g. broadcasting mechanisms), or at least use the local information passively by means of monitoring. Instead of relying on SoA approaches a more lightweight and flexible approach would be better, e.g. a peer-to-peer-based solution. Last but not least, due to the seamless location service provisioning requirement, appropriate handover methods and triggers need to be supported in addition to location service discovery mechanisms. Modeling organizational structures of LPs with their service areas as done by Nimbus and Leonhardi is essential.

The privacy and access control of user location information on such location platforms is very crucial. The passive monitoring and transmission for localization or training purposes by the client which can be transparently controlled on the device dramatically increases location privacy awareness for the user. The ACL management on the location information should also be transparent to the user [ZD05b] using certificate authorities, white-/black lists, and can be stored centralized with the global location service user profile (i.e. RPA and Nimbus). The presented decentralized middleware-based approaches combined with ACL and certificate mechanisms provide a basic and sound solution. Full scale AAA mechanism relaying on existing IT infrastructures are best integrated following the presented SoA approaches. Extended privacy mechanism (e.g. blurring effects introduced by Leonhardi) can be seen as future improvements.

The initialization and self-optimization (learning) is an important part and new future feature which has to be considered in distributed location service provisioning combined with community models as well as clients as sensors in the environment approaches followed by this work. Hereby, the local LP responsible for environment the user client currently resides in should perform the initialization and self-optimization. The global location service should merely support this process for the respective local LP discovery and signaling of client location data. All of the presented architectures and frameworks provide location service solutions without feedback, considering the availability of appropriate, cultivated reference databases as a perquisite. The RPA is the only solution considering history location information results to modify current location estimates. The works by LT, Placelab and PILGRIM consider the initialization and generation of location information and content within the location service provisioning process, supporting concepts developed in this thesis. The respective manual and automated feedback/learning heuristics need to be developed as well.

Altogether, a lot of interesting and good approaches to the specified problems already exist. Most of the aspects either focus more on the location service provisioning or processing. But there is a big gab especially in the initialization and self-optimization capabilities, often taken for granted. The community-related concepts have looked into this problem in some ways. Thus it is worth pursuing the goal on combining these efforts and using existing approaches where needed, forming a new distributed location service architecture approach.

Location service architecture/ framework	**Heterogen. location service post-processing (mapping/ selection/ fusion)**	**Homogenous location service provisioning/ negotiation/ adaptation**	**Location service discovery/ handover**	**Scalability/ signaling optimization**	**Security/ privacy on location**	**Learning (initialization/ self-optimization)**
RPA	√ / √ / √	√ / – / –	–	–	(√)[f] / √	– / (√)[h]
Location Stack	√ / √ / √	√ / – / –	(√)[c] / –	–	–	–
LASAP, OSDE	–	–	√ / –	(√)[e] / –	(√)[f] / –	–
Symphonia	√ / – / –	– / – / √	√ / –	(√)[e] / –	(√)[f] / –	–
Polos	(√)[a] / – / –	√ / – / –	–	–	(√)[f] / √	–
TraX	√ / – / –	–	–	– / √	– / √	–
Nimbus	√ / √ / –	√ / – / –	√ / (√)[d]	√ / –	√ / –	(√)[i] / –
PILGRIM	(√ / √ / √)[b]	– / – / (√)[b]	–	–	– / (√)[g]	(√ / √)[b]
LT	–	–	–	–	– / (√)[g]	√ / √
Placelab	–	–	–	–	– / (√)[g]	√ / √
Leonhardi	√ / √ / –	√ / – / –	–	√ / √	√ / √	–
Covered by this work	√ / √ / (√)[k]	√ / √ / √	√ / √	√ / –	(√)[k] / (√)[g]	√ / √

[a] Mapping of location information into a generic format
[b] Only relating to content using a common location data format not processing location data itself
[c] Rudimentary Java RMI methods from the J2ME framework only suitable for local domains
[d] Specifies inter-LP protocol but no seamless inter-working
[e] The OSGi and other SoA frameworks have built in load balancing and resource allocation
[f] Uses readily available network-based AAA mechanisms, e.g. certificate authorities, SSL, PKI, etc.
[g] Privacy control on device and/or anonymization of generated user data before sharing/collaboration
[h] Modification of location information based on history locations and mobility prediction
[i] Importation of GIS-data from available systems for geographic and semantic location mapping
[k] A possible concept will be supported but a concrete fusion solution left subject to future work

Table 3.1 Comparison of different related work approaches

Chapter 4
Modeling a New Location Service Architecture

The analysis of the previous chapters has shown many concepts of providing a location service from a distributed network of LPs, fusion as well as modeling of heterogeneous location sources. None of these concepts consider the capturing and generating the relevant location information from the actual usage environment, hence seeing this (if at all) and readily available interfaces to existing systems as a perquisite. In this thesis, the challenge is to develop a community model with potential user roles and degrees of involvement on generating location information, both manually and automatically, for a common location service. The resulting new architecture will have to provide appropriate interfaces and functions to support this new concept and readily available LP solutions.

This chapter will give an overview on the new location service architecture LACBA (Location-Aware Community-Based Architecture), which fulfills the set-out requirements as specified by this work. As already mentioned multiple times, the focus of LACBA is to support community-based location information capturing and generation from the respective usage environment in addition to readily available LPs, offering a generic, seamless location service for LBS. We will explain the abstract concepts and system model which will provide the basis for further detailed architecture descriptions. We will describe LACBAs core processes and relevant data models as processed by respective system components. A more detailed specification on how location information is collaboratively generated from the wireless network environment using LACBAs flexible LES framework will be presented in chapter 6. The control and adaptor elements responsible for the user-, LP- and LBS-session management as well as signaling concept will be specified. A detailed specification of the LACBA location service discovery mechanism will follow in chapter 7.

This chapter is structured as follows. We will first introduce the basic concepts of LACBA as set out in the architecture requirements in chapter 2, showing the different actors as well as their respective interfaces in the LACBA ecosystem. Then we will go into the modeling of our service architecture. First of all, however, the layer architecture and data model of information processed by LACBA will be shown. Describing the LACBA components and their major functions, we will then introduce the control and adaptor entities, the signaling concept between them, and show their functionality using respective examples.

4.1. Basic Concepts of LACBA

The main goal of LACBA is to provide a generic location service from multiple independent LP solutions for location-aware applications and services. Its fundamental principle can be best explained by an example as seen in Figure 4.1. The client device is used as a sensor in the environment scanning for available wireless networks (e.g. GSM/UMTS Cell IDs, WLAN MAC addresses/ESSIDs etc), identification tags (e.g. RFIDs), or using dedicated positioning information from readily available GPS devices. This information is transmitted to a central control entity (1) which then performs a lookup search (2) using this discovered information thus determining potential LPs being able to serve location information from this discovered data. The architecture can support an arbitrary number of LPs (LES/GIS pairs) or independent GIS solutions (e.g. Google Maps). Once the target LP has been determined and user access rights have been verified, a LP session (3) is established and the sensor data stream forwarded accordingly. The target LP generates location information as requested (4) by the central control entity. The central control entity in turn provides a generic location service for requesting (5) LBS.

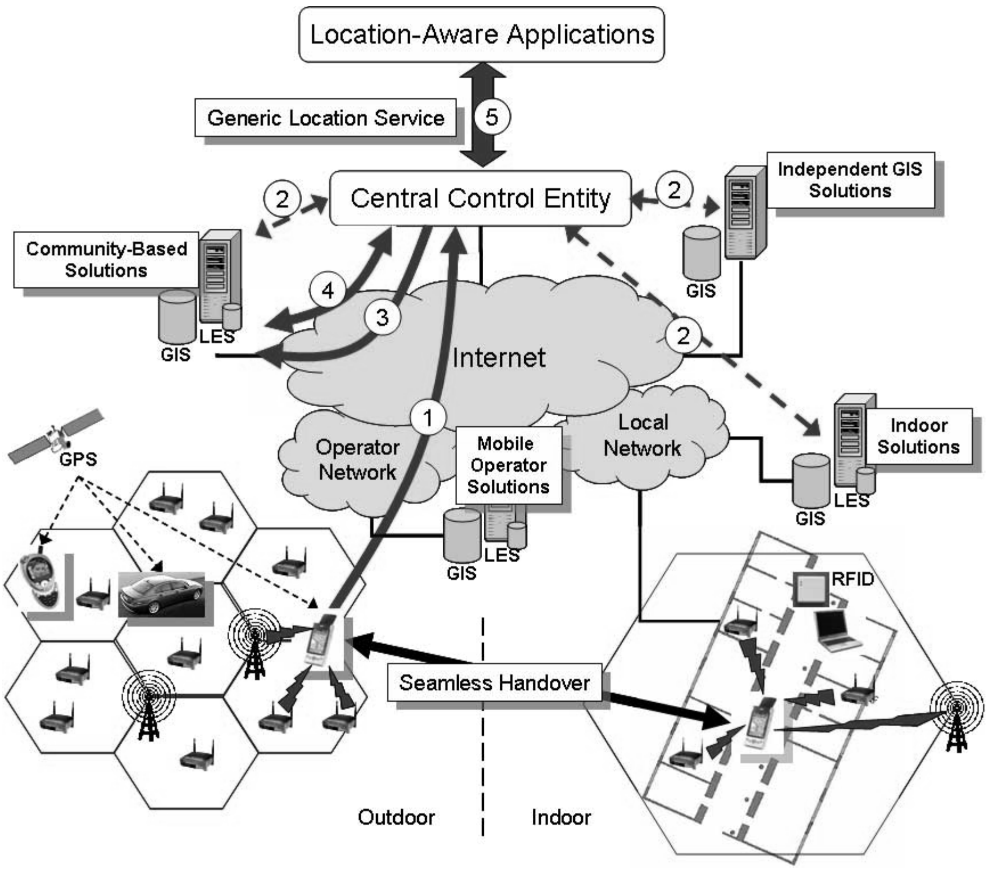


Figure 4.1 The location service provided by LACBA

Since our novel location service should be seamless, location information needs to be provided from multiple LPs simultaneously. The central control entity manages single as well as multiple, parallel LP sessions (2)(3)(4), processing and providing location information to the requesting LBS via designated control and adaptation entities as explained in the next sections. A centralized, server-based location service for LBS is an essential part of the LACBA concept and will be referred to as the **location core server** (LCS) from now on.

4.1.1 The LACBA Use Case Model

Before discussing the LACBA location service concept in detail, it is necessary to introduce the LACBA use case model and its ecosystem of generating, providing and using location information. The LACBA community can be distinguished into three major use case groups (Figure 4.2): the service provider, the location platform provider, and the user community. A service provider uses the LACBA location service via the LACBA location service API providing LBS for the LACBA user community. (Note: The relation of community users using the LBS provided by the service provider has been omitted in Figure 4.2 for simplicity reasons).

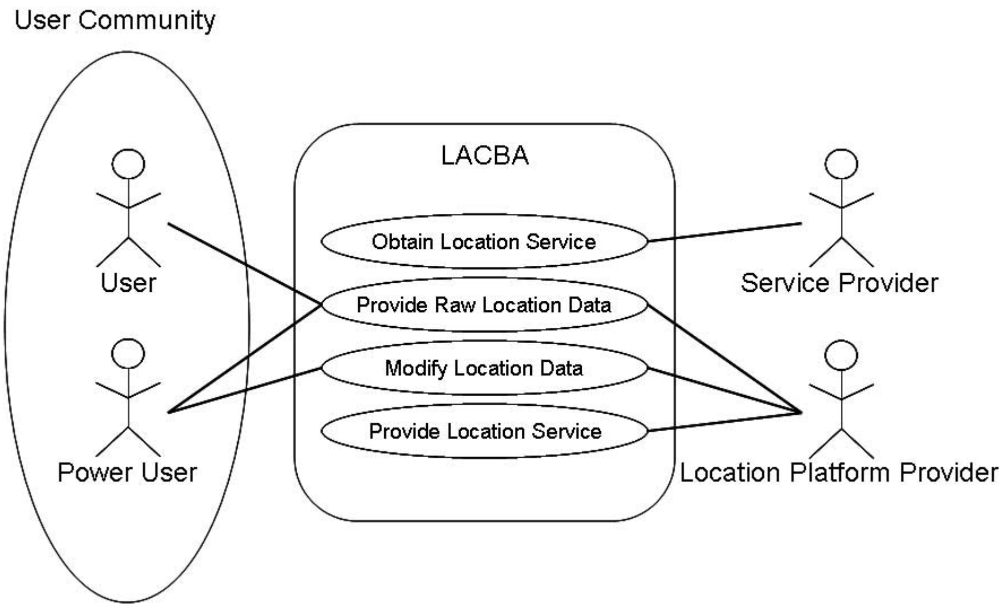


Figure 4.2 The LACBA use case model

Community Users

The user community in LACBA not only uses the LBS offered by the service provider, but also provides autonomously discovered or manually entered environment sensor data via their LACBA enabled mobile device to enable the LACBA location service. Hence, the task of all community users is to provide sensor and identification data to discover the relevant location platform provider in their current usage environment. A distinction can be made between normal users and power users within the community. This sub-group of power users has special administration/modification rights (e.g. service engineers). They can have a certain "trusted" status within the user community by using and uploading/correcting data to the relevant LPs in their usage area more frequently. They usually also have more knowledge of the usage area they are collecting data from compared to normal users. This is especially important during initialization or training of positioning systems. Here, power users can perform "structured" and/or guided initialization in order to obtain the "best possible" positioning results within the designated service area within the usage environment (see section 2.2.3).

Location Platform Providers

The location platform provider hosts a LP providing a location service for a designated usage environment having a well defined service area. He can use the sensor data provided by the user community to generate and provide location information via his location service. In providing a location service to LACBA he not only has to provide appropriate positioning estimate based on user sensor and/or other identification mechanism, but possibly also some “quality” estimates according to the LACBA location model. Obtaining such information from a mobile operator [OM04b] or WLAN indoor-based [Eka06] LP solution is difficult to obtain since it is not specified in the respective standard or not seen as required. Thus LACBA can determine possible location model QoS parameters itself e.g. availability, initialization time or location update intervals (see section 4.4.1).

A community-based location-platform provider is certainly also capable of modifying and providing location data for positioning itself i.e. mobile operators maintain detailed lists and maps of their base station positions for maintenance purposes; internet communities [Wig06] maintain databases of WLAN hotspots and their locations etc. In the latter case, a user community could also be a location platform provider, but won’t be considered in this work.

The location service and location information generation from user sensor data features are logically provided by or via LACBA, but are distributed among different components with the LACBA framework. This allows us to define separated modeling approaches as we will show in the following sections.

4.1.2 The LACBA Location Service

The LCS provides the generic location service in LACBA. As pointed out in 2.5, the location architecture has to cope with multiple, heterogeneous LPs and their respective positioning technologies. Furthermore, it has to provide a generic location service possibly involving adaptation, fusion, and/or estimation mechanisms. In order to perform mechanisms such as fusion at this level a common location model as we will present in section 4.5.1 is required.

Realizing these requirements within LACBA requires a separation of the **application** from the LACBA **location service plane**. The resulting interface has to model the LBS requirements to the currently available location service capabilities in respect to the user’s privacy and security settings.

Moreover, the inter-working and accessing the location service from multiple heterogeneous LPs requires a separation of the LACBA location service from the underlying **location platform plane**. Hence the LACBA location service control has to be independent of the underlying LP location services. This control process involves the user access management and signaling of user sensor data, LP discovery and inter-working as well as location-information adaptation to the LACBA location model. This decoupling in LACBA is realized by the **adaptation plane**. The components of the adaptation plane (from now on described as **adaptors**) translate the control processes to the respective LP service descriptions.

In order to meet the requested LQoS as required by the LBS, the location information from the adaptation plane can undergo additional fusion and estimation processing. A **post-processing plane** with its respective fusion and estimation modules provides these functions if the location information from the adaptation plane fulfills the LACBA location model requirements fully (see section 4.5.1). In short, the fusion module fuses input from two or more LPs, and the estimation module tries to predict future position estimates from history location data from individual LP sources. The location information results from the post processing plane are additionally made available for evaluation along with the un-processed location information from the adaptation plane to the LACBA location service.

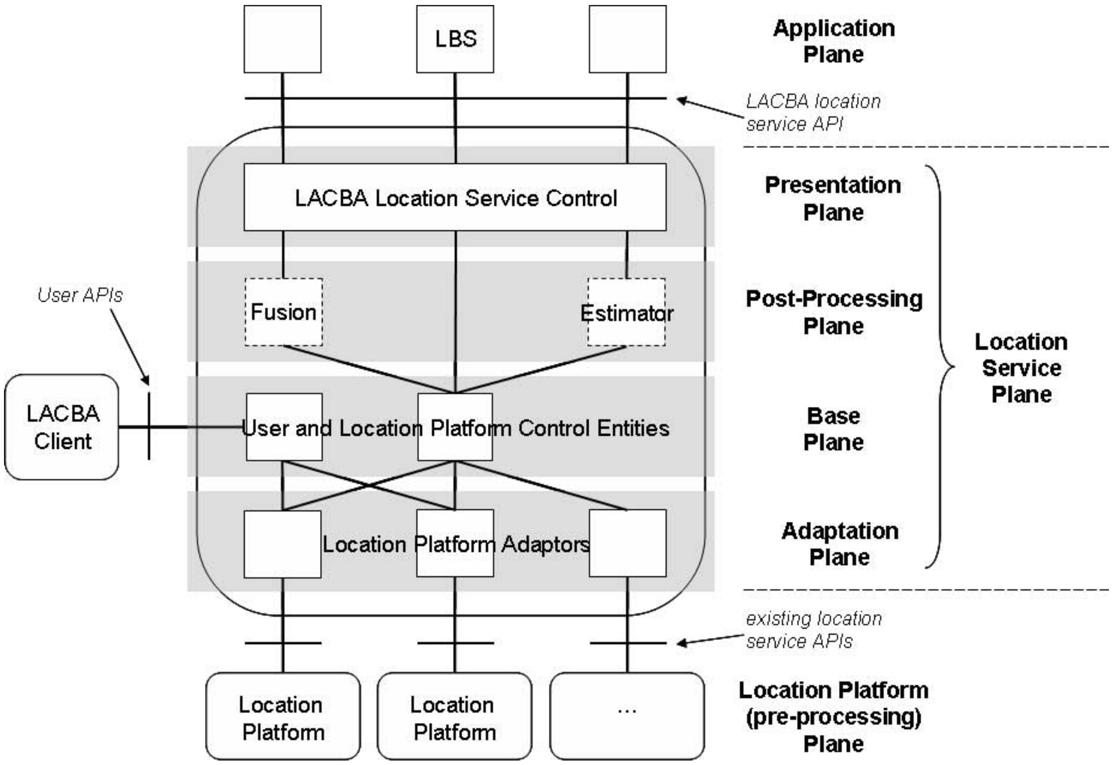


Figure 4.3 Basic principle of the LACBA location service

The user-, LACBA location service- and LP control is performed by **control entities** at certain location service planes. The LACBA location service control i.e. LBS session handling, matching and (re-)negotiation of LQoS, is performed by the *Location Service Control* entity in the **presentation plane**. Performing LP discovery, inter-working and session management is carried out be the *Location Platform Control* in the **base plane**. The respective *User Control* manages the LACBA client sessions at the LCS and controls access to, as well as privacy settings on the user location information.

4.1.3 Generating and Adapting Location Information from the Environment

As already mentioned, the functionality of LACBA is distributed among various components. The location service is provided by the LCS and the generation as well as adaptation of location information from the usage environment for providing a location service in a designated service area is carried out by the respective LP. In particular, the LES component of the LP performs this task. The accompanying GIS component provides the "value added" location content and/or services. Hence we will focus on describing the LES principle when describing the unique location information generation and adaptation features of LACBA.

Henceforth, the unique properties of a LACBA LP (i.e. LES part) can be summarized as follows (see comparison in section 2.5):

- The framework has to be flexible and generic in that it can support various LP solutions ranging from indoor- to the most demanding and complex outdoor community-based systems.
- It has to support various automated as well as manual initialization and feedback on location reference data
- Dynamic adaptation of generated location reference data during the lifetime of a LP (i.e. learning changes in the wireless environment)
- Processing of multiple client initialization instances and reduction of sensor data volume (i.e. collaborative location reference data generation)
- It is to cope with potentially huge spreads in measured sensor data for initialization (i.e. differences in hardware of supported positioning technology)
- Robustness on faulty sensor measurements (i.e. integrity checking)
- Establishment of a stable LES system "without administration"
- A repository of suitable initialization and positioning models for specific environment and positioning technology applications.
- High availability and reliability of offered location service

One major perquisite in realizing these requirements is to separate the *LES Location Service Control* and the sensor data processing from the client(s). This is especially relevant during initialization processes not interfering with the positioning processes, and is realized by separating the location service provisioning plane from the data-processing plane within the LES framework. During positioning it is essential that the current user sensor data is being processed when a positioning request via the LES location service occurs. During initialization multiple user sensor data streams have to be possibly processed in parallel. Therefore the user control has to be separated from the initialization and positioning processes as well, but both processes need to be fed by the same user sensor data. This is achieved by separating the LACBA client control and data acquisition plane from the data processing plane. The initialization and positioning processes are treated both within the data processing plane but have to run independently in order to support different initialization strategies as explained in chapter 5 later. The resulting LACBA LES framework is represented in Figure 4.4.

The *LES Location Service-* and *LACBA Client Control* are also realized by respective control entities within the corresponding LES planes. The **data acquisition plane** provides the LACBA client interface for client session management and access control. Before being able to carry out initialization or positioning from multiple LACBA client sensor streams, the data has to be prepared and integrity checked accordingly. This is the first processing stage in the **data-processing plane** of the LES. From that stage the initialization- (for reference data building) and positioning modules (using the reference data) are fed. The positioning can be followed by additional fusion mechanisms for fusing multiple homogenous or heterogeneous sensor data [Hig03] positioning results. Furthermore, manual user or automated feedback information on positioning results can be fed back to the initialization modules for updating or correcting reference data.

Finally the LES **presentation plane** provides the location service via the LES API to the LACBA location service from the data processing plane. The detailed LACBA LES framework and suitable initialization/positioning models for collaborative community-based applications will be discussed in chapter 6.

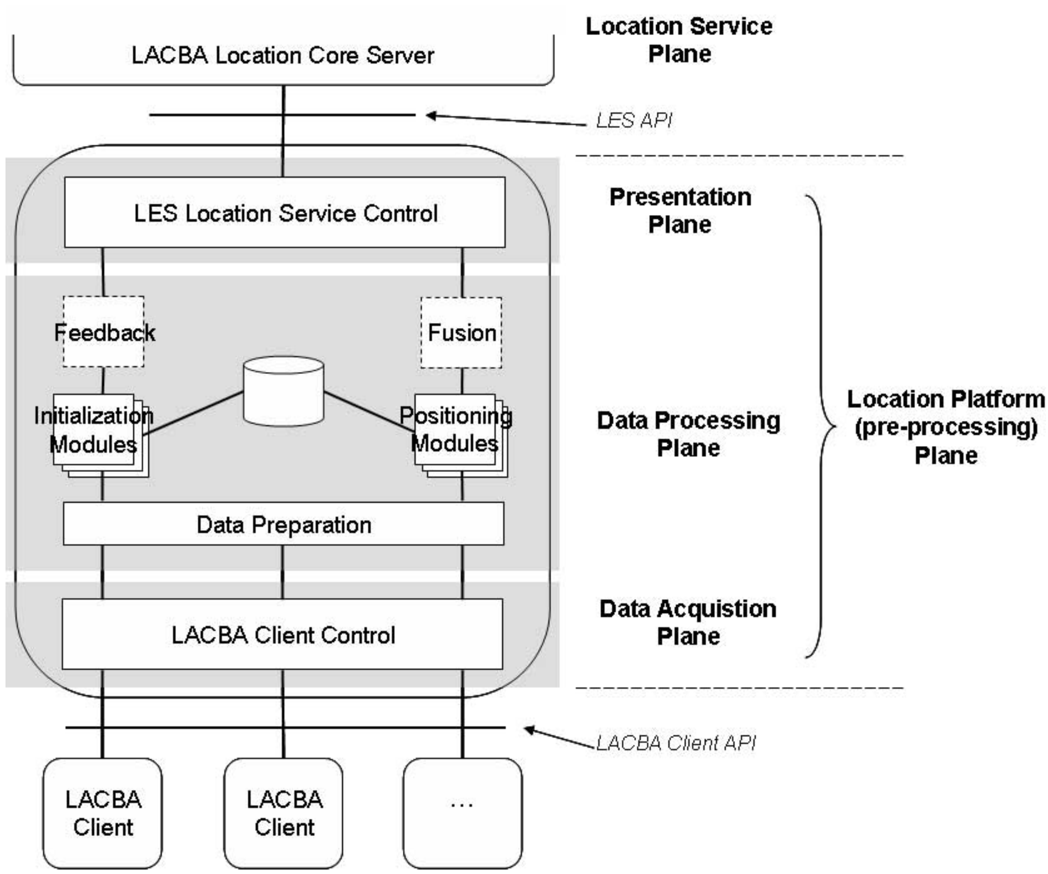


Figure 4.4 The basic LES principle in the LACBA framework

4.1.4 Intelligence Distribution

The centralized location service provisioning from and the control of multiple heterogeneous LPs has been already defined as a fundamental feature of LACBA in section 2.5. LACBA aims to keep the intelligence centralized and server-based with the respective coordination and control entity (the LCS), as well as with the various LP solutions (e.g. mobile-operator-based solutions [OMA]). Hereby the intelligence separation between the homogenous location service provisioning and location information generation (i.e. initialization processes) as well as provisioning from various usage environments are distributed respectively. A Location platform provider is assumed to provide the best possible location information and GIS content since their LP supports the technologies and uses the positioning methods best suited for the local usage environment. Furthermore, as discussed in chapter 2, local communities only have the potential to provide valuable local and up-to-date information form the usage environment.

Purely end-device-based approaches lack the potential required location information processing performance or required seamless LP handover and inter-working capabilities. Furthermore, such an approach does not fulfill the requirement of supporting heterogeneous and legacy end-devices. This last point is especially critical when trying to build communities around certain LPs and reaching critical user masses fast. Hence, supporting and reaching many potential clients is crucial.

In LACBA a minimum of end-device intelligence is necessary in using it as a sensor in the environment, transmitting the discovered data to a central distribution point for further processing. Intelligent location update signaling [KTL06] aids in reducing signaling overhead and incurred communication cost. This is however seen as a minor issue in the age of flat-rate internet connectivity and ever falling mobile communication prices compared to LP content and service provisioning costs.

Contrary to an end-device-based approach the centralized location service principle in LACBA allows for LBS session continuity whilst dynamically handling multiple LP location service sessions in case of seamless LBS scenarios. Here the LCS can also provide caching and optimize the LP signaling at much lower cost and higher reliability.

Regarding privacy and access control to user location information, the centralized approach manages user profiles and keeps privacy settings for LBS and other user access control. Although this imposes privacy issues as pointed out in [ZD05b] where user control at the end-device is clearly beneficial the user should be directly possible to access and manage the privacy settings via appropriate interfaces. The LCS on the other hand manages the user access control authentication at various LPs for LBS sessions automatically.

Picking up on the required modularity and flexibility of the architecture it should be possible to allocate the functions of the LCS onto the mobile device as well provided it has sufficient computational performance (i.e. similar to end device focus in Nimbus [Rot05] but having the flexibility sought by TraX [KTL06]). Such a configuration clearly favors LBS configurations running on the mobile device itself (e.g. navigation) and result in a peer-to-peer based interaction with the distributed LPs. However, for the scope of this work we will focus on developing the centralized location service provisioning approach and keep the device centric variation and respective architecture implications to future work.

4.2. The LACBA Core Processes

The LACBA model has been presented in horizontal control planes. However, the core processes of location information generation, provisioning and self-optimization in LACBA are distributed vertically across several LACBA entities. In order to understand how the individual data flows are processed between its components the LACBA framework can be viewed as a layer model having various processing steps similar to [KTL06][Hig03][Pfe05].

4.2.1 The LACBA Layer Model

The LACBA layer model distinguished four layers building on the respective sensor technologies used. These layers can be mapped to LACBA components and planes as described in the previous sections. Figure 4.5 shows the LACBA layer model, respective component mapping and the three core processes: 1) the learning, initialization, and generation of location information from readily available network and dedicated positioning sources 2) positioning of users, fusion and adaptivity of location information for location-aware applications, and 3) feedback and optimization of location reference data.

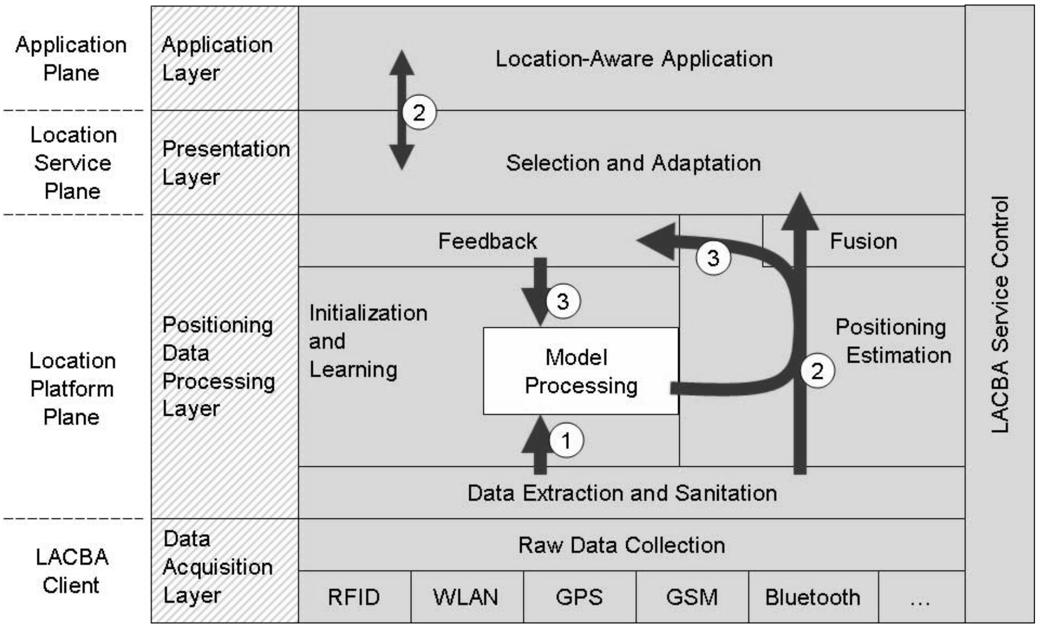


Figure 4.5 LACBA framework and processing layers

Although the following sections describe these processes from a universal LACBA point-of-view, initialization, positioning, and learning processes can be seen as different states of the LACBA system and inter-working of various LACBA components. The LES LP part (i.e. location platform plane) is mainly concerned with the environment specific initialization, positioning and learning processes and respective LES states as discussed in section 2.5. Chapter 6 will cover these issues more deeply.

The **Data Acquisition Layer** involves the data capturing of various sensor interfaces, transmission preparation and signaling by the LACBA client on the mobile device to the LP LES component for further processing. The capturing by the client is done via drivers at suitable intervals affixing timestamps, hardware and network specific identification means during data packet compilation. The LES uses this data for LES session handling and preparation of subsequent data processing mechanisms.

The **Positioning and Data Processing Layer** is entirely represented by the LP. As mentioned in section 4.1.3, the perquisite for initialization and positioning from heterogeneous data streams is an integrity check, possible normalization, verifying and filtering according to specified rule sets, e.g. correct data formats, parameter ranges, malicious code, etc. This pre-processing stage feeds both initialization (1) and positioning (2) processes.

4.2.2 Initialization and Location Modeling

During initialization (1), the pre-processed data stream is fed into the pre-selected and registered initialization modules for location reference data generation. These modules have been selected the most suitable for the usage environment. Depending on the implemented positioning methods, the initialization modules can perform geo-coding of wireless base stations or RSS fingerprints of observed wireless networks, and store these in a reference database for positioning purposes. Furthermore, the database can hold additional information on "how much sensor data" was used to generate this data set or how many users have verified/seen this base station location. This allows for evaluating the quality of location reference data generated.

4.2.3 Positioning and Adaptation

A LACBA client is assumed to connect to at least one LP i.e. LES entity during positioning (2). Upon a positioning request of a LBS, the positioning estimation methods perform a positioning calculation using the location data stored in the reference database and the current sensor data available from the to-be positioned LACBA client.

The obtained location information can be further processed by fusion processes. At the LES plane, this is typically the case if multiple heterogeneous, and in itself inaccurate sensor technologies are used in the user environment [Hig03]. A perquisite for fusion is that all sensor technologies and positioning methods provide a common location format and accuracy interpretation, e.g. by means of probability-based measures. This is however is seen as optional and should be taken care off by the respective LES entity as discussed in section 2.5.

The calculated location information along with respective confidence measures, i.e. inaccuracy and probability confidence of the measured location information, is then provided via the LES location service. The confidence measures can be obtained from the initialization process and reference measurement used (see section 2.2.3). A later section in this chapter will pick up on this dealing with the LACBA location and QoS model.

The **presentation layer** processes the location information from tracked entities of single as well as multiple LES location services in the LACBA location service plane as described in section 4.1.2. Here it is ensured that the location aware application/LBS obtain their requested location information quality as specified by the LACBA location model (e.g. LQoS parameters such as location accuracy, precision, update interval, availability). If the required LQoS parameters cannot be met, optional fusion and location smoothing by means of interpolation as well as estimation can try and combine multiple LES sources in order to obtain the required LQoS [Hig03][Ng03]. Lastly, if the required LQoS still cannot be met, the task of the presentation layer will be to signal this to the LQoS-aware application appropriately to adapt its QoS needs or terminate the service.

Unlike fusion at the positioning and data processing layer fusion at the presentation layer is a special case and is less likely to occur frequently. Fusion of heterogeneous LPs requires both location information (LES data) and usually also location content (GIS data) adaptation, i.e. they need to be in the same location coordinate form and data format. Fusion of an outdoor and indoor-based LP rarely makes sense since the latter are assumed to provide much more accurate building map data with customized local positioning technologies. As mentioned in section 4.1.2, LACBA concept will support fusion capabilities to some extend at the LACBA location service plane.

The LACBA location model aims at signaling changes to LQoS registered with the LACBA location service and adaptation information (i.e. on how to handle the new location content and information) to the LQoS-aware application. Location content adaptation eventually is performed and taken care of at the **application layer** level (i.e. by the application itself) and not of concern by the LACBA location service.

4.2.4 Feedback and Optimization

The feedback and optimization process (3) represents the self-optimizing capability of LACBA. It aims at keeping the LES location reference databases and location information generated from the environment up-to-date. This again is only possible through the collaborative efforts of all LACBA clients. During the positioning process of a LACBA client it is possible to register potential positioning miss-matches e.g. missing WLAN access points within a geographic cluster of WLAN hotspots [Ipp06] or along a common and busy traveling route. Frequent miss-matches of LACBA client entities exceeding a set threshold can then be interpreted as degrading of positioning quality of a LES entity and determine obsolete data entries within the location reference database. The reverse is also possible in detecting new networks in the environment. The newly discovered network could be considered after a defined threshold of users having detected the same network over a defined period of time. Aging and flagging of potentially obsolete data-entries are means of indicating potential re-training efforts needing to be undertaken to appropriate LES entities, if a pre-determined threshold has been met.

Apart from these automated feedback and learning mechanisms manual feedback by users can be supported on the location platform plane. The overall aim is to minimize manual contributions and inter-actions by community users, but on the other hand it could provide valuable corrective input if the system is lacking information and could not determine the user’s current position. Means of manual corrective input could be the user manually entering his current physical address or the location of a WLAN access point [ZD06]. In addition to the manual input, the system could add the current wireless fingerprint or GUI map coordinates via a touch-screen enabled device [LaM05].

4.2.5 The LACBA Modelling Views

Service architectures are often very complex due to the features and functions required, being comprised of many distributed sub-systems. There are several abstract modeling techniques available in order to handle this complexity. In general, a respective system model thus includes several modeling views in order to describe each aspect of a service architecture (e.g. business model, data model, use case model, etc.). The work by [Kel02] provides a very useful modeling approach combining the benefits of both RM-ODP [X.901] and common object-oriented design (OOD) principles found in software engineering [BD00]. The architecture described and modeled in [Kel02] has similar requirements as LACBA on providing an independent service control for distributed and heterogeneous resource environments.

The standardized *Open Distributed Processing* reference model (RM-ODP) enables the development of services in distributed, heterogeneous resource environments but lacks modeling views to describe actual system behavior. On the other hand, object-oriented software engineering does provide several system behavior modeling views (e.g. *Object*, *Functional* and *Dynamic Model* [RBP91][BD00]). Coming from [Kel02], the combined modeling views can be summarized in the following way giving relations to the respective original modeling views will be given in each case:

- The **business model view** shows the interfaces to various external and/or internal applications as well as entities inter-acting with the architecture. This is synonymous to the UML use case modeling in OOD and *Enterprise Viewpoint* in RM-ODP. This has been partly covered in section 4.1.1 and Figure 4.2 showing user, location platform provider and service provider connectivity to LACBA.
- The **session view** identifies independent control areas within the architecture. A session describes the amount of functions and data which are necessary to perform a certain task for each individual entity before going into the architecture component modeling. Furthermore, the session view shows the different stages of user as well as provider interaction, including the inter-working between various control entity sessions. It will include modeling of the service mobility in case of LP handovers. It relates to the *Computational Viewpoint* in RM-ODP and *Functional Model* as specified in [BD00] in OOD. In LACBA, the session view
- The **information view** describes the data structures and information passing relationships i.e. the data models for the user, LPs and registered LBS via the LACBA location services, as well as respective information passed in each session. The data model is formally described using class diagrams OOD and relates to the *Information Viewpoint* in RM-ODP.
- The **component view** shows the detailed LACBA architecture and its inter-working components, as well as sub-components, with their respective interfaces i.e. the sub-components resembling respective control areas and interfaces within and to other control areas in LACBA. In RM-ODP this translates to the *Computational View* and partially to the Object Model in OOD.
- The **communication view** shows the signaling behavior between the various architecture components i.e. synchronous/asynchronous and/or request – response communication models, etc.
- Finally, the **system behavior view** will show the signaling between the various system components e.g. by means of UML sequence diagrams relating to OOD (e.g. Dynamic and Functional model according to [RBP01]).

All but the last viewpoints describe the static features of the service architecture relating to distributed entities (e.g. sessions, data objects and components). The modeling of the entities behavior i.e. dynamic model is represented by the last viewpoint. These modeling views will be used to describe the LACBA location service functionality covering the following functional requirements as provided by the centralized LCS entity, i.e. for the provisioning of a "seamless" location service from multiple location platform providers:

- User connectivity to LACBA (i.e. the LCS) and via the LCS to LPs regarding sensor data forwarding.
- LP connectivity i.e. LP connection establishment between client (sensor data transmission) and LACBA location service (LP location service access)

- LP discovery
- LP handover
- LACBA location service provisioning of user location information and respective LP service to requesting service providers.
- LQoS negotiation, matching and adaptation signaling.

This excludes the location information generation and positioning processes relating to the location platform planes as described earlier. From the LACBA location service perspective, these are seen as external functions and will be modeled differently using approaches as defined in chapters 5 and 6. Chapter 7 will shift focus back to the LCS entity covering the location service discovery.

Due to space limitations within this thesis, simplified UML notations will be used to describe the respective modeling views. An appropriate description and explanation of the notation can be found in [Kel02][SK01].

4.3. Interfaces and Data Flow in LACBA

As already discussed in section 4.1.1, the LACBA community distinguishes between the following roles: community users, service providers and location platform providers. Figure 4.6 shows the interfaces and data flow between the various participants.

All community users use their LACBA enabled client and interact solely with the LCS via the *User API* (U API). The user registers with the LCS, manages his user profile privacy settings and generally interacts via this interface with the LP providing user information and privacy settings on his location information. The sensor data discovered by the client is forwarded to the LCS for further distribution to appropriate LP entities. Furthermore, the user control allows the specification of location update signalling intervals by the LCS and under some circumstances by the LACBA client itself in order to reduce signalling overhead [KTL06].

The LCS discovers, manages and accesses various LP services via the *location platform API* (LP API). In LACBA the LP is addressed as a whole although the LP API in fact represents the respective GIS, LES, LACBA client (LC) and manager interfaces which will be covered in chapter 5 and 6 appropriately and have been omitted in Figure 4.6 for simplicity reasons. The LACBA client sensor streams and further user interactions are forwarded to the target LPs via the LP interface.

The service providers interact with the LACBA location service via the *location service API* (LS API). Apart from the normal positioning and GIS information requests, LQoS-aware applications can also obtain and negotiate LQoS requirements as specified in the LACBA location model. As part of the LACBA location model the LBS can also obtain additional information on the current LP relevant in obtaining and handling the location service from the current LP being provided. Similar to the user sensor data GIS requests and responses are forwarded between the location-aware application/LBS and respective LP.

The user – service provider interface concerning LBS access is not considered by the LACBA framework, hence not modeled by this work. Concerning the LACBA client and service provider, the LCS has properties in form of a reverse proxy [3GP03]

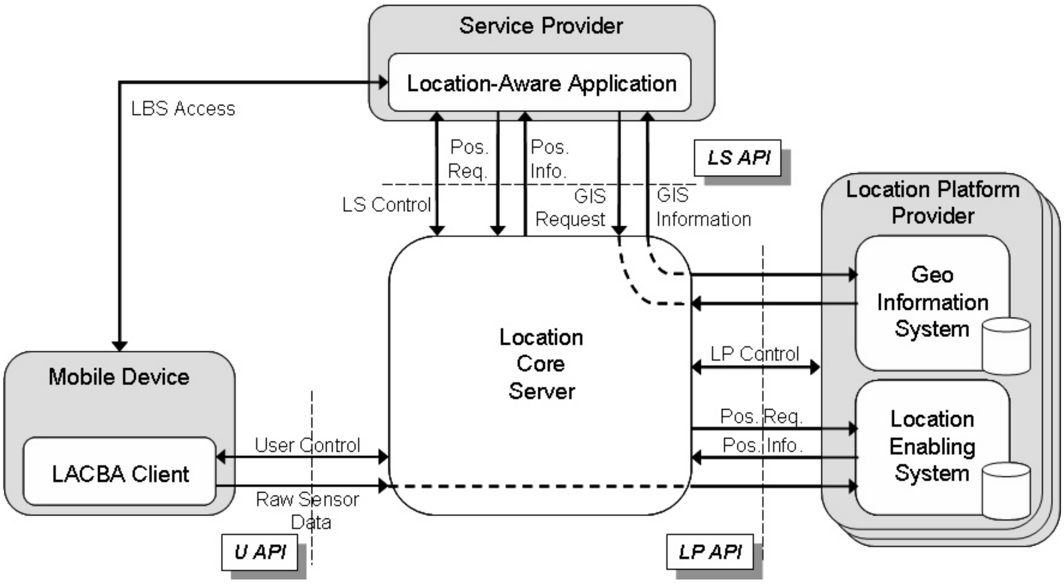


Figure 4.6 The data flow and relevant interfaces in LACBA

concerning LP access and retrieval. Hence, the LCS provides buffering, "smoothing" and to some extend supporting post-processing functions (e.g. fusion) on providing its location service from heterogeneous LP sources as explained earlier.

4.4. Modeling of the LACBA Control via Sessions

The LACBA location service control phases and processes can be modeled as session descriptions according to [Kel02]. Hereby, the session definition can be used from the TINA architecture being as follows [ILM99]:

> *A session describes common procedures and information which is shared by all processes participating in a service for a certain amount of time.*

According to this definition a session can describe complex processes involving changing user and resource configurations e.g. keeping location service sessions alive for LBS whilst changing LP sources in event of a handover. The notion of session is further refined to allow for separation of access, usage of a service and communication related issues, supporting the complex modeling of LACBA location service provisioning when handling multiple heterogeneous LP services. Furthermore, a session defines actions and not the system components.

4.4.1 Sessions in LACBA

Coming from the requirements as specified in section 2.5, LACBA defines three different session types which represent the identified major roles of the LACBA community: the community user, the service provider, and the location platform provider, all of which interact via the central LCS entity. Figure 4.7 summarizes the session relationships between these roles. A formal description in the form of use case diagrams and UML terminology can be found in appendix B1.

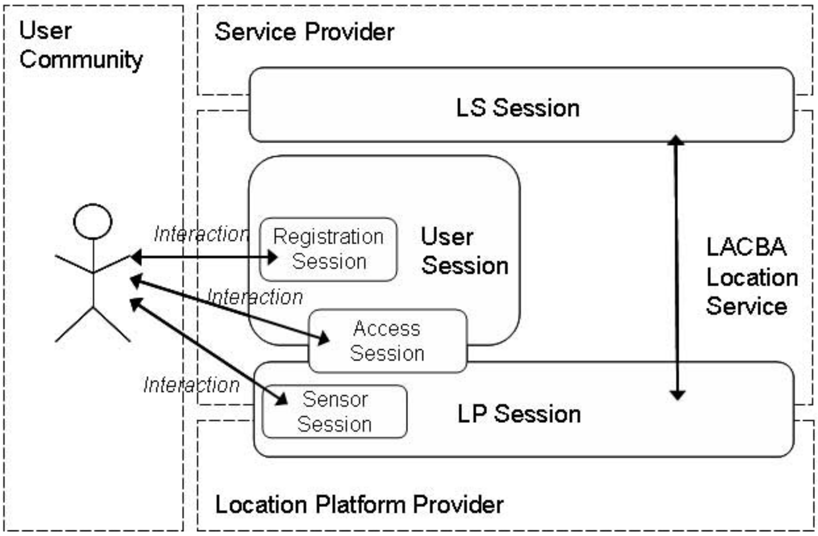


Figure 4.7 Modeling Sessions in LACBA

The sessions can be defined as follows:

- The **User Session** includes all functions which are needed to activate user localization via the LACBA location service (e.g. authentication and authorization) triggering appropriate LP sessions. Every user has only one user session, which is further divided into a registration and sensor session.
- The **Registration Session** describes the activity of a user for managing his personal settings (e.g. personal access credentials, end-device profiles, privacy settings on location information for services and other users). This session is independent of the actual LACBA location service execution.
- The **Access Session** is associated with the TINA-Access Session [ILM99] and allows the user to initiate a Sensor Session as well as participate at an existing LP Session. The Access Session remains active during a Sensor Session since it manages the Sensor Session configurations e.g. location update signaling.
- A **Sensor Session** generates a data communication session between the user's LACBA client and respective LP. It is essential for a LP Session which provides access to respective location services for the LACBA location service on the user's current location. The Sensor Session is active during a LP Session since it ensures that the LP obtains the necessary sensor or identification information which is needed to perform the positioning calculation on the user. The Sensor Session is terminated when the current LP cannot provide the location information for the respective user anymore.
- A **Location Platform Session** (LP Session) describes the discovery of-, coordination and access to LP resources (hence user location information and GIS content), providing the relevant location service access to the LACBA location service. The session is started by selection of a suitable LP discovered by a LCS location service discovery process but can terminate itself, if the location information cannot be provided anymore from the data transmitted by the respective Sensor Session. This ends the respective user–LP relation.
- The **Location Service Session** (LS Session) provides the LACBA location service for service providers, setting up and adapting/updating LBS LQoS

requirements, using LP Session resources to gain access to currently available user location information and adequate LP services. The LS Session is terminated if no more LP Sessions for a respective user are available.

According the [Kel02] we thus can determine the following LACBA control elements for our architecture:

- The **User Control** which controls the User Sessions, as well as respective Registration- and Access- Sub-Sessions.
- The **Location Platform Control** which controls the Sensor and LP Session
- The LACBA **Location Service Control** which controls the LS Sessions

The control elements have to handle all user, LP and LBS instances currently registered with the LACBA architecture.

4.4.2 Modeling of Location Service Mobility in LACBA

The modeling of the LACBA control elements using sessions also can be used in modeling the location service mobility, i.e. LP Session handovers. This is realized by the dynamic generation as well as provisioning of Sensor and LP Sessions independent of the LS Session control. The aim is to provide seamless LP Session handovers whilst maintaining a LACBA LS Session with the LBS and respective service provider. This is made possible by allowing multiple, parallel Sensor, and LP Session relations with LPs per LACBA community user via the LCS. Furthermore, the availability of at least one LP in almost any daily usage environments can be guaranteed. As discussed in section 2.5, various and readily available LP solutions can provide location information and GIS content for most situations. Hence a "handoff" to such a LP providing this kind of service should almost always be possible. Figure 4.8 illustrates a vertical LP handover (i.e. from C to D) using a hierarchical tree structure of LPs. The proposed hierarchy and geographic service area definitions will be covered in chapter 7 more thoroughly.

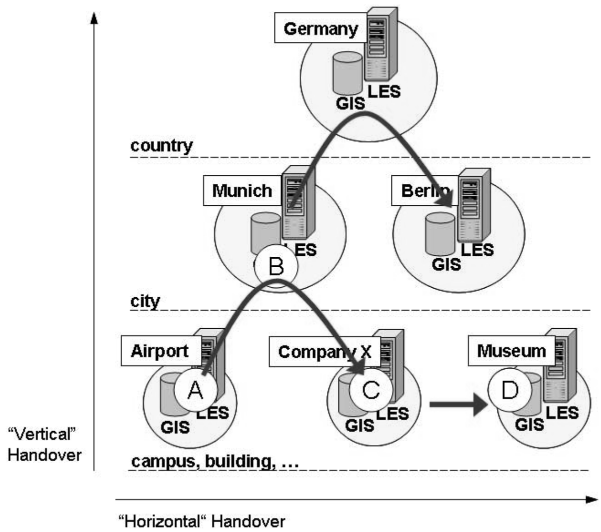


Figure 4.8 Location platform handover examples

A horizontal handover from A to C, i.e. on the same hierarchy level is less likely to occur if the respective service areas do not overlap. A more likely case is a vertical handover to a higher level LP since the LP in "B" most probably includes both A and C within its service area. Hence the handover scenario will probably be A – B – C. If a suitable LP is found in the event or an impeding handover, LACBA follows a "make before break" principle. LACBA supports the service mobility across different LPs but not changing end-devices (i.e. device mobility). When the user changes his end-device, he has to re-register with LACBA and re-initialize all of his sessions. Seamless network mobility and internet access is seen as a perquisite for the LACBA location service, e.g. a constant network link over UMTS for sensor data transmission and LACBA access. Providing service mobility across heterogeneous networks should be provided independently of LACBA. In the event of a network failure or any termination of an active communication link, i.e. moving out of the coverage of a WLAN hotspot and re-establishing a network connection using the same end-device over a mobile operator network always requires re-registration by the user with LACBA. Current active session tokens with respective LPs, e.g. Sensor and LP Sessions can be re-activated within a certain specified time-out period.

Figure 4.9 illustrates the service mobility in LACBA using sessions. User 1 accesses the LACBA control mechanisms at different times (t_x) using different location platform providers. In doing so his current LACBA LS Session always remains active. User 1 has to initialize a new Sensor and LP Session every time he enters a different LP service area designated for his current usage environment. Location Platform Provider 1 and 3 could be non-overlapping, building-based indoor LP solutions as mentioned earlier (Figure 4.8, A and C). Location Platform Provider 2 could be the respective outdoor, city or country level-based provider system, demonstrating the vertical LP handover situation (A – B – C). Upon just about leaving the location platform service area of Provider 1, User1 can initialize a new Sensor and LP Session via an Access Session within LACBA, hence being connected to Provider 1 and 2 simultaneously (t_1 + $t_{1\text{-}2}$). Both LP Sessions are available to the LS Session at that time, but t_1 remains the currently active one. By leaving the service area of Provider 1, User 1 looses the Provider 1 Sensor and LP Session and the LS Session has to switch to Provider 2 LP Session resources completely ($t_{1\text{-}2}$ / t_2).

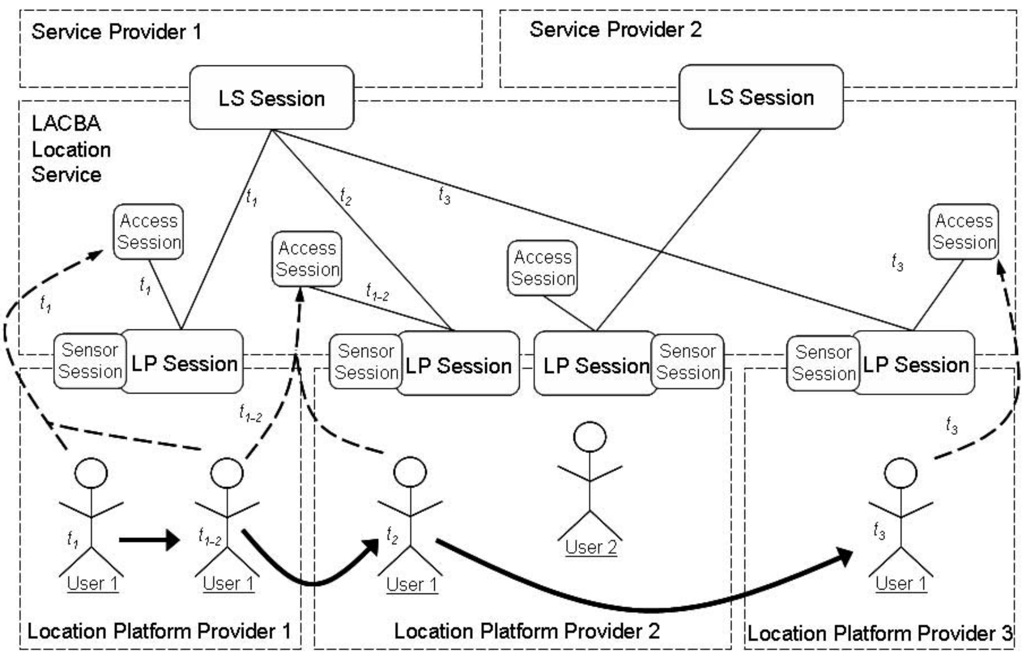


Figure 4.9 Modeling of the location service mobility via Sessions

When User 1 moves into the service area of Provider 3 (t_3), this could potentially result in User 1 loosing his Provider 2 Sensor and LP Sessions if Provider 3 is indoors and GPS was used with Provider 2, thus User 1 loosing his GPS signal before the handover transition ($t_2 \rightarrow t_3$) or LP Session t_3 could have been activated for the LS Session. In such cases special pre-signaling and LP discovery mechanisms are needed [ZD06b]. The formal description of Figure 4.9 illustration using use case diagrams will be given in appendix B2.

4.5. Information Modeling and Data Structures

Having defined the session view, this section will now present the data structures of each session types. The information content of each session will be defined as a session *description* and the corresponding data structures exchanged between the control areas will be defined as session service *graphs* [Kel02]. Table 4.1 shows the LACBA session descriptions and graphs, their information content and parameters, the responsible control area(s) as well as typical example service and parameter configurations in each case.

Control Entity	**Description**	**Parameters**	**Example**
User Control	User Session Description	User profile, LP- discovery and connection services	UserA, findLP(), connectLP(), disconnectLP()
User Control → LP Control	User Service Graph	Service name, user, LP and sensor parameters from user point-of-view	findLP(UserID, Sensor Packet), connectLP(UserID, LPID)
LP Control	Sensor Session Description	Participants, session token, explicit addresses, sensor interface, type and format, identification location update interval	LACBA client–LP: Session(session.token, GSM, CellID, MSISDN, 6 sec)
LP Control → Adapter	Sensor Connection Graph	Parameters of a Sensor Session connection	Packet(session.token, timestamp, [CellIDx, RSS88], [CellIDy, RSS67] ..)
LP Control	LP Session Description	LP profile, LP service offers	getPos(), getMap(), route(), pointOfInterest()
LP Control → Adapter	LP Connection Graph	Service name, user identification and parameters form location service point-of-view	LCS–LP: getPos(UserID), getMap(size, resolution), route(startAddress, endAddress)
LS Control	LS Session Description	LACBA location service offers, Location source type and information on user, LQoS requirements and classes	posRequest(LQoS parameters, no care), LBS–LP:getPos(), getMap()
LS Control → LP Control	LS Graph	User location information and services	UserA.LP1.getPos()

Table 4.1 The LACBA location service information descriptions

The functional relationship between each service session description and graph is illustrated in Figure 4.10. From the user/LACBA client perspective, the *User Session Description* holds service descriptions of services which can be triggered by a registered user at the LCS (i.e. triggered by the LACBA client) handing the required parameters to the LP Control via the *User Service Graph.* On such a service request, the LP control performs a discovery query and/or establishes the required Sensor and LP Sessions. Using the user information handed via the *User Service Graph* and sensor data provided by the LACBA client, the LP control hands the required parameters to a suitable Adapter component in a *Sensor Connection Graph* using the corresponding LP connectivity information provided via the *Sensor Session Description*. The service descriptions on particular LP entities on accessing respective location services are provided in the *LP Session Description* and are also handed to the same Adapter component used for the corresponding Sensor Session via a *LP Connection Graph*. Regarding the LBS perspective, the *LS Session Description* provides the service descriptions for LBS on accessing user location information of a user via the LACBA location service from available LP resources.

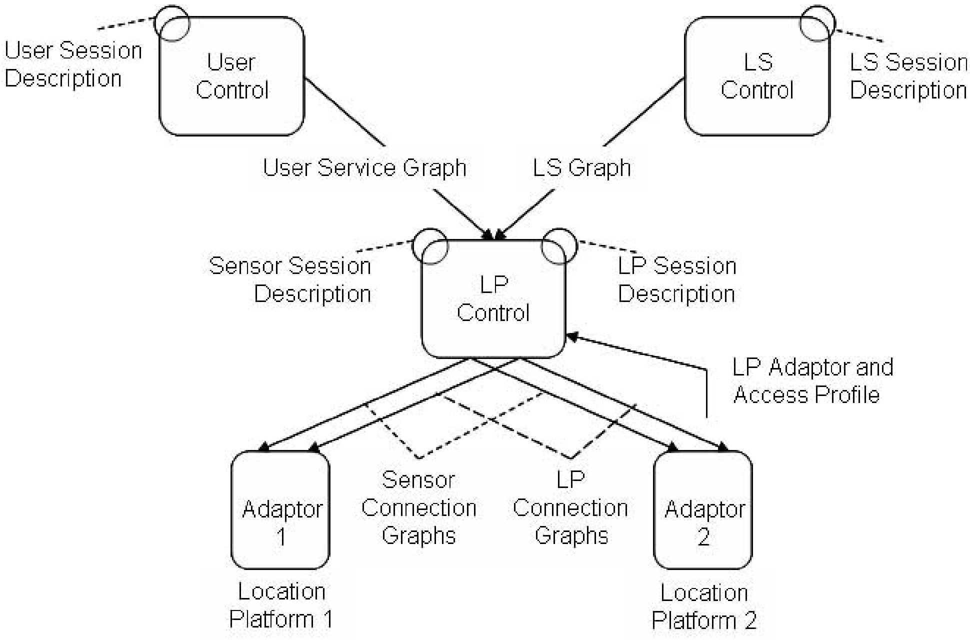


Figure 4.10 Functional relationships between the information descriptions

The parameters required for location service access of a LP being capable of providing the agreed-up LQoS on a user are handed to the LP control via the *LS Graph*. Figure 4.11 shows the relationship of the session information models and modeling definitions used in the next sections. The *Communication Relationship* relates to an *Information Path* describing the end-to-end communication between two *Participants*. A Participant can either be a User/LACBA client or server.

In the sensor session case, the end points (i.e. *Terminals*) of the *Information Path* are the LACBA client and the currently active LP whereas in the *LP* and *LS Session* case, the end points are the LCS and the active LP corresponding to the Sensor Session, as well as the LBS service provider entities respectively. A *Service Relationship* consists of a *LP Resources Set*, which in turn can consist of many Information Path (i.e. LP Sessions) references which are active for a current user. The *LP* and *Sensor Connection Graphs* describe on how to the end-to-end communication relationship to each end point can be physically realized in each case (i.e. IP addresses, target web service URL, etc.). The LS Graph on the other hand describes the currently active LBS and LP relationship used.

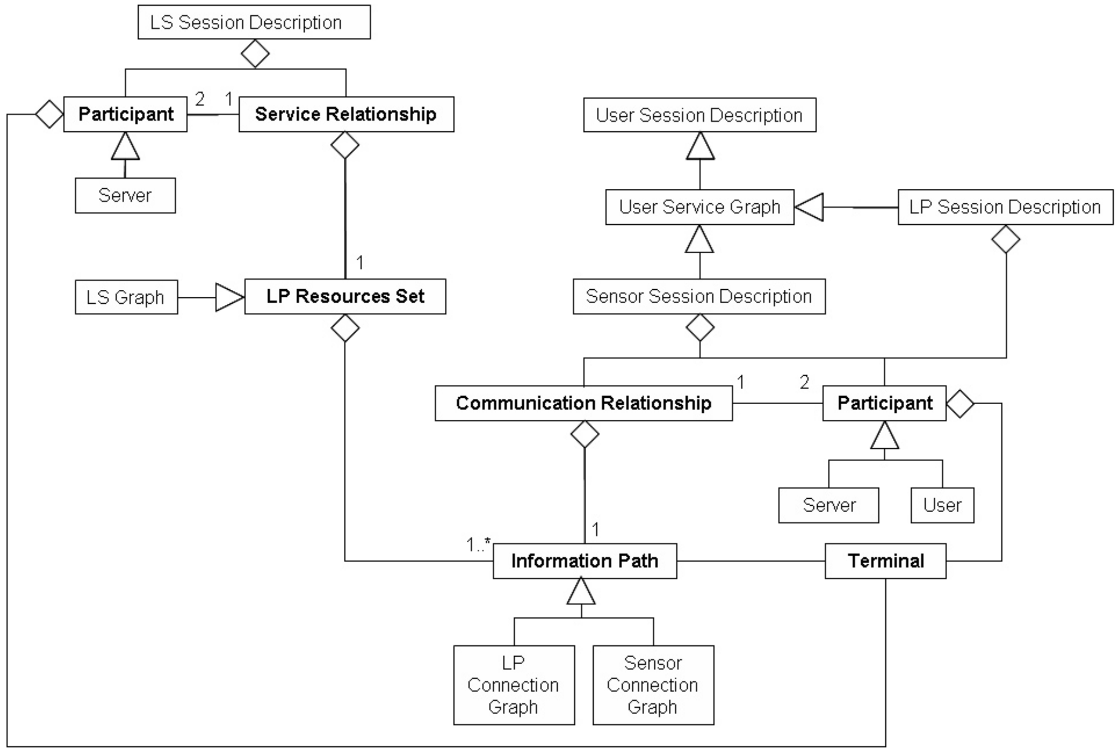


Figure 4.11 Modeling definitions and relationships used for information modeling

4.5.1 The LACBA Location QoS and Model

In service architectures one has to distinguish between different levels of QoS i.e. user perception, end-device, communication networks etc. The user QoS is usually described by subjective parameters e.g. "high quality", "voice-quality" or "fast transmission". When talking about LBS the user may describe "high accuracy", "high map resolution and/or detail", or "map/location update speed". This usually coincides with the related end-device QoS, which mainly is concerned with the presentation quality of LBS e.g. screen resolution, general I/O capabilities. The communication QoS in turn deals with parameters such as bitrate, delay, jitter etc.

In this work we concentrate on the quality of location information and how it is delivered to the respective LBS. The latter aspect is mainly concerned with the availability of the location information source and how fast the current location information can be made available to the LBS. Apart from the actual location information value and its format, the LBS needs to be able to evaluate the quality of this information, which is partly related on information used in generating this location information (e.g. type and quality of reference source(s) used) as well as positioning calculation which has been carried out.

Defining the Location QoS Parameters

In order to model the QoS of location information for LBS the parameters for the LACBA location model have been adapted from the FRIARS[6] model [FM01]. The FRIARS model states that values are associated with: 1) information needs, and 2) the way in which information is delivered. As such, they provide a direct measure of how

[6] FRIARS: Freshness, Reliability, Initiation Time, Accuracy, Resolution and Scope

the customer (i.e. service provider) needs the information to perform the task of interest, and how well the ubicomp system (i.e. LACBA location service) can provide (or receive) the information, given the currently available bandwidth, devices and information sources. The goal of the ubicomp system is to find information sources, devices, and bandwidth that satisfy the need with appropriate (matching) values for the qualities of service (QoS).

The following LQoS values have been determined to be the most relevant for the LACBA location model: *initiation time*, *accuracy*, *precision*, *availability*, *scope*, *resolution*, *location update interval*, and *freshness*. Each of them specifically can provide useful information about the location source (i.e. LP), its provided location information, and delivery aspects for evaluation in the LACBA location model.

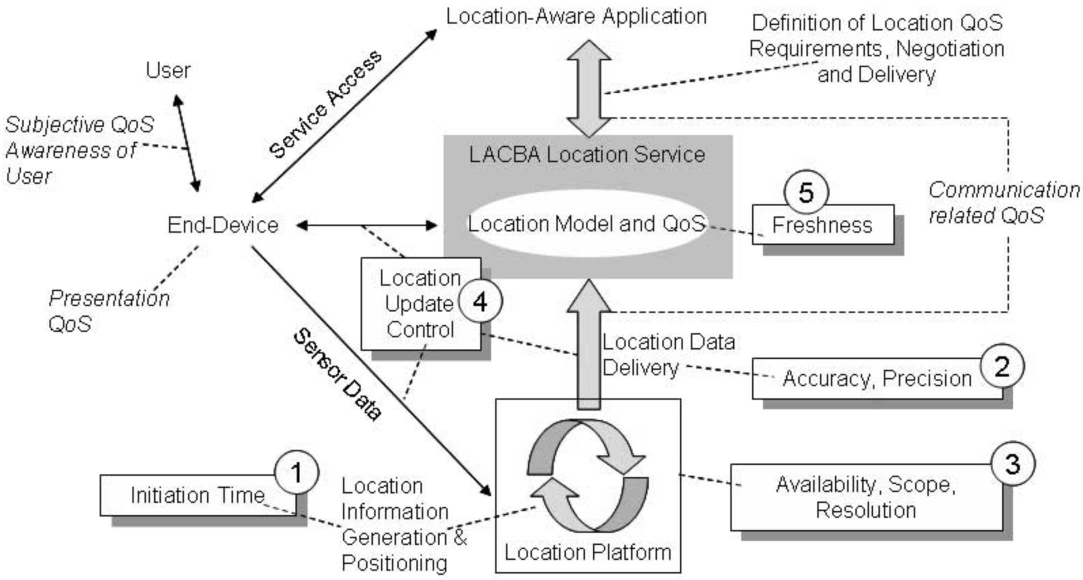


Figure 4.12 Location QoS and origins within the LACBA framework

Their specific origin within the LACBA framework is illustrated in Figure 4.12. The parameters (1) and (3) can be associated with the LP source whereas the actual location information from the respective positioning methods can be evaluated by (2). The latter values are determined by the location information generation, reference source and technology, as well as positioning methods used by the LP, but the LACBA location model can only measure and evaluate the respective outcome as given by (2). Location information delivery aspects are determined and controlled by parameters (4) and (5), whereby (5) is just an indicator since the last location update.

Communication QoS aspects such as transmission delays can influence (4) in case of communication bottlenecks (e.g. using GSM GPRS or even CSD connections) or high network traffic load. In the age of ever increasing wireless and wireline bandwidth, however, this is seen as a minor effect and won’t be considered by this work. User subjective QoS awareness and presentation QoS are also not the focus of this work and should be treated outside the LACBA framework.

Modeling of the Location Service QoS

LACBA supports both legacy LBS and LQoS-aware applications which require individual parameter configuration possibilities (i.e. "customized" service class). Legacy LBS can specify their LQoS needs globally via three abstract parameter classes: "high", "regular" and "no care". These levels of LQoS have specific requirements which will be matched by the LACBA location service on available location sources respectively within the location service session. The "high" class requires highest location source availability, location information quality and fast delivery. The "regular" class has normal/acceptable quality suitable for a majority of LBS, and "no care" has no specific expectations to the LACBA location service requiring nothing more than a location service source. Table 4.2 gives an overview on these classes with service examples from section 2.1.

Service Class	**Description**	**Location Platform QoS**	**Location Information QoS**	**Location Update intervals**	**Additional Function Support**	**Example Services**
high	"seamless" availability of location information, very high accuracy and precision, usually very short location information delivery times	High availability (95% of time) global and local scope point accuracy	High accuracy (outdoors <10m, indoors <2m), <1m e.g. robot control High precision (2dRMS, 95%)	Usually very short, 2-3 seconds, "real time" for control purposes	Fusion and/or estimation at LACBA location service level pre-handover signaling for LPs	Emergency services, medical alert, asset tracking, automated control of robots, family finder
regular	Locally and globally available location information at medium to high accuracy/ precision, tolerating occasional lack of availability	High to average availability Global and/or local scope Point accuracy and/or high probability	Medium accuracy (outdoors 10-50m, indoors <10m High to medium precision (2dRMS 95% to CEP 50%)	Varies depending on speed of tracked object, on average 5-10 seconds	Not required ad-hoc discovery of new LPs	Navigation, person/fleet-tracking, friend finder
no care	Availability of any kind of LP is more important than other LQoS factors	Average availability local scope area accuracy	Average accuracy 50m-500m, "room" and "building" accuracy	Usually not tracking services, or triggered/ event-based	Not required ad-hoc discovery of new LPs	Bulk Emergency alerts, m-info/events, m-adverts,
customized	Completely customized LQoS specification	Individual values on every LQoS parameter				LQoS-aware LBS

Table 4.2 LACBA LQoS service classes

The high class must ensure a very high availability of location information, i.e. ensure the availability of appropriate local and global LPs. In order to compensate for accuracy deficits, location source fusion could be applied. On the other hand estimation functions can compensate for a lack of availability or enable shorter location update intervals. In order to ensure a seamless transition between LPs, pre-signaling, and handover preparation mechanisms can be provided [ZD05b]. The lowest class of "no care" is usually dedicated to LBS covering a certain local area, i.e. traffic or local event/information broadcasts. A special case is represented by the "customized" class. This is dedicated to LQoS-aware applications allowing free parameter configurations and dynamic adaptations as needed.

The LACBA Location Model

As mentioned earlier each localization type has a different location model output (i.e. point model used in geometric and scene analysis methods, statistical approaches provide probabilistic location models and the proximity types a respective area model). A common location model is needed to evaluate between various location sources and respective LQoS. In LACBA, we have chosen physical location over semantic location models as discussed earlier in this section. The most logical representation of a physical location is a point in space given respective measurement error with regards to the "true" physical position. This point definition as illustrated in section 2.2.1, is the most desirable location model, but this does not work for all positioning systems: COO-based systems define the service area of the serving GSM/UMTS cell as possible user locations, hence follows the area model. As mentioned in earlier chapters, probabilistic location models need to convey the probability distribution for a measurement expressing the probability that a user could reside on a reference grid mapping to a point in space.

The area localization type provides a much simpler and neutral method to combine certain positioning system outputs [Rot05][Leo03]. This can be best explained by the following example where a point location can be expressed as an area location:

- GPS provides a point location with an accuracy equal ~15m 2dRMS in inner-city locations i.e. 95% probability that all measured locations reside within a ~15m radius around the given location.
- A GSM COO-based method provides an area location with an accuracy equal to the cell radius of roughly 120m in inner-cities, and several km on the country side.
- A probabilistic method can be mapped to an area model whereby the variance can determine the accuracy value i.e. probable radius. The most probable location on the reference grid can be mapped to a point in space respectively.

Fusion is also possible with the area model. Estimated positions and overlap of respective areas (i.e. intersection) can represent the most likely position of the target [Rot05][Leo03].

The main function of the LACBA location service is to provide an intelligent selection as well as adaptation of required LQoS. Hereby the area model provides the most effective and simplest approach on processing the various LP location data outputs as well as ease of converting them into this common location model by means of corresponding adaptor entities. The further modelling of the LACBA location service will still consider fusion as an option, but leave a concrete adoption of it as future work. Both fusion and intelligent selection of location sources are important when aiming for seamless and best possible location service provisioning using multiple and heterogeneous positioning systems.

4.5.2 Data Model of the User Session

The *User Session* (Figure 4.13) manages the user data of LACBA community members which is represented by a number of specific profiles. Each profile contains a set of data objects which can be either manipulated directly by the user, or only by specific providers via the respective service control elements.

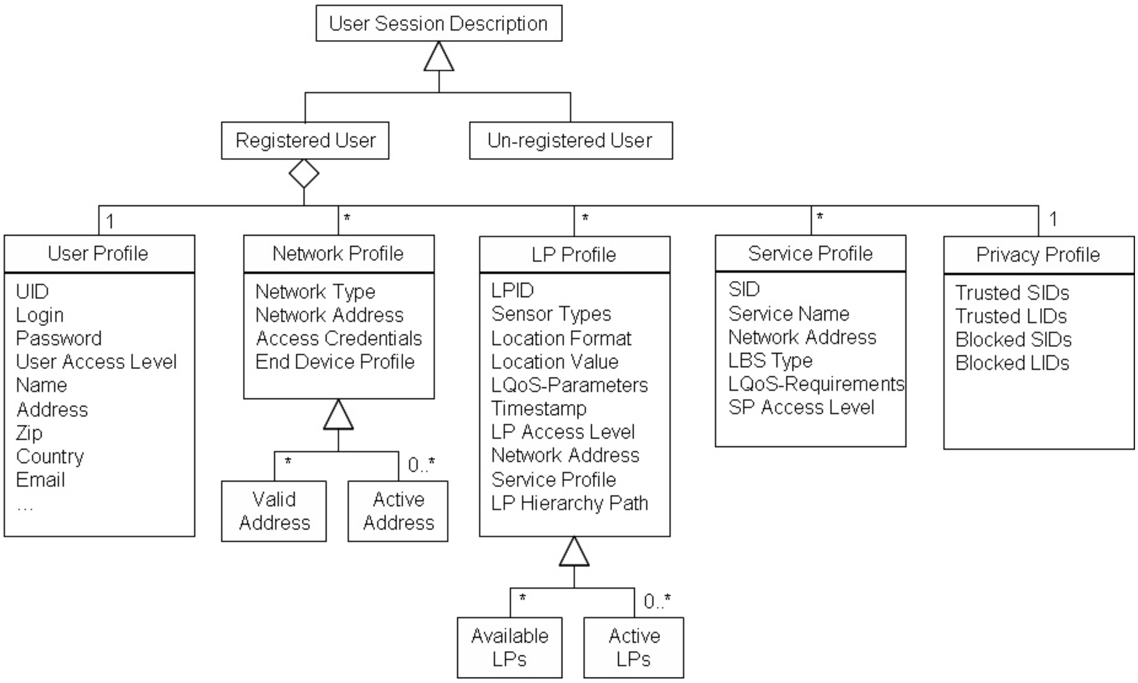


Figure 4.13 UML class diagram of the User Session

A user can manage his personal information regarding his identity, privacy and contract with LACBA, end-device configuration as well as access credentials within the *Personal Profile*. All LACBA users are addressed via their unique user ID (*UID*) with the LACBA location service.

The current LBS service and location platform providers registered for the user providing information about current user location as well as LBS using it are stored in a *LP*- and *Service Profile*. The *LP Profile* enables to differentiate from which LP providers the user currently obtains location information (i.e. *Available LP*) and which of these is currently being used by a service provider (*Active LP*). The current service provider LBS with their LQoS requirements currently available and obtaining location information from a user are stored in the service profile. Each provider also has a unique LP (*LPID*) and LBS service IDs (*SID*) within LACBA.

Information about the current communication network attachment(s) i.e. ability to reach a user from a network perspective are stored in the *Network Profile*. Apart from the UID, a user can also be described by the network addresses he is currently reachable from. Network addresses through which the user is currently connected to LACBA and location platform provider via a Sensor Session are indicated by an *Active Address*. The available network and access credential information can be used for the discovery of- and location information retrieval from respective LPs (e.g. mobile operator). The *End Device Profile* described the features of the users end-device(s) associated with the corresponding network interfaces.

The *Privacy Profile* holds a list of trusted and blocked SIDs as well as UIDs respectively. They are checked every time a candidate LP is discovered for a particular user or if a LBS service provider tries to obtain a location service on that user. Whilst in the latter case it is clear that a user wants to block certain “annoying” LBS service providers (e.g. LBS push advertisements), the former case ensures the privacy of the user on location platform providers with limited trust despite of their possible sufficiently high LP Access Level (e.g. community-based providers)

The *Access Session* triggers the LP discovery and initiation of the *Sensor* and *LP Sessions*. Within these sessions the current client sensor data and selected data objects from the *Network-* as well as *LP Profile* are handed to the *LP Control* in each case.

4.5.3 Data Model of the Sensor and Location Platform Session

A Sensor Session in LACBA describes the state of an active sensor data transmission session between a LACBA client and a LP. Similarly to the object-oriented session model described in [Kel02], the session model used in LACBA describes the current state of a “session” by means of relevant objects (e.g. user or server objects) and corresponding parameters. In this context, the work by [SK01] provides a generic service model to be used for rapid service realization, describing the session in terms of end points (e.g. participants), communication channels (e.g. information paths) and communication relationships. Figure 4.14 illustrates the object-oriented description of the sensor session based on this service model.

The *Sensor Session* in LACBA is globally described by a unique session ID which is generated for each LACBA user and LP relationship, and the duration of this relationship. It is composed of two participant and one communication relationship objects. The LP server object is described using the LID, its security (LP) access level, a network IP contact address, and the sensor types and values it supports for localization.

On the other hand, the user object is described using the UID, user access level and network profile containing his current network access and end device configuration (including sensor devices available). The communication relationship object consists of the information path containing all information on how to connect the LACBA client and LP physically (e.g. IP addresses, URLs), as well as appropriate sensor data configuration for transmission. This information is communicated to the relevant adapter entities via the *Sensor Connection Graph*.

Following the *Sensor Session* modeling example, the *LP Session* in LACBA (see Figure 4.15) is also globally described using a unique session ID generated for the respective location service access to the LP, the duration of the relationship, descriptions on services available from the LP, as well as a specified update interval on triggering respective service calls. This time, the LP server object includes the service access descriptions (e.g. URLs of location service web services) instead of the sensor type information required during the *Sensor Session*. Again, the communication relationship object of the *LP Session* consists of the information path describing the appropriate physical access to the LP services. Once the LP Session is established, the respective LP source is registered as an "Available LP" for a user in the user and LP profile and respective location service calls can be made via the *LP Connection Graph.*

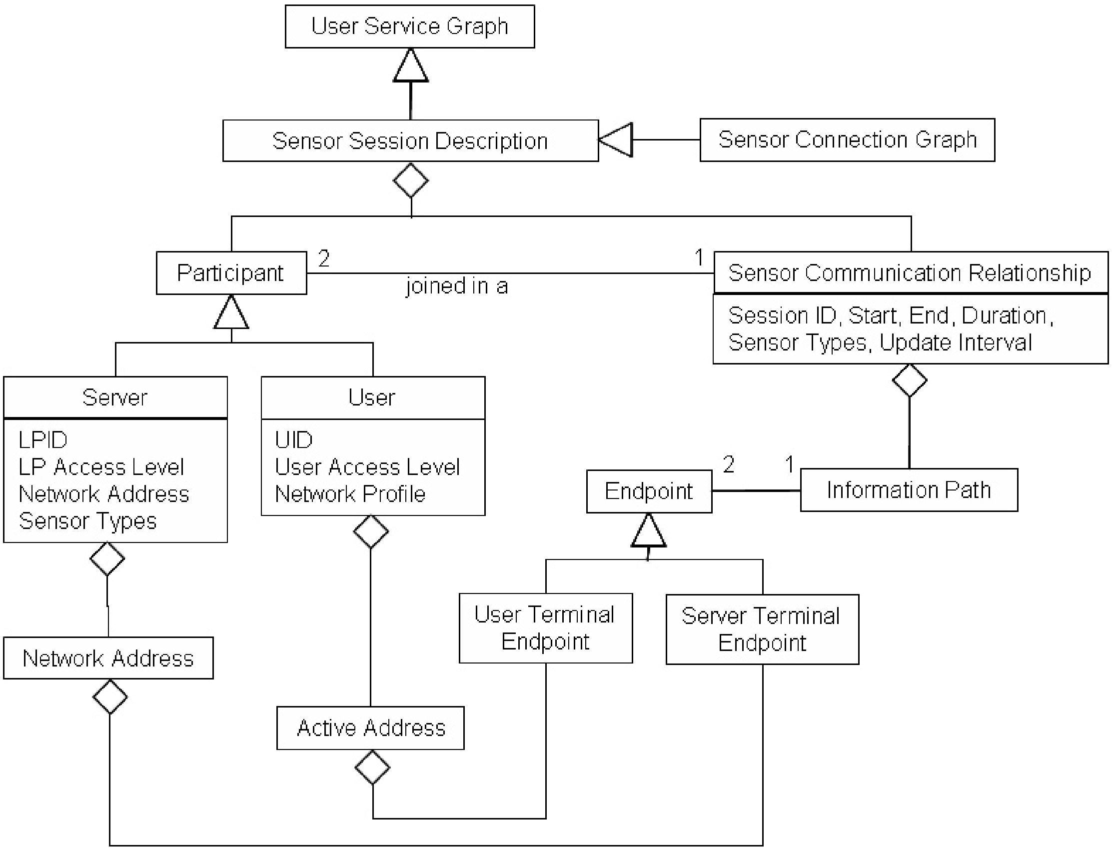


Figure 4.14 UML class diagram of the Sensor Session

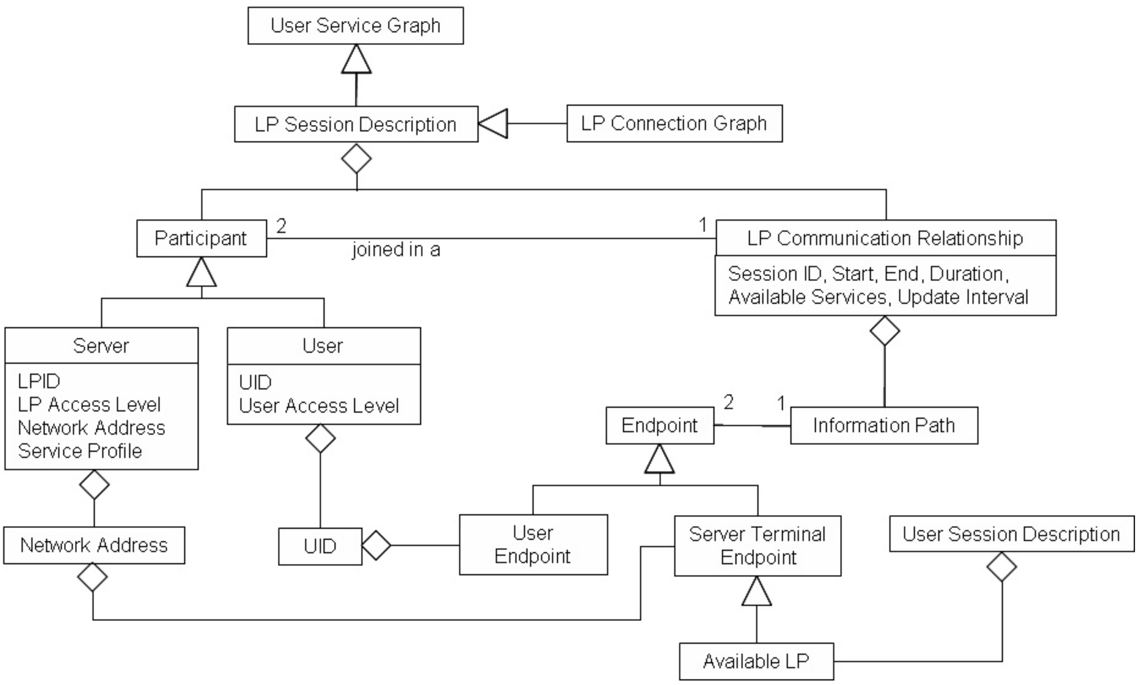


Figure 4.15 UML class diagram of the LP Session

4.5.4 Data Model of the Location Service Session

A LS Session (Figure 4.16) is also comprised of two participant and one communication relationship objects describing the location service access for LBS requesting access to location information and further location services on a particular user. The LBS provider server object is described by the unique SID identifier, connectivity information, the required LQoS parameter settings, the LBS type definition, as well as his service provider (SP) access level.

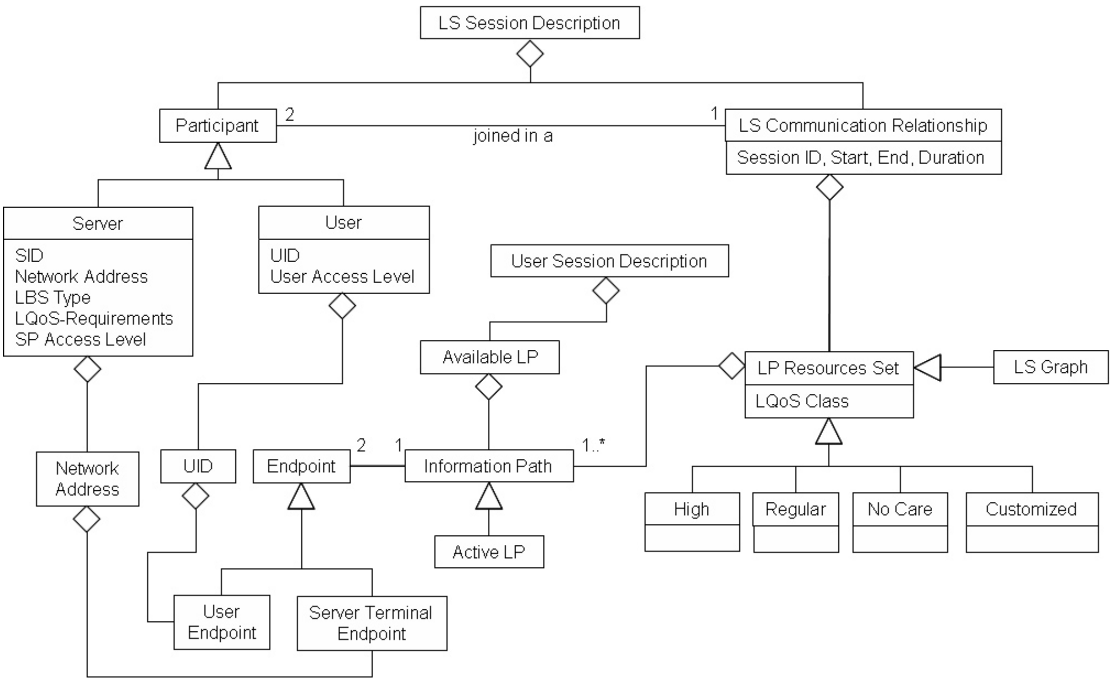
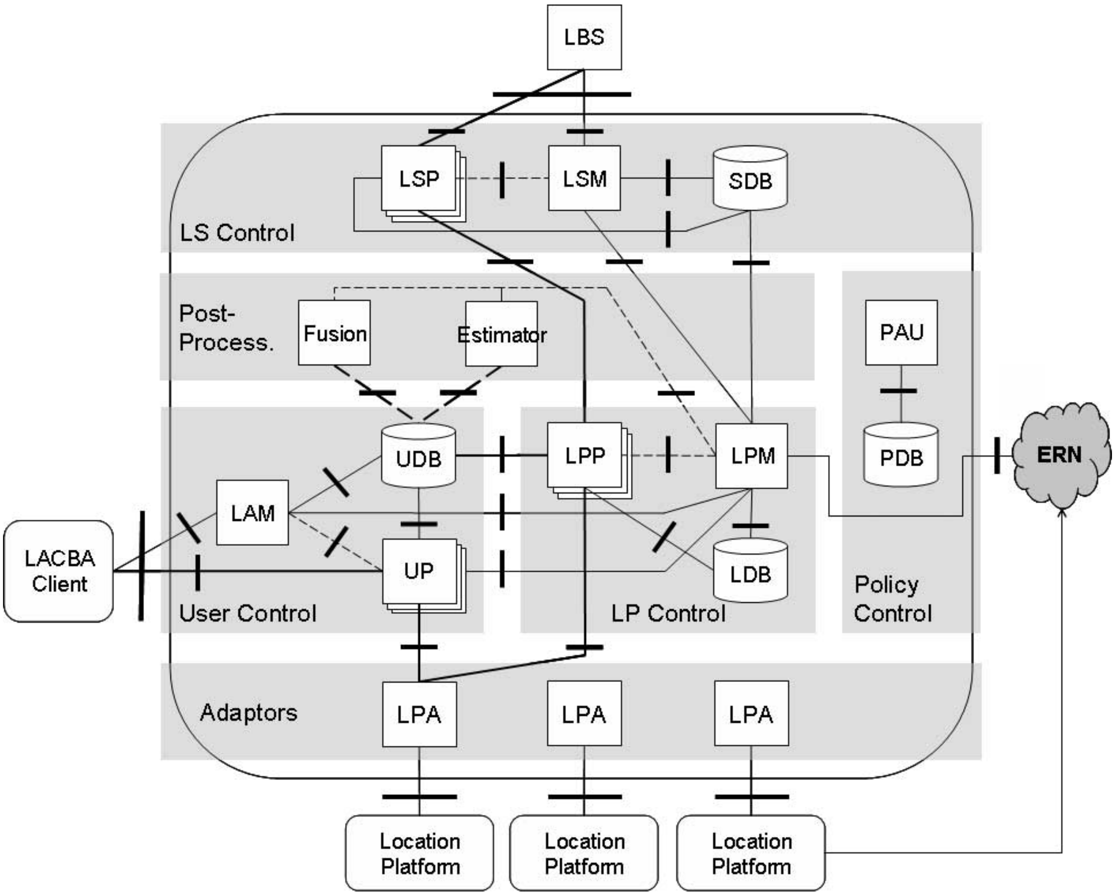


Figure 4.16 UML class diagram of the LS Session

Again similar to the other session types, in the communication relationship object with the LCS, the requesting LBS is uniquely identified by a session ID and the duration of the connectivity relationship as well. The communication relationship object consists of a *LP Resource Set* (similar to the Information flows in [Kel02][SK01]) which is marked by the LCS and LBS agreed upon LQoS class (e.g. high, regular, no care or customized), whereby the customized class contains the individually negotiated LQoS parameter configurations. The LP Resource Set itself contains all “Available LP” Information Path references of the user in question, meeting the agreed upon LQoS requirements. From this set of LP Information Path references, the LP source selected by the LBS for location service access is defined as the “Active LP” in the user and LP profile appropriately. Communication to the appropriate LP Information Path reference occurs via the *LS Graph*.

4.6. LACBA Components

The goal of the component view is to show the decomposition of the LCS (i.e. LACBA location service) in specific components with their respective functionalities. By doing so, we will define the internal interfaces of the architecture. Coming form the previous modeling views, Figure 4.17 shows the LACBA architecture in resemblance to the Sessions distinctively defining the three control parts.



Key:

LAM LACBA Access Manager
UP User Proxy
UDB User Profile Database
LPM Location Platform Manager
LPP Location Platform Session Proxy
LDB Location Platform Database
ERN External Resource Network

LPA Location Platform Adaptor
LSM Location Service Manager
LSP Location Service Session Proxy
SDB Service Profile Database
LBS Location-Based Service
PAU Policy Authentication Unit
PDB Policy Database

Figure 4.17 The LACBA components and internal interfaces

4.6.1 The User Control

Tasks related to managing user preferences and interactions with the LACBA platform are taken care of by the *User Control* and are governed by the *User Session*. The tasks can be subdivided into those relating to tasks independent of the LP Session issues e.g. managing personal user settings in the user profile, and those relating to active task involving LP discovery and sensor Session initialization when establishing a LP Session. The former tasks of user registration and management of personal user settings will not be covered in this thesis.

Components of the *User Control* include a user database (UDB) holding the user profile information of registered users, an access manager (LAM) controlling the user access to the LACBA platform as well as a *User Proxy* (UP) which is instantiated for every active user managing his service access and sensor data forwarding. This trisection of the control components has been found very useful and bears close resemblance to the SAMSON architecture approach on performing the user control in distributed service architectures [Kel02].

The *LACBA Access Manager* is the centralized entity for all community user signaling requests which register with the LCS. The LAM handles the login sessions and verifies the users access rights and privacy policies via the *Profile Authenticator Unit* (PAU). If the login is successful, the IAM generates a UP instance and directs all signaling requests relevant to this Access Session to this entity.

A *User Proxy* primarily manages the user profile access by users (i.e. for administration of personal settings), registers current end-device network configurations and provides service access to the user client. The service access involves the initiation of LP discovery and *LP Session* requests as well as *Sensor Session* setup and control (i.e. sensor data transmission flow regulation via appropriate location update signaling intervals). In each case the UP will include the necessary parameters from the user profile database and forwards sensor data streams from the client to LPs (via a respective LPA) and/or *LP Control* entities. The UP is kept active as long as a user is participating with LACBA.

The *user database* (UDB) holds all user profiles, including currently associated and active LPs as well as associated service providers obtaining location information via the LPP. Hence the UDP is not only being accessed by UP and IAM entities but also has an interface to the LPP of the *LP Control* part.

4.6.2 The Location Platform Control

The *LP Control* is similar to the *User Control* in that it manages to interactions of location platform providers via *LP Session*s from the LACBA perspective. Again the tasks can be subdivided into LP service access and location platform provider specific information relating to the location information provided. Unlike the *User Control* the location platform providers as participants in the LACBA framework just provide their services to LACBA and do not administer their information themselves.
Pursuing the *User Control* configuration example, the *LP Control* includes a location platform database (LDB) holding location provider specific information and service state with LACBA, an access manager entity (LPM) handling the LP connectivity and service distribution in LACBA, and a proxy entity (LPP) which is instantiated for every *LP Session* forwarding respective LP service requests.

The *Location Platform Manager* (LPM) handles all tasks related to the LP interactions with the LCS. The tasks involve the LP discovery, *LP Session* initiation and inter-working, LP handover, and LPP provisioning according to LQoS parameters for *LS Sessions*. The LPM receives LP discovery and *LP Session* initiation requests via instantiated UP entities. The LPM performs search queries using sensor data “snapshots” from the UP via the interface to the *External Resource Network* (ERN). The developed LACBA LP discovery service introduced in chapter 7 will return candidate LP access information. Upon successful user or service provider access rights verification via the PAU unit the LPM provides corresponding *Sensor Session* descriptions to the requesting UP and initiates a LPP instance, or in the latter case provides candidate LPPs meeting service provider LQoS requirements to requesting LSM entities. Despite being primarily concerned in managing *LP Control* related tasks, the LPM also handles the interactions with the post processing entities involved in fusion and estimation functions when establishing LPP entities.

A *Location Platform Proxy* (LPP) provides access to all services available from a registered LP including location information of a respective user. Location information and LQoS parameters on a users position obtained at specified location update intervals or ad-hoc location service requests are stored in the respective user profile in the UDB. The LPP is instantiated after a corresponding UP unit is established, and is kept active as long as the LP can provide a location service on the corresponding user and/or the UP is still active. Similar to the UP the LPP interacts via an appropriate Location Platform Adaptor (LPA) with his designated LP.

All LP related information is stored in LP profiles in the *location platform database* (LDB). The LPP states (i.e. availability or actively used by a LSS) and other LP related information are registered and administered by the LPM and LPP in the LDB. This information also includes LP service and Sensor Session descriptions as well as a *Location Platform hierarchy Path* (LP hierarchy path) indicating the hierarchy level and vertical handover LP relationships. The meaning of this will be discussed in chapter 7.

4.6.3 The LACBA Location Service Control

Following the example of the previously explained control entities, the *LS Control* bears strong resemblance to these as well. It manages the interactions of service providers via *LS Sessions* from their perspective. Its task is to provide "seamless" access to current user location information and additional LP services with respect to LBS LQoS requirements. Again, a service provider cannot mange his LACBA profile and service requirements directly. They are established and registered with LACBA only during the lifetime of a *LS Session*.

The *LS Control* includes the following components: The service provider application profile storing the current location service requirements is stored in the service database (SDB). An access manager (LSM) component manages the provisioning of available LP resources according the requirements set in the application profile. The provisioning of user location information corresponding to the requested LQoS and interactions with respective LP services, i.e. the *LS Session,* is carried out by a designated proxy entity (LSP).

The *Location Service Manager* (LSM) provides access to the LACBA location service for service providers. It handles the initial LQoS request, negotiation and LS Session initiation, registering the agreed LQoS in the SDB for the to-be served LBS. The LQoS can be dynamically adapted and re-negotiated during the LS Session via the LSM. LQoS-aware applications can register specific LQoS parameters, whereas legacy LBS are supported via static LQoS class descriptions with set LQoS parameters (section 4.5.1). Upon successful verification of the service provider LP access rights via the PAU and LQoS registration, the LSP is configured and instantiated accordingly.

All signaling relating the access to user location information and LP services access is done via the *Location Service Proxy* (LSP). The LSP uses appropriate LPP entities in order to provide access to LP services and user location information. Furthermore, the LSP compares the available user location information LQoS against the registered LQoS in the SDB and can thus trigger a necessary LP handover (i.e. perform a LPP switch) whilst maintaining an active LS Session with the LBS.

The *service database* (SDB) basically registers the LBS LQoS “contract” with the respective service provider. This information is monitored by the LSP for possible LQoS degradation (i.e. decreasing quality of location information due to environment changes e.g. EMI, GPS urban canyon problem, etc.) and/or LPM if new LP entities become available. The LSM is the only entity being able to manipulate this data.

4.6.4 Profile Authentication and Policy Control

The *Policy Control* has a subordinate role compared to the other control parts in providing a LACBA location service, but is very important in enforcing privacy and security policies relating to user location access by service providers and other users, as well as LP access by users and service providers. The *Policy Authentication Unit* (PAU) performs the access verification (using respective LS, LP and User access level records) on respective policy records in the *policy database* (PDB). The LAM, LPM or LSM units in their control sections of the LCS query the PAU every time before a corresponding proxy, hence Session is allowed to be initiated. These relations and interfaces have been omitted in Figure 4.17 for simplicity and presentation reasons, as well as due to the global relevance within the LCS. The setting and managing of respective policies will not be further discussed in this work since they do not help in explaining the LACBA location service principle.

4.6.5 Interface Components to Location Platforms

As shown in Figure 4.17 the *Location Platform Adaptor* (LPA) provides an important interface to the external LACBA location service entities, mapping the service requests and sensor data transmissions to respective standardized and readily available LP interface descriptions. Their main function is to provide translations from the internal generic LACBA, i.e. LP independent to the external LP dependent signaling requirements. The LPA units and their function will be taken for granted and not examined further in the scope of this work.

4.7. LACBA Signaling and Behavior

The signaling principle (also known as communication view) in LACBA relates to the information exchange between the LACBA control parts. IT also provides the basis for describing the system behavior in certain LACBA location service states. Modeling LACBA by the use of common OOD methodologies allows us to explain the system behavior in terms of sequence diagrams, which are common practice in software engineering development processes. The communication view modeling by use of sequence diagrams will exemplify the inter-actions of the various LACBA participants with the LACBA location service and shows the behavior of the latter in providing a “seamless” location service to service providers.

We will first illustrate the general control signaling and data flow principles in LACBA. Supporting these descriptions of establishing various sessions in LACBA, the messaging procedures, distribution of messages as well as data transmission concepts will be exemplified. Having shown the signaling principles, the basic functionality of LACBA in establishing user, LP and LS Sessions, as well as perform LP handovers will be described using UML sequence diagrams.

4.7.1 The Signaling Principle

The end-to-end information signaling between the various LACBA participants via the control areas can be seen in Figure 4.18. Here we can show the distinct control signaling paths for establishing *User*, *LP* and *LS Sessions*, and the data paths along which the user sensor data, location- and LP service information is exchanged (dark and thicker arrows). The sessions established for information exchange between the various participants include the corresponding data forwarding and adaptation control area components (i.e. proxies and adaptors).

As shown with the User *Access* and *Sensor Session* signaling, the respective transactions can pass through other control entities, whereby the messages can be modified accordingly by the control components. A User *Access Session* establishes a new *LP Session* instance, which in turn triggers the creation of a *Sensor Session*. The *LP Control* logic decides over the *LP Session* creation if candidate LPs can be determined from the client sensor stream. Similarly, the *LS Control* logic decides over the *LS Session* creation if the requested location information LQoS is available on a user. Since multiple LP relations (i.e. multiple Sensor and LP Sessions) can exist for a user, the LSP can switch between various LP Sessions, i.e. LPP entities dynamically as needed for a requesting service provider.

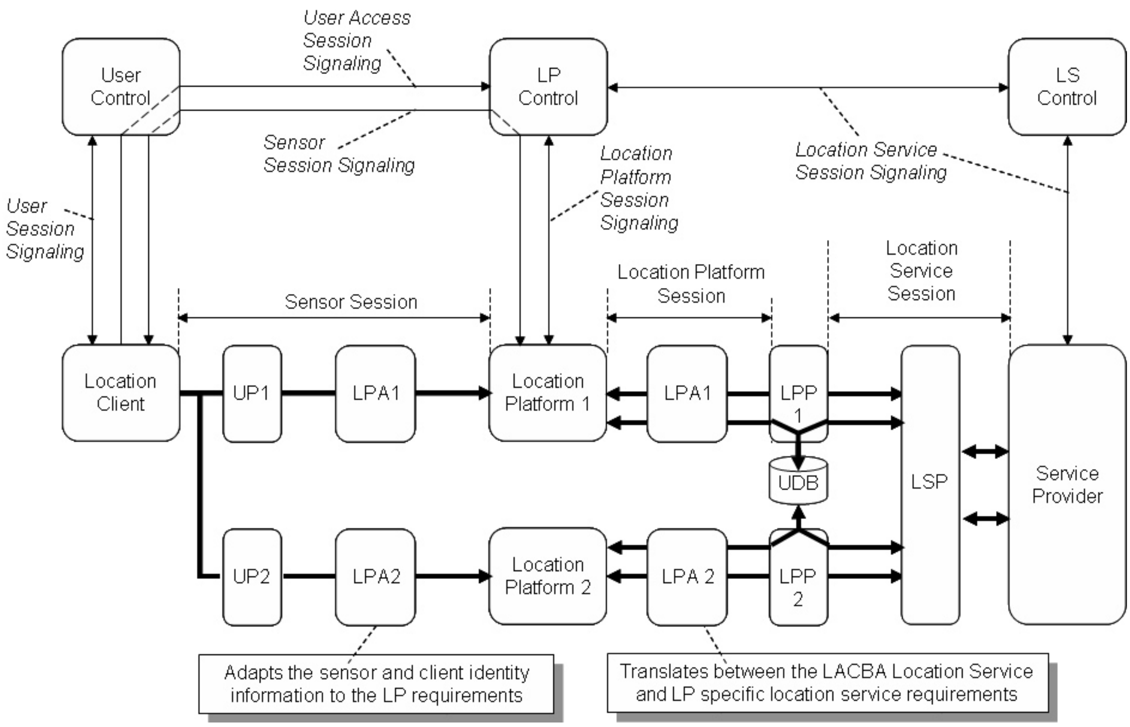


Figure 4.18 The signaling principle of LACBA

The communication between the LACBA location service control components is asynchronous via the exchange of messages. The messaging procedures involve a series of request/response/acknowledge transactions following the *Three-Phase Commit Protocol* (3PC) [Ske83]. It ensures non-blocking probability in that it allows all nodes in a distributed system agree to commit to a transaction, e.g. negotiating LQoS in the *LS Session* between the LACBA location service and the service provider, ensuring that a *Sensor Session* is established with a LP before the start of the LPP is instantiated, etc. This is also very useful when performing a LP search i.e. sending multiple requests to distributed LPs, processing the responses and then acknowledging the candidate systems.

4.7.2 Functionality of the LACBA Location Service

The functionality of LACBA location service and the inter-working between the various control parts i.e. respective components will be described in this section. Hereby, we will use the previously explained component view at certain stages. Simplified UML sequence diagrams will help to explain relevant communication principles.

The following illustration will describe the interaction of a user (User_A) with the LACBA location service, which in turn will determine a target LP (LP_1, e.g. providing mobile operator Cell-ID-based positioning) for his current location and establishing a *LP Session* with it. A service provider LBS (LBS_B) will establish a *LS Session* requesting the location information of a user with best possible LQoS at regular intervals. Another LP (LP_2, e.g. WLAN hotspot-based positioning) and *LP Session* for User_A will become available serving location information at better LQoS than LP_1. This will be signaled to LBS_B and a LP handover triggered.

It is assumed that User_A is a registered LACBA community user with a multimodal device having both WLAN and GSM/UMTS communication interfaces available. A constant communication link to LACBA is provided via GSM/UMTS all the time.

User Registration, Location Platform Discovery, and LP Session Establishment

Figure 4.19 illustrates the following procedure using the component view model and Figure 4.20 shows the registration of User_A as well as Sensor and LP Session establishment using a simplified sequence diagram:

1) The LACBA Client of User_A registers via the U API with the LACBA location service using his login name and password at the LAM. The User Registration Session has been now started.

2) The LAM checks the data in the UDB and verifies the access rights of User_A with the PAU.

3) Upon successful verification the user proxy (UP1) is instantiated and the corresponding registration reference returned to the client. The UP1 and UDB contain both the current registration as well as client configuration information.

4) The client can access available services via the UP1 in an *Access Session* or remain in the *Registration Session* to modify his personal information. Within the former, he now requests a LP Session and forwards his sensor data to the

UP1. Furthermore, User_A is now addressable by his unique ID (UID) within the LACBA location service.

5) The UP1 forwards the request and a snapshot of the current sensor data to the LPM, which performs the findLP service using the unique network identifiers (e.g. WLAN MAC or GSM Cell ID) and user identification information, e.g. (MSISDN). The latter type could be used to identify mobile operator-based LP.

6) The LP discovery is performed via the ERN

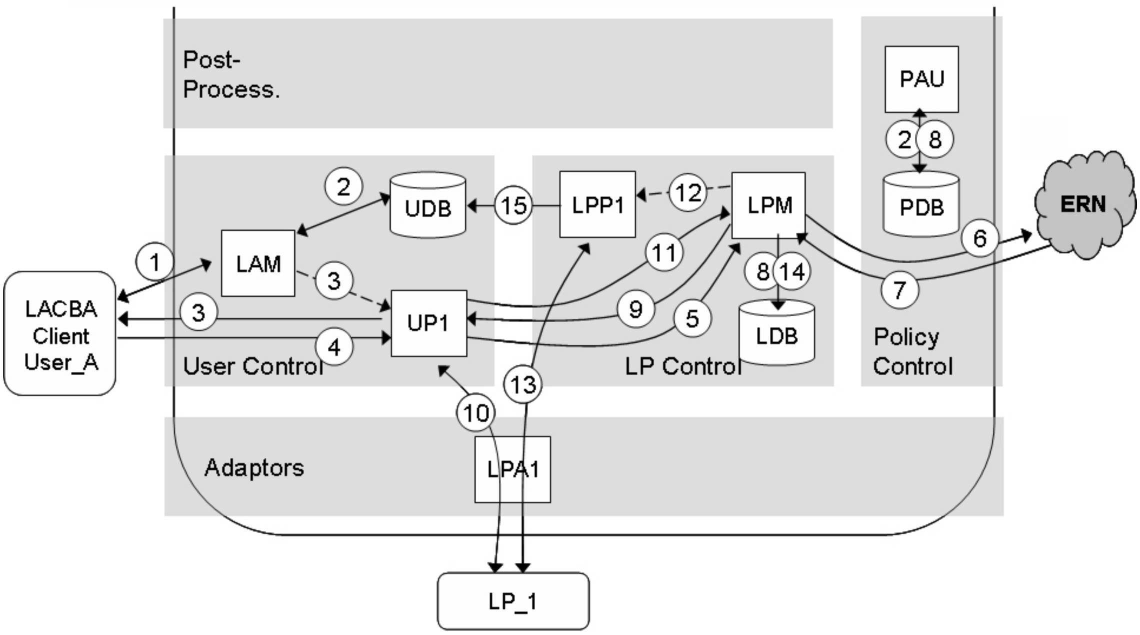


Figure 4.19 User registration and LP Session establishment in LACBA

7) Candidate LP(s) respond(s) being able to provide location information on the provided sensor/identification data are returned and aggregated at the LPM within a determined response timeout.

8) The candidate LP(s) response information is evaluated by the LPM and user access levels verified using the PAU. This enables to enforce access rights on private or LPs with restricted access (e.g. airport locations). Upon successful authentication access tokens for the user are generated and other relevant LP information is stored in the LDB (forms the LP Session Description). The LP is referenced by its unique ID (LID) within the LACBA location service.

9) The LP description includes a sensor session description, which together with the access token as well as LPA information (LPA1) is send back to UP1.

10) From this information the UP1 can register at the LP and establish the Sensor Session. Once established, the sensor data stream of the client is forwarded to the LP accordingly. The LPA1 filters and adapts the sensor data stream to the target LP requirements.

11) The LPM is notified by the UP1 once the Sensor Session has been established successfully. This ensures that LP can perform the positioning on the provided sensor data, which is a perquisite for the LP Session.

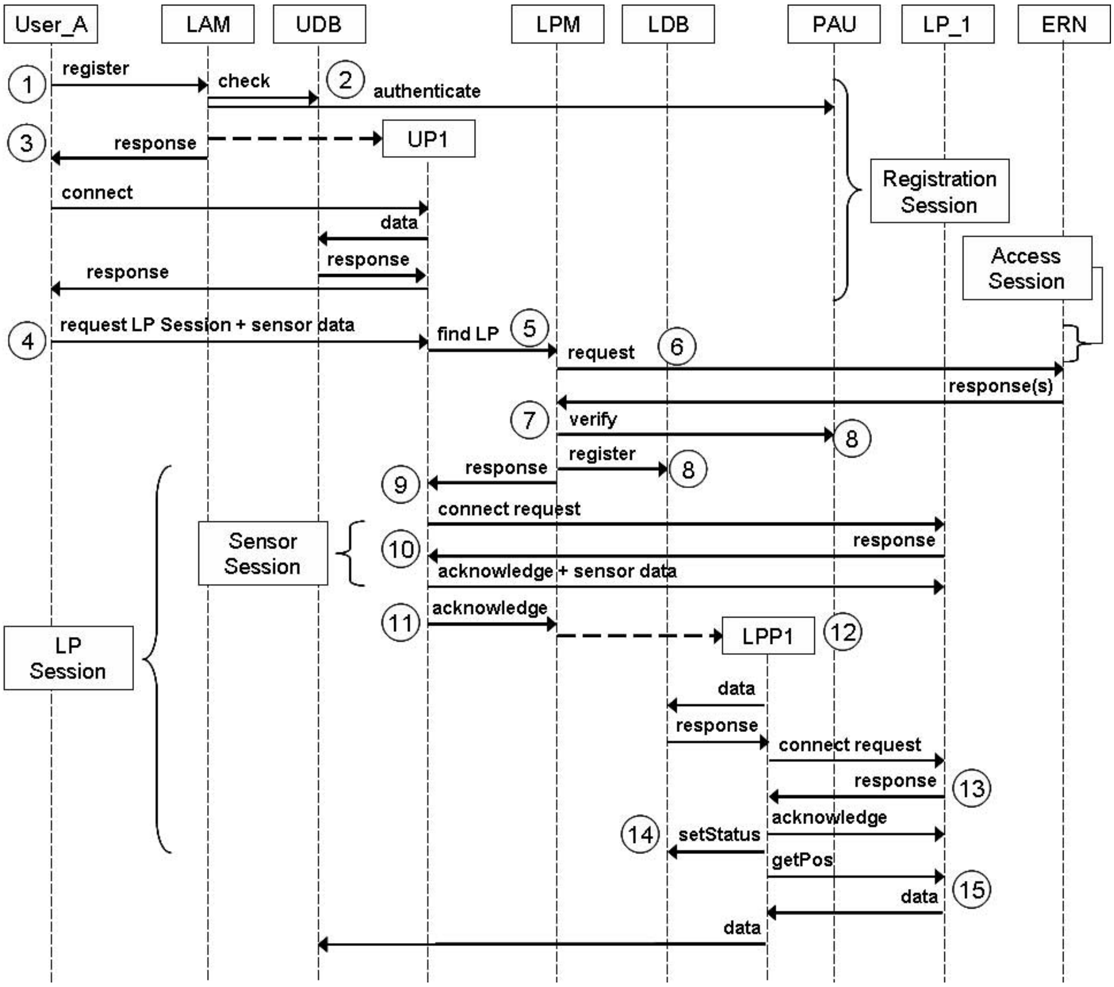


Figure 4.20 Simplified sequence diagram of user registration, Sensor and LP Session signaling

12) From the UP1 acknowledgement the LPP1 is instantiated from the LP description registered in the LDB. The LPP1 holds information and session parameters on accessing as well as providing all available location services from the respective LP. This includes references to both UDB and LDB profiles, in the former case also access to respective user location information.

13) The LPP1 initiates the LP Session between the LACBA location service (i.e. LCS) and LP, setting appropriate location update intervals. This regulates the getPos() queries to the LP which are adapted to the relevant LP service descriptions by the corresponding LPA1 entity as well. The LPA also adds additional LQoS information on the LP to the location information values and LQoS from the getPos() response.

14) The LPM registers the LPP1 status in the LDB marking LP_1 as "available" for requesting service providers.

15) Upon successful location information retrieval from the LP the corresponding location values and LQoS are entered into the appropriate user profile in the UDB

Provisioning of the LACBA Location Service

The LACBA location service provisioning to LBS involves three steps: the LACBA location service request, negotiation and actual provisioning. These are illustrated in Figure 4.21 and Figure 4.22 (sequence diagram). During the lifetime of a LS Session there can be a re-negotiation step if the currently available LQoS have changed compared to the registered LQoS requirements.

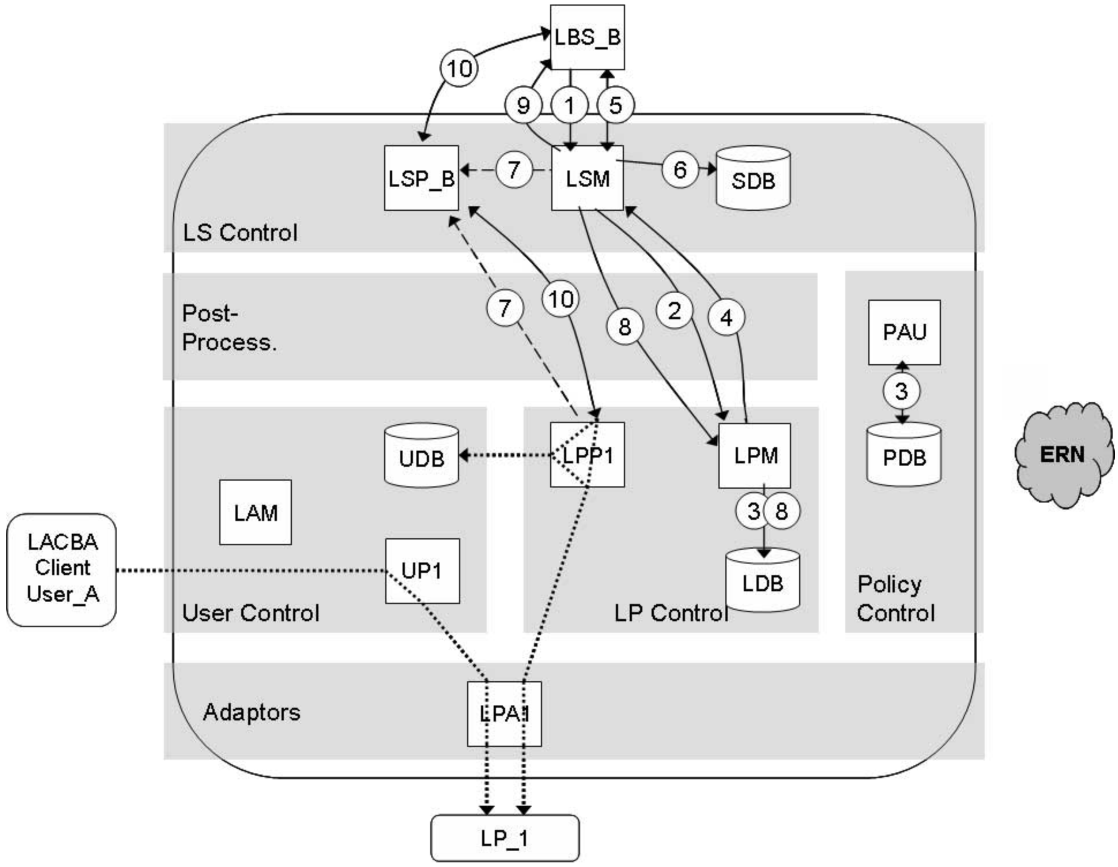


Figure 4.21 LS Session establishment and LACBA location service provisioning

1) A LBS (LBS_B) of a service provider requests the location information of User_A using his UID, specifying the LQoS level "high" via the LS API to the LSM.

2) In doing so the LS Session has been started and the LSM queries the LPM for currently available LP sources on User_A.

3) The LPM checks the registered *LP Session* for User_A in the LDB, verifying the access rights of the service provider on the LPs as well as User_A's privacy settings using the PAU. The required LQoS is evaluated from the corresponding LDB (relating to LP LQoS) and UDB (relating to the location information LQoS) profile on User_A.

4) The candidate LPP reference(s) with a current LQoS summary are returned to the LSM for the *LS Session* negotiation phase. In this case the only available LP is LP_1, hence provided reference to LPP1 is returned.

5) The LSM begins a negotiation phase with LBS_B. Since the requested LQoS level is “high” the LBS_B will acknowledge the first LQoS offer by the LSM.

6) The acknowledged LQoS offer is registered along with the SID of LBS_B in the SDB.

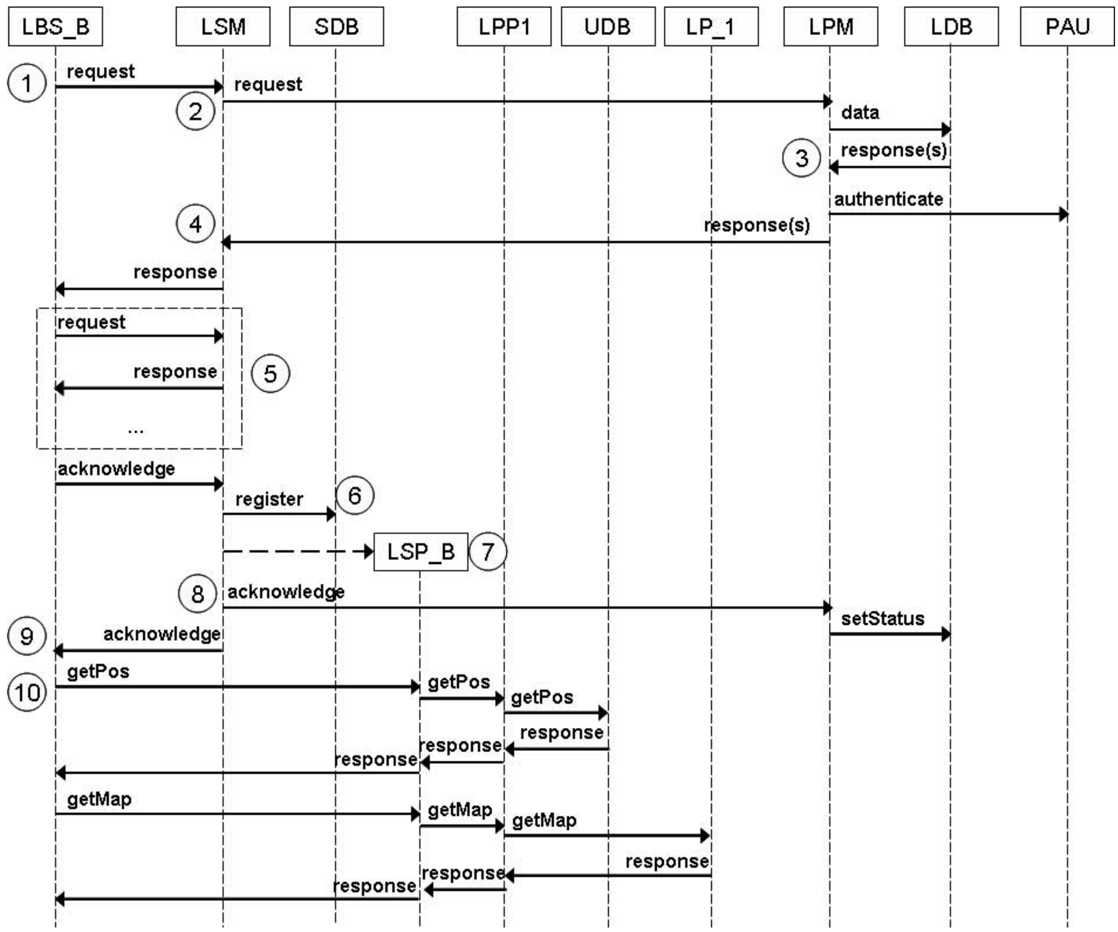


Figure 4.22 Simplified LS Session establishment sequence diagram

7) The LSM instantiates the LS Session proxy (LSP_B) with the chosen LPP1 access configuration and negotiated LQoS parameter information from the SDB. The latter is needed for potentially determining if the current location information LQoS has degraded and a re-negotiation or handover to a “better” LP source is required.

8) Once the LSP_B (hence LS Session) has been created the LSM acknowledges the LPP1 selection to the LPM. The LPM registers the LDB entry of LPP1 as “active” along with the SID of LBS_B.

9) The LSM passes the access reference of the LSP_B to the LBS_B indicating that the location information on User_A is available now

10) The LBS_B can now access the required location information and respective LP services via the LSP_B – LPP1 entities.

Registration of a second Location Platform

A LP handover in LACBA can be triggered in several ways: by a currently serving LSP entity identifying a degradation of LQoS from the current LP (e.g. degradation of sensor data quality: RSS dampening, GPS urban canyon problem etc.); by the LPM in case of a LP failure (e.g. the LACBA client leaves service area of current LP) or if a “better” LP than the one currently serving has become available. We will illustrate the latter case in our example as shown in Figure 4.23. Figure 4.24 shows the simple sequence diagram for the following LP discovery and registration procedure:

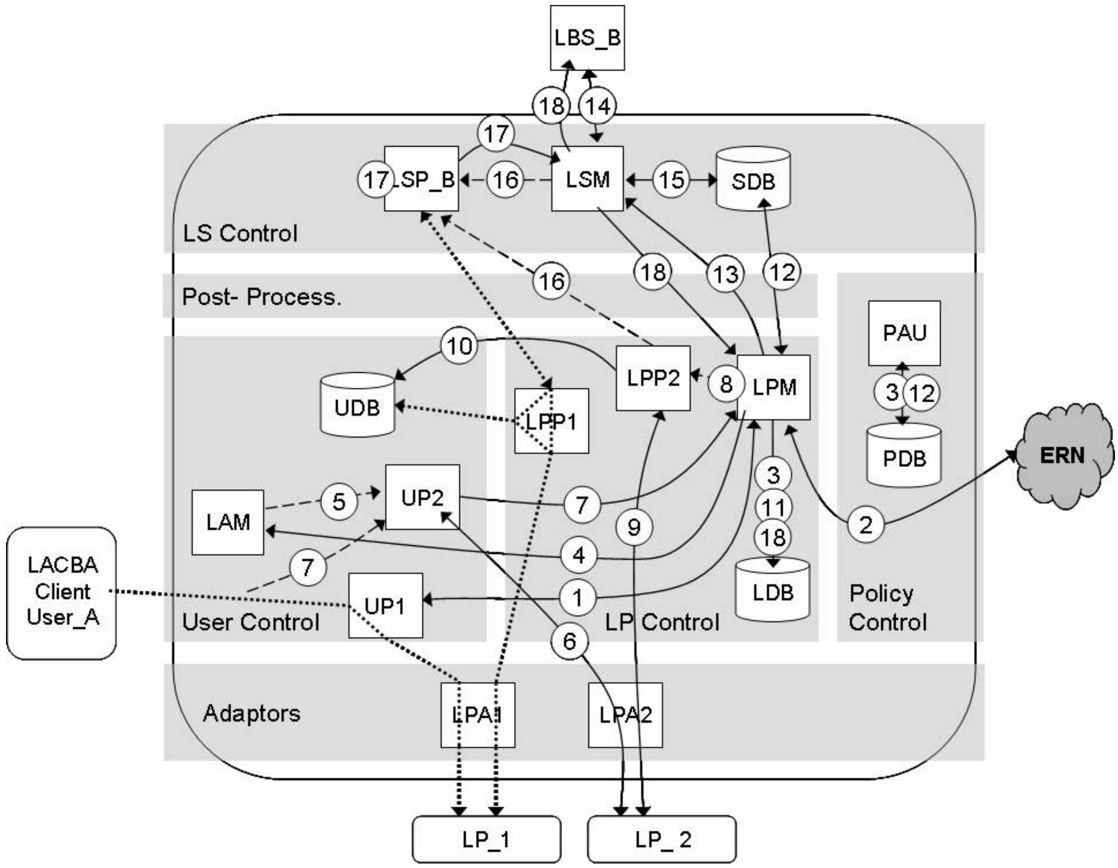


Figure 4.23 The LACBA location platform handover

1) The LPM takes sensor stream samples via user proxy entities at regular intervals. In this example the LPM obtains a sensor snapshot via UP1.

2) The LP discovery is performed via the ERN using the unique network and user identification information as explained earlier. The responses are collected within a set timeout period.

3) The LIDs of the LPs who have responded are compared against the registered ones relating to User_A in the LDB. If a new LID is found, it is verified against User_A’s access rights via the PAU. If the access rights are satisfied, the LPM will trigger the creation of a new *LP Session*.

4) The LPM sends a *Sensor Session* request to the LAM giving the UID of User_A as well as *Sensor Session* description of the newly discovered LP (LP_2). What follows is the standard Sensor and LP Session setup procedure as explained before [steps 5) to 11)]

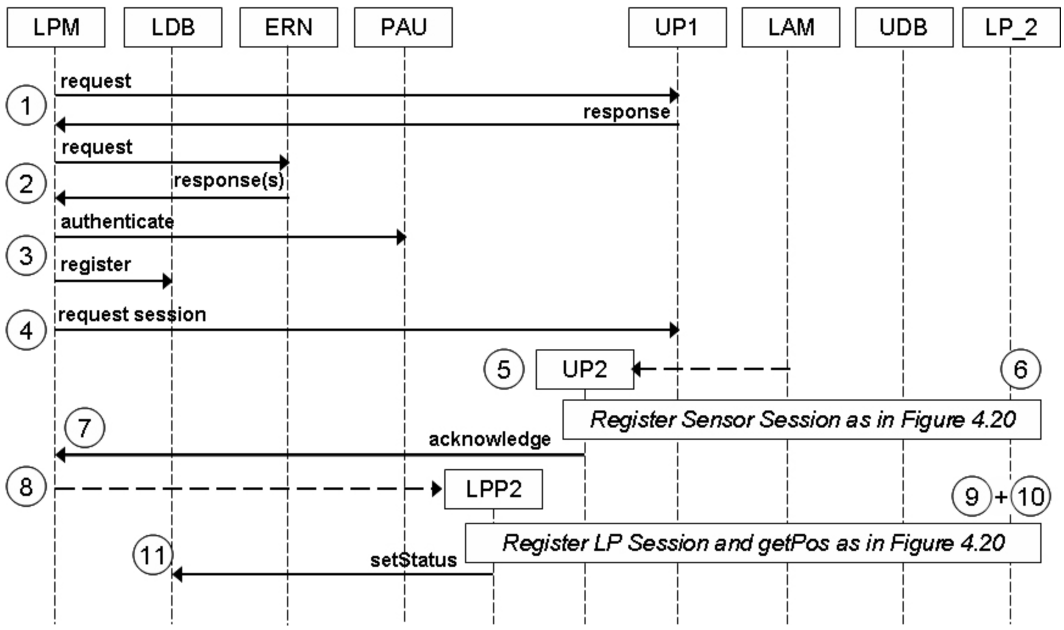


Figure 4.24 Simplified sequence diagram of the second LP discovery and registration

5) A respective user proxy (UP2) is created to forward the LACBA client sensor stream to LP_2 as well.

6) The *Sensor Session* is established via an appropriate LP adaptor (LPA2).

7) The LPM is notified upon *Sensor Session* establishment by the UP2

8) The LP proxy (LPP2) for LP_2 is started.

9) The *LP Session* with appropriate location update interval is established via LPA2.

10) Location information and LQoS generated by LP_2 on the user position are queried and stored in the corresponding user profile of User_A in the UDB.

11) The LPM registers the new LPP2 relationship with User_A in the LDB database and marks LP_2 as "available".

Performing the Location Platform Handover

12) In order to verify whether to perform a LP handover, the LPM verifies the LQoS of the newly available (LP_2) and currently active (LP_1) entries against the LQoS level registered for LBS_B in the SDB.

13) Since LP_2's LQoS parameters are better than LP_1's (GSM Cell ID compared to WLAN hotspot accuracy), the LPM forwards LPP2's reference and LQoS parameters to the LSM, requesting a *LS Session* update (LSP_B).

14) The LSM notifies the LBS_B of the newly available location information source. Since the LQoS configuration is set at "high", the LBS_B will not negotiate the new LQoS but acknowledge the new LQoS immediately.

15) During LP handover the access to LSP_B is momentarily blocked. The LSM updates the current SDB entry for LBS_B to the new (improved) LQoS configuration. This ensures that the currently agreed LQoS is always used in LP LQoS comparisons.

16) The new LPP2 session reference is added to the current LSP_B configuration where LPP1 is still actively used.

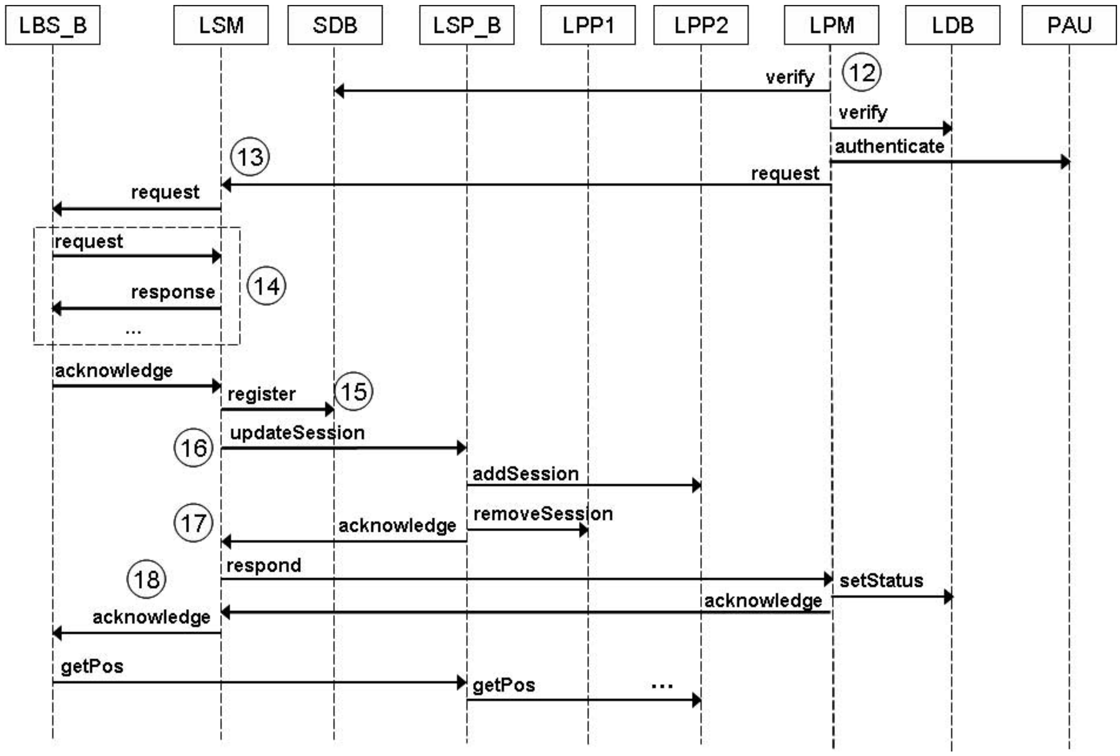


Figure 4.25 Simplified sequence diagram showing the location platform handover

17) The LSP_B switches the current LP Sessions internally (LPP1 → LPP2) and acknowledges this to the LSM upon completion.

18) The LSM signals the updated LSP_B description to the LBS_B indicating the new location information and service availability. An acknowledgment is sent back to the LPM, which updates the LDB entries by setting LPP1 back to "available" and LPP2 as "active" accordingly, which completes the LP handover. The complete handover procedure and LBS_B notification should occur within the agreed location update period. Figure 4.25 shows the handover signaling respectively.

4.8. Summary

In this chapter we have described the basic principles of LACBA and general functionality of the LACBA location service (i.e. LCS part). We have introduced the fundamental processes in LACBA governing the sensor information capturing from the environment, the collaborative generation of location information and seamless provisioning. From this, we have focused on describing the principles governing the handling of location information from multiple LPs and provisioning via the LACBA location service. Distributed system modeling approaches have been used to describe the LACBA system model based on several modeling views in order to fully grasp the service architecture in terms of software engineering concepts. The system structure as well as external/internal interfaces have been described using business, session and component modeling views. Three inter-working control parts have been defined for handling the User, LP, and LACBA location service issues. We have illustrated the data structures and relations between the control parts using the information view, and exemplified how the data was exchanged between the respective components using the communication view model.

Chapter 5
LACBA Usage Behavior Modeling and Simulation

The following chapter will focus on the novel initialization and location information generation part of LACBA. As part of realizing the special features of a community-based architecture as set out in section 2.5 we first model the collaborative information contribution process of the users within a community, defining the respective boundary conditions for such an approach to work in particular usage environments. Subsequent chapters will focus on the generation of location information from multiple sensor streams, evaluating the positioning performance i.e. LQoS.

Building upon the community-based location service description introduced in section 2.4.4 we will further define the LACBA community model discussing contribution methods and user -roles leading to a (hopefully) stable location service for various usage environments. The model will be completed by specifying environmental, technology infrastructure, user and community-based factors which influence the location service building process from a community point of view. Choosing suitable usage scenarios, we will perform simulations on the community-based location service usage behavior given the respective modeling factors determining initial user community sizes in a given usage area over a certain period of time. A particular focus will be on user contributions, satisfaction levels and completeness of the community data pool on the respective usage area as discussed in section 2.4.4.

Although the scenarios simulated more or less present idealized models of real world applications, they can be taken as viable guidelines for general trends and interdependencies. Empirical results from test drives within the inner-city of Munich have been used to verify some of the established community model parameters.

5.1. The LACBA Community Model

One of the requirements set out in building a community-based/supported global location service architecture was that the LP provided and community in a particular usage environment are responsible for the local location service building, maintenance, and provisioning of respective location information. Hereby, the LCS has the sole role of ensuring that the correct LP is discovered and addressed by each community member. Figure 5.1 exemplifies the different user contribution methods and input channels which can be provided to community members. All user types (1) can provide sensor information automatically discovered from their LACBA enabled clients via the LCS during LBS usage or actively when in initialization mode. The first case is an example of non-intentional value creation where the perceived value to the user is the successful localization and execution of the LBS. The latter case is where a user actively contributes without immediate personal visible benefit. It is also possible to allow users to submit additional location information manually via appropriate client application interfaces by this method as well.

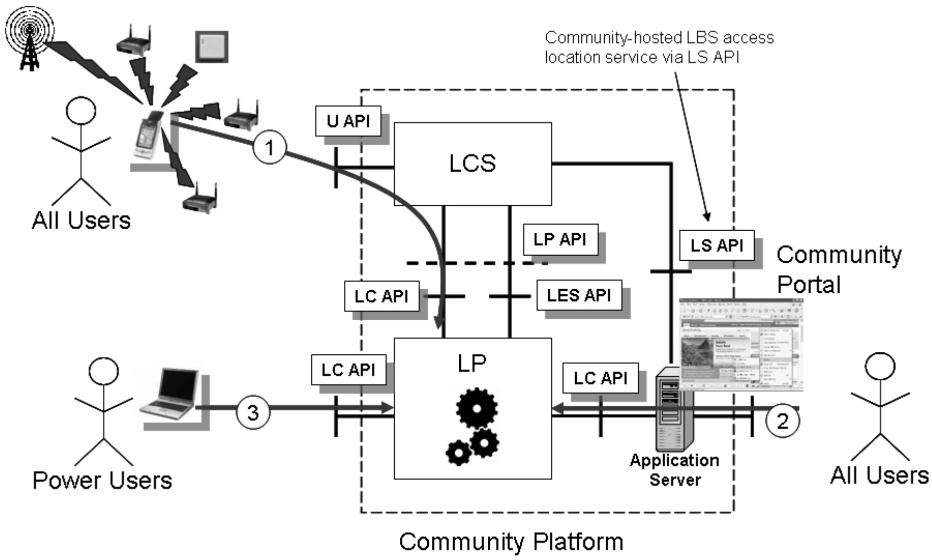


Figure 5.1 User contribution methods for location service building

There are two LACBA independent contribution methods which the LP can additionally provide. According to the three-layer community model proposed earlier (section 2.4.1) a community usually consists of a platform (layer 1) representing the "visible product" of the community, e.g. a website or community portal on the mobile device showing the kinds of community-enabled LBS services and location information provided/discovered from the environment. Hence, the users should be able to contribute location content and information (e.g. geocoded WLAN hotspots) manually via this platform (2). The user's incentives to contribute can also be explicitly communicated and managed here as well. The immediate incentive is the availability of LBS which would not be possible without user contributions. Other incentives can be user ranking schemes and high scores on number of contributions, etc. The third and last contribution method (3) is by LP administrators or system engineers (i.e. power users relating to the LACBA use case model definition). They have the main task of contributing location information within the community, i.e. by training of LPs by generating location references for the location service following certain initialization procedures, which will be elaborated on later in this section. The approach taken in (3) is common to all positioning system initialization procedures involving training by location reference generation. The clear distinction between (1) and (3) is that in the latter case, sensor data capturing and contribution does not necessarily happen at the same instant. Furthermore, (1) involves small incremental, whereas (3) exemplifies usually large sensor data volume contributions.

Layer 2 in the three-layer community model specifies different communication spaces provided by the community platform e.g. different contribution channels and methods via LACBA clients, HTTP web interface or other upload mechanisms as represented in (3). Other services and LBS enabled by the community-based location service provided by the platform can also be seen as communication spaces. Examples of this are location-based dating, child-finder or meeting-management services. The respective sub-communities forming around these particular services represent layer 3 in the proposed community model. Depending on the contribution approaches and strategy chosen layer 3 sub-groups can also relate to location service building process in that there are distinct community sub-groups only contributing location information, and other groups mainly using the location service but only contributing little or nothing at all. This will become more apparent in the following sections.

Every community user perceives the community platform as a monolithic system of which the “visible” part is only the community website or portal provided via the application server. The workings of the LCS and LP are hidden to all users, but respective interfaces provided as needed. Nevertheless, when modeling the contribution processes of the whole community platform for the usage environment, all user contribution types of the respective community have to be considered. In terms of the LACBA principle the automated, client-based contributions provided via (1) clearly outweigh the manual contributions via (2) and (3) in terms of number of users and contribution effort (i.e. the contribution effort via (1) is much lower), hence dictate the contribution modeling behavior in community-based systems. Bulk uploads such as depicted in (3) are more suited in smaller usage environments (e.g. offices) when fewer people are involved in the building and optimization processes.

5.1.1 Location Service Building Strategies

The user type as well as integration and participation in a community collaboratively enabling a location service can be different depending on the nature and infrastructure of the respective usage environment. As already depicted in Table 2.2 of section 2.4.4, initializing a location service in a fairly static office environment requires only very few people with knowledge on the available infrastructure and surroundings (i.e. administrators/engineers performing site mapping). This is usually how WLAN RSS-based positioning solutions are implemented [Eka06]. On the other hand, implementing a location service in a dynamic inner-city environment requires multiple users most suitably using a community model. Therefore, we can summarize possible user integration and contribution strategies as shown in Table 5.1 having different impacts on signalling overhead as well as general system performance needing to be considered.

Restricting the number of users actually contributing to the build-up of a location service is often only meaningful if the respective usage environment is fairly small in size and consists of a more or less static infrastructure including wireless networks present (e.g. offices). In larger environments such as inner-cities with fluctuating infrastructures especially in the case of small wireless WLAN hotspot networks, restricting the number of users reduces the probability of successfully initializing the entire usage environment in time to ensure an acceptable location service availability and quality. In community terms a critical mass of users may never be reached in time and more users leaving, eventually resulting in the death of the community service. However, having an additional sub-group within the community contributing all the time (at least on average much more than the bulk of the community) by being paid or intrinsically motivated (e.g. Linus type users) may result in an improved location

Strategy Type	Users Contributing	Contribution frequency	Signalling overhead	Typical Usage Environment
1	All users	All the time	Very large	Outdoor, dynamic inner-city environments with fluctuating wireless infrastructure, e.g. residential areas
2	All users	Selected times	Large	Outdoor, fairly static inner-city environments, e.g. commercial or industrial areas
3	Selected users	All the time	Mediocre	Small outdoor or large indoor infrastructures, e.g. airports, hospitals
4	Selected users	Selected times	Little	Indoor environments with fairly static wireless infrastructures, e.g. offices

Table 5.1 Location service building strategies

service stability and quality, or even reduce the contribution effort experienced by the overall community members. The contribution frequency and effort required by the individual members is crucial to the acceptance (i.e. satisfaction) of the provided location service. Automated contributions via LACBA enabled clients by every user has the least contribution effort and provides the fastest location service buildup but high costs in terms of network signaling. The contribution effort and cost can be optimized towards the user by having him only contribute to the location service buildup during LBS usage. Further optimization can be employed by using dead reckoning location update protocols or data compression techniques in order to reduce signaling overhead and cost even further [Leo03]. Apart from the continuous, small, incremental location contributions single-event bulk contribution uploads of location data logs collected over the duration of up to day for the location service buildup process are plausible as well. This is quite common in type 3 and 4 location building strategies e.g. indoor WLAN positioning system deployment but also thinkable as an addition in type 1 and 2 cases.

In every case, employing manual contribution effort in addition to automated contributions or on its own results in an increase of frustration incurred by each user. Manual contributions and location corrections by the user could be essential in the early phase of a community location service due to a lack of reference locations available in the community database. Although manual contributions should be minimized, they are important feedback mechanisms in the event of a localization failure at an early stage further improving the quality of the location service.

Types 1 to 4 represent general location service building strategies yet there are no hard boundaries between them. Principles incorporated by types 2 and 3 could be also applied additionally in type 1 and 2 location building strategies. Types 1 and 2 relate closely to the approach developed by this work, hence both will be applied in our further quantitative analysis since they enable us to explore the boundary conditions of community-based location service building more closely.

The most likely contribution strategy for the majority of users is during the actual LBS usage. As discussed in section 2.4.3, the majority of users (i.e. Bills) have a balance between contributing whilst equally being able to use the location service in order to obtain desired LBS. Thus the *average contributions* and efforts incurred by each user in the community should be treated equally when modeling the overall location service building and optimization process.

5.1.2 Location Service Development Model

Coming from traditional positioning systems requiring extensive training effort, the location service development usually follows set (re-)initialization and positioning phases. Applying the community-model to our location service development as described in section 2.4.2 introduces the collaborative, continuous, self-optimization process of the location service after an initial location reference data pool buildup by the user community. According to the principles set out by [DH03], the community-based location service development and relationship between location service usage, required user contribution effort and applied system heuristics expressed in LACBA can be presented as shown in Figure 5.2. The user community and thus usage probability of the community location service follows the tipping-curve behavior (see section 2.4.2) experiencing exponential growth after a critical mass of users is reached and reaching saturation once sufficient data is captured from the environment.

Apart from the automated contributions users may be required to provide additional information manually if the user's current location cannot be determined from the provided location data. The user is not aware of the contribution effort of automatically providing location data discovered by the client during LBS usage (i.e. the non-intended value creation process), yet may exhibit localization failures at the beginning of the location service buildup and may be required to contribute additional manual location corrections (i.e. intended co-value creation). As more and more of the usage environment (e.g. wireless networks) is discovered the necessity of required user contributions (i.e. user interaction) is gradually reduced, but always needed over the lifetime of a community-based location service.

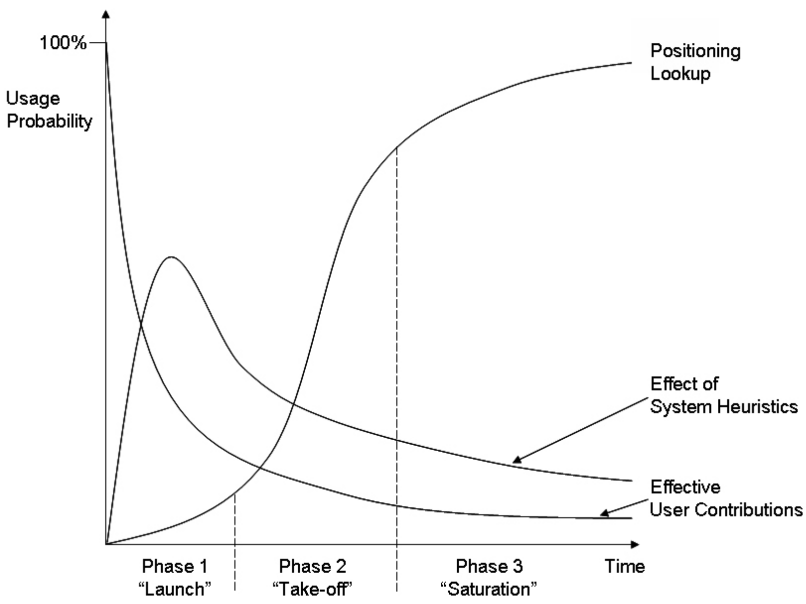


Figure 5.2 Location service development adapted in LACBA

System heuristics supporting the location reference data generation, improving the location service provisioning when only limited reference locations are currently available in the community database can be applied e.g. interpolation, nearest neighbor lookup functions [BB03], etc. The usage pattern depicts that the effectiveness of such methods is limited when only little to none location reference data is available in the community database. As more and more location data is generated by users, the system heuristics will eventually kick-in and accelerate the location service buildup as well as improve initial location service quality. The positive contribution effects provided by these system heuristics will gradually decrease as more "real" location data references are generated by the users and thus the location services' localization methods provide better positioning results. Over the lifetime of the location service however, such system heuristics will continuously help adapting the location service to changes in the environment e.g. wireless networks which have been removed can be discovered by nearest neighbor mechanisms. An analysis on the impact of system heuristics used will be covered in chapter 6 later on.

The established community-based location service model and building strategies set out in the previous section can be used as a basis for a further quantitative analysis on evaluating its application to certain usage environments and scenarios, establishing boundary conditions on initial critical user mass on respective usage environment properties (i.e. environment size, coverage and evolution of networks, etc.).

5.2. Determination of the Modeling Factors

In order to evaluate the community-based location service principle by using and generating location information from the environment as realistically as possible, we have to look at the various influencing factors having possible affects on the proposed concept. In general the factors can be divided into three main categories: environmental, technology and user-based factors. Some of these have already been briefly introduced as potential error sources on defining the positioning system model in section 2.2.3. These factors can be again subdivided into location provisioning and quality related aspects, the former of which will be of only interest for the quantitative analysis of this chapter. Due to the different kinds of community natures, motivations, and human aspects involved in community processes, it is important to stress that the following analysis represents an idealized model of the real world keeping complexity as low as possible.

5.2.1 Environmental and Technology Related Factors

For analyzing the LACBA community model in a particular usage environment, we assume a certain amount of *community users* N at time t for a particular quadratic area A. The area itself is partitioned in equally sized squares of area M, where each square represents an idealized network cell in a cellular (e.g. GSM) or wireless hotspot network (e.g. WLAN). The *average cell radius* will be denoted by R, such that the network cell area M will be approximated to:

$$M = 4R^2 \tag{5.1}$$

GSM coverage can be virtually guaranteed in populated areas where more or less regular size structures are available in city locations. Real cellular network cells have more complex polynomial-type structures and their boundaries are less sharply defined i.e. are often overlapping and using signal strength thresholds. The existence of larger umbrella cells complicates matters slightly further, but for the sake of simplicity we can approximate the average network cell structures using our square-based cell model. Similarly by varying the radius of the network cell denoted by R, we will simulate the size and coverage of WLAN hotspot cells typically encountered within inner-cities of major cites worldwide. Test drives in Munich and other cities have shown more than 75-80% coverage of WLAN hotspots within typical inner-city locations. Many providers aim at similar coverage as GSM cellular networks in densely populated cities within the near future [Var06].

In this model every user is unambiguously located within one of these network cells. The *initial number* of users *I* which are randomly distributed across the usage environment area *A* can be set at the beginning of each simulation run.

Each cell has the property *locations* denoted by the letter *l* which counts how many locations have been mapped to the cell's unique cell ID, ESSID, or MAC address. If a cell is mapped to at least one location the model assumes that a location service request in this cell can be satisfied. The more contributions are made relating to that cell the higher the locations count becomes, which could be used as a "quality" indicator in later evaluations.

The cellular network environment is modeled by the coverage of each cell and the number of hours *h* after which the community data pool is degenerated to simulate a change in the network infrastructure, e.g. WLAN hotspot removed or newly set up in the environment. The value of *l* for a particular cell is thus reset to zero meaning that this cell has yet to be (re-)discovered by the community. The rate at when this degradation occurs i.e. the *network evolution* factor *E* is also set at the beginning of a simulation and can thus be used to model the network dynamics in different usage environments, e.g. static office or dynamic inner-city urban scenarios. The network evolution factor can also be used as an indicator to mimic areas of no network coverage in the environment, i.e. a user enters a location denoted by the area *M* for which no contributions yet exists in the community data pool and thus the location service request is failure. Hence 75-85% coverage of WLAN hotspot location encountered in current inner-city locations can be modeled by varying *E*, thus can be considered in later usage environment scenario analysis.

5.2.2 User Behavior and Contribution Factors

A critical factor when modeling communities for a certain product or service is to determine the *theoretical addressable maximum target group size* denoted by *T* for a particular environment. In economic terms this factor relates to the potential market size, where the target group is also identified. When modeling the growth of corresponding communities, the tipping curve behavior (i.e. saturation) will occur towards this theoretical maximum.

Looking at the individual user factors, every member in the community model has the properties *requests*, *contributions*, *satisfactions,* and *frustrations*. The request indicator *r* counts how frequently the user requested a location service, and the contribution counter *c* indicates the number of location references he has contributed to the data pool. Satisfaction *s* measures the satisfaction incurred by each user by incrementing the current value each time the user requested a location service and was automatically provided with the location service on automated contributions (i.e. non-intended value creation). On the contrary, the frustration *f* counts the situations when the user requested a location service and had to provide the current location reference manually.

The user factors determining the user behavior in the LACBA community model can be best described by using the control flow model as depicted in Figure 5.3. After the appropriate environment variables and global user parameters have been set as well as initial users of the community defined by *I* randomly distributed on the environment space, each individual user behaves autonomously on the start of the simulation. Along the lines of three distinct states the user's tasks can be defined as follows: 1) Use the location service and contribute to the community data pool, 2) recruit new users, and 3) move about in the usage environment. At every simulation time step the probability of carrying out each task is computed for every community member accordingly.

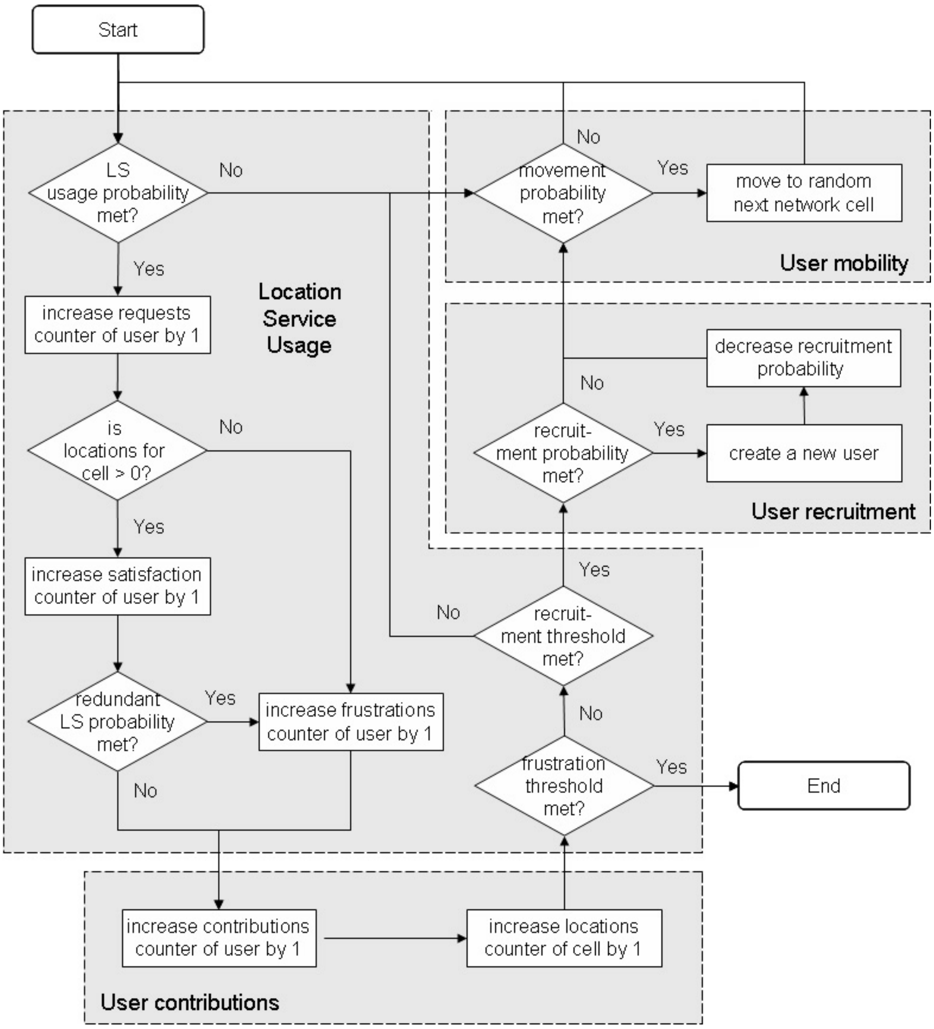


Figure 5.3 The LACBA community control flow used in the simulation

Location Service Usage and User Contributions

The probability term p_{ls} represents the *location service usage probability* which is globally set the same for all users at the start of the simulation. It determines whether or not a user actually uses a location service whilst residing in a particular cell. The average location service usage of users within the community per week is specified by *U*. Furthermore, the probability of using the location service will increase proportionally to the incurred satisfaction (i.e. willingness of the user to use the service grows). If the user requests a location service, he will either be satisfied (i.e. the value *s* incremented by one) if the cell has been already mapped in the community data pool (i.e. the locations value for that cell is not zero) or frustrated if it hasn't (i.e. the value of f incremented by one if the location value for the cell is zero). In the latter case the user will be forced to contribute the current location manually to the community data pool.

In addition to the manual contributions upon location service request failure the system can ask the user to contribute although a location mapping for the current cell he resides in is present in the community data pool (i.e. the locations indicator for the respective cell is greater than zero). This *redundant location service request* is performed at a probability value p_{rls} on every successful location service request carried out by the user. The redundant request can be used to maintain the integrity of the community data pool and simulate possible localization errors, i.e. a location information quality degradation aspect which could also result in a frustration of a user.

If the value of frustrations outgrows the value of satisfactions by a pre-defined number set by the *discouragement threshold*, also labeled as d_t, the user will stop using the system and be removed from the community.

$$(f - s) \geq d_t \tag{5.2}$$

If the value of satisfaction outgrows the value of frustrations by the pre-defined *recruitment threshold* indicated by r_t, the user will recruit a new user with a probability set by p_r i.e. *probability of recruitment*.

$$(s - f) \geq r_t \tag{5.3}$$

New User Recruitment

Upon successfully computing p_r a new user is spawned in the simulation following the same autonomous behavior as the rest of the community members. On that event at time = t, p_r is decreased by a multiplication factor defined by the maximum addressable target group size and current user community size, i.e.:

$$p_r(t) = p_r(t-1) * \left(1 - \frac{N}{T}\right) \tag{5.4}$$

This results in the natural saturation effect incurred by a community, i.e. it will be harder to recruit new users of a maximum addressable target group the more the community grows unless further network externalities such as product hypes potentially take effect (see section 2.4.2).

User Mobility

A user moves randomly within the quadratic area A at a speed defined by the *mobility* factor m moving from one network cell to another. A mobility probability p_m is computed at every simulation turn indicating whether the user remains in the current cell or moves on to the next. The mobility probability at each simulation step (one hour) depends on the network cell radius and speed of the user thus can be derived as follows:

$$p_m = \frac{m}{R} \tag{5.5}$$

5.2.3 Summary of Relevant Factors

A summary of the important modeling factors and units necessary for the following simulations of different usage environments is given in appendix C1. The most crucial factors which will define the community development relating to the usage environment will be the size of the environment area and network cells available within, as well as network evolution rate relating to environment dynamics. Concerning the user factors the usage behavior of the users within the community is mainly influenced by the average number of times per week they actually request a location service and by their mobility specifying how frequently they move between cells. In turn the mobility depends on the size of the cells in the environment. Both types of factors will have an impact on the minimum required initial user base for respective usage environments ensuring a fast enough establishment of the location service and survival of the community under the corresponding environment as well as usage conditions. Relating to the tipping curve formula (2.1) from section 2.4.2, the rate of growth of the community as defined by β is greatly determined by all of these factors and balance/tradeoff between some of them as will be seen in later simulations, e.g. initial user number vs. environment and network size vs. user mobility etc.

5.3. Simulation of the LACBA Community Model

This LACBA community control flow model as represented in Figure 5.3 was implemented using the StarLogo 2.0[7] simulator developed at the Media Lab at the Massachusetts Institute of Technology. StarLogo is a programmable modeling environment for analyzing the workings of decentralized systems, e.g. market economies, traffic jams, virus spreading, etc. The StarLogo source code can be found in appendix C2.

For each environment type chosen for simulation the following criteria will be analyzed in order to verify the plausibility of LACBA community model:

1) Given a particular usage environment having area *A*, evolution factor *E* and cell radius *R*, what will be the initial required user base *I* at start-up in order for the location service to stabilize and the community to survive?
2) Similar to 1) but keeping the initial user base fixed, what effect does a change in average usage per week (*U*) have? For example, half the initial users but double the average usage per week?
3) Especially relating to the evolution of WLAN hotspot networks within inner-cities, what is the maximum rate of change (*E*) tolerable in the environment in order for a location service to function and hence community to survive? This will give an indication on the possible (re-)initialization strategies and learning/adaptation quality required by the system. Furthermore, by varying the evolution rate we can explore network coverage gaps in the environment and their effect on the community-based location service.

Points 1) and 2) will especially aim to discover the boundary conditions determining also the respective tipping point (i.e. where positive feedback occurs) and time taken

[7] StarLogo 2.0: http://education.mit.edu/starlogo/

until sufficient data on the environment has been captured in the community data pool (i.e. analogous to initialization time of positioning systems) as was implicated earlier.

Although LACBA allows for heterogeneous positioning, we will restrict the environment simulations to one location technology type in every case, since this does not directly influence the community behavior modeling i.e. a common fixed area size for *C* will be initial set. Having a single positioning technology available also represents a worst case scenario.

The simulation period will be over several weeks to allow for a comprehensive overview of the LACBA launch, take-off and saturation periods. For every one hour time step (simulation instant) the control flow shown in Figure 5.3 and respective parameters will be computed for every user in the community at that moment. The simulations for each scenario and change of selected initial variable have been carried out 100 times each in order to ensure an empirical sound representation of the observed variables (on a Pentium 4 running at 2.4 GHz using a Microsoft Windows XP operating system). Apart from the global parameters set for the environment and the user behavior every user will have its own satisfaction, location service requests, frustrations, and contributions counter.

5.3.1 Descriptions of Environment and User Behavior for Simulation

For the collaborative community-based location service approach the most suitable scenario is provided by a typical outdoor, inner-city (urban) usage environment for WLAN-hotspot- and GSM Cell-ID-based localization. This environment provides very high densities and coverage of respective network cells, density of users living in the environment, and dynamics relating to network evolution rates.

A rural scenario is less suited as a simulation scenario since GSM cells for Cell-ID-based localization are several kilometers in radius and satellite-based positioning systems outperform network-based localization in such an environment by far. Furthermore, WLAN-hotspots are -if at all- sparsely available and user densities are too little. Concerning user mobility, due to the larger size of the network cells, users are much less likely to change network cells and potentially remain in one network cell for too long. The small office- or factory-based scenario is also unsuited for effectively analyzing a community-based location service scenario. As mentioned before due to the fairly static and small usage environment having small user groups or single service technicians (re-)initialize the provided location service at selected times should be the most sufficient in such cases.

Although the inner-city/urban simulation used represents an idealized model, the simulation model has been closely mapped to real world environment settings where possible. For establishing the basic environment model test drives from the inner-city of Munich as well as global statistical data from official government have been used [Muc06]. A summary of the simulation parameters for the respective usage environment situation is given in Table 5.2.

In the urban inner-city scenario, the simulation covers a quadratic area of 32km² which approximates the area covered within the city of Munich (see appendix section C3). The average cell radius has been set to 250m for GSM network cells using the following statistical tool[8], and 35m for WLAN hotspot networks (see analysis in chapter 6 later on). This assumes 100% network coverage in both cases which could be true for GSM but not for the WLAN-hotspot case.

For the WLAN hotspot scenario a network evolution factor of 168 has been chosen, meaning that every 168 hours (i.e. on average once a week) a network cell will be reset. In order to determine this value regular test drives along a fixed route over the duration of one year have been carried out in the city Munich [Ipp06], being cross checked with similar urban environment scenarios [LaM05].

Modeling factor	**Urban inner-city environment**	
	GSM	**WLAN-hotspots**
Environment area (*A*)	32km²	
Average cell radius (*R*)	250m	35m
Idealized cell area (*M*)	250.000m²	4900m²
Network evolution factor (*E*)	Zero	168 (i.e. cell failure on average once a week)
Maximum addressable target group size (*T*)	760	
Average usage per week (*U*)	7 (i.e. on average once every day)	
Initial location service usage probability (p_{ls})	4,2% per hour	
Redundant location service request probability (p_{rls})	10%	
Recruitment threshold (r_t)	3	
Discouragement threshold (d_t)	3	
Initial probability of recruitment (p_r)	10%	
Average mobility (*m*)	75m/h	
Mobility probability (p_m)	30%	100%

Table 5.2 Summary of simulation parameters for LACBA scenario

The maximum addressable target group size of the observed usage environment area has been estimated to be roughly 760 potential community members. A detailed step-by-step derivation and reasoning of this value can be found in appendix section C3.

The average usage of the locations service has as been set to 7 times a week which corresponds to one usage per day. This is a reasonable estimate especially when curiosity and motivation are still high at the beginning. Therefore the usage probability for every simulation hour is 4,2%. Due to location measurement errors, signal fading and other sporadic errors incurred during location service usage, a 10% redundant location service request probability has been chosen meaning that the location service has an average one in ten probability in malfunctioning despite the presence of appropriate mapped locations in the community data pool.

Every user in the community accepts on average three more malfunctions of the location service (i.e. having to contribute his location manually) than successful automated localization cases using the automatically provided sensor data from the users LACBA client, before leaving the community. On the other hand, a user needs at least three more successful localizations using the community location service on automatically rather than manually provided sensor data in order to start recruiting new users to the community. When recruiting a new user there is only a 10% chance that this user will actually join and participate in the community. These global variables have been chosen to simulate the majority of “neutral” users (i.e. Bills) and average contribution efforts incurred in a community (see section 2.4.3).

[8] Member of senderliste.de: http://gsm.yz.to/karte.php

Most people spend several hours a day in the same building where only a few cell changes occur. Otherwise people cover long distances when going to work and back (i.e. resulting in potentially many cell changes). Thus the average mobility of users in the simulation has been set to 75m per hour. In the case of the WLAN hotspot scenario this results in a 100% mobility probability that users change network cells every hour but only in a 30% chance in the GSM case due to the much larger cell radius.

5.3.2 Varying Initial User Count on Location Service Deployment

The first part of the analysis is to determine the required initial user community base in order for the location service successfully go from the launch to the take-off phase where it exhibits positive feedback and exponential growth occurs. Hereby the initial user variable *I* is the only parameter varied on the simulation setup. The first simulation batch is carried out where *I* = 50, every subsequent batch run incremented by 10 users.

At every simulation run the following occurrences are registered as well:

- Average number of community users lost before positive feedback occurs
- Time taken to first map 95% of the network cells in the community data pool
- Time taken where the average user satisfaction exceeds the average user frustration (point of positive feedback) and respective percentage of mapped network cells.
- The tipping point where exponential user growth occurs and respective percentage of mapped network cells i.e. $\bar{s} - \bar{f} \geq r_t$

Figure 5.4 and Figure 5.5 show the simulation results for the WLAN hotspot and GSM urban inner-city scenario respectively. In both cases, the number of users currently in the community at a particular point in time are recorded over the duration of the simulation for *I* = 50, 75, 100, 125 and 150. A summary of the monitored occurrences for every initial setting of *I* is given in Table 5.3 (WLAN hotspot) and Table 5.4 (GSM).

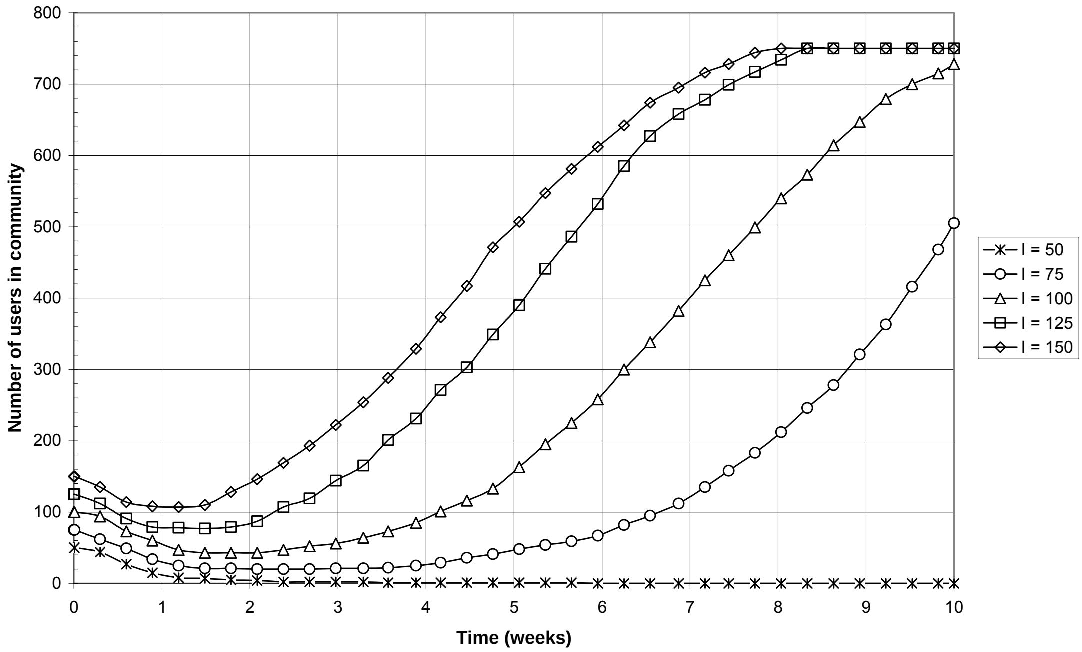


Figure 5.4 Variation of initial number of users in urban WLAN scenario

Initial user count (I)	% of users lost until $\bar{s} \geq \bar{f}$	Time when $\bar{s} \geq \bar{f}$ (weeks)	% of mapped network cells when $\bar{s} \geq \bar{f}$	Time until tipping-point (weeks)	% of mapped network cells at tipping point	Time until 95% of cells mapped (weeks)
50	100	N/A	N/A	N/A	N/A	N/A
60[9]	88,5	1,95	47,7	6,95	60,5	10,21
75	73,7	1,58	56,2	2,98	71,3	7,26
100	48,9	1,25	68,4	2,19	81,2	3,46
125	37,2	1,03	71,5	1,92	85,6	2,25
150	28,3	0,84	72,9	1,63	91,5	1,81

Table 5.3 Variations on initial user count for WLAN simulation scenario

In both scenarios, an almost constant initial loss of users in the community occurs until sufficient network cells have been mapped in the community database. This point is best described where the average satisfaction becomes equal and/or starts to overtake the average frustrations, i.e. the positive network effects start to occur. The tipping point where exponential growth of user numbers occurs follows a particular time later. In both scenarios this time interval grows linearly and is inversely proportional to the initial user base in the community, i.e. the time interval incurred with I = 150 is almost half than the interval incurred when I = 75. This is due to the fact that the lower the value of I is close to the identified initial user boundary condition the more balanced the number of frustrated and satisfied users close to the respective discouragement and recruitment thresholds are within the community (i.e. $\bar{s} \cong \bar{f}$).

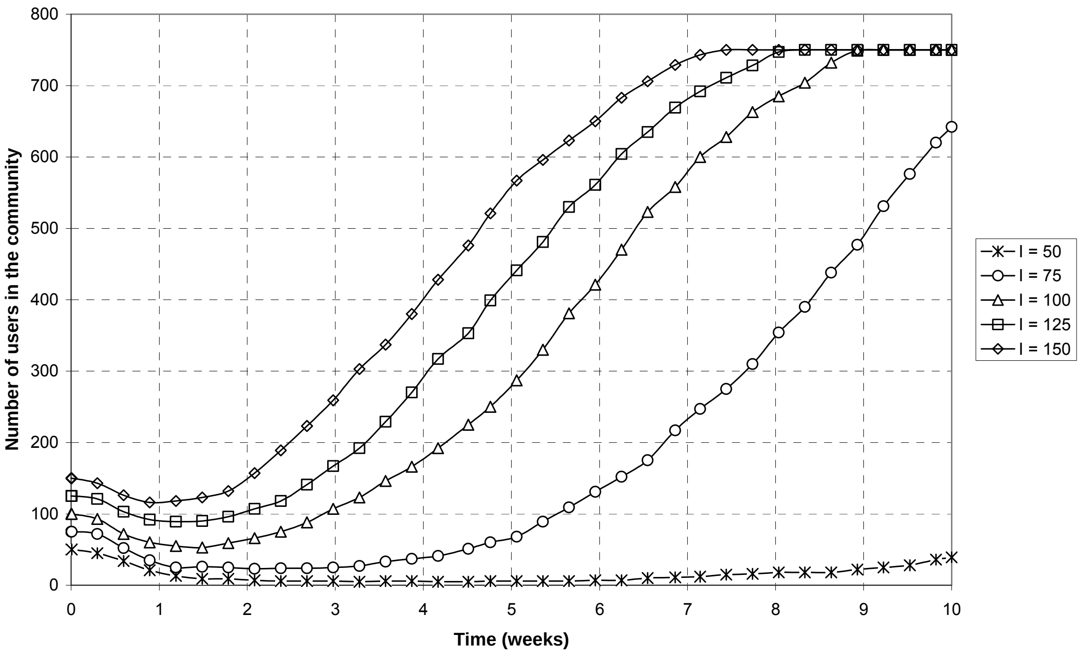


Figure 5.5 Variation of initial number of users in urban GSM scenario

In every case user satisfaction and positive feedback occurs when roughly 2/3 of the network cells have been mapped, indicating that overall location service acceptance is reached although the community data pool may be still incomplete or contain faulty data. The time taken between the tipping-point and until roughly 95% of the network

[9] This graph has been omitted in Figure 5.4 for presentation reasons

cells are mapped increases inversely proportional to the initial user count as well. The lower the initial user count, the longer it will take for the tipping point to occur within the community. After this point, the rate of growth exhibited is similar in all cases.

Initial user count (I)	**% of users lost until** $\bar{s} \geq \bar{f}$	**Time when** $\bar{s} \geq \bar{f}$ **(weeks)**	**% of mapped network cells when** $\bar{s} \geq \bar{f}$	**Time until tipping-point (weeks)**	**% of mapped network cells at tipping point**	**Time until 95% of cells mapped (weeks)**
40[10]	100	N/A	N/A	N/A	N/A	N/A
50	79,4	1,62	30,3	6,64	57,3	16,32
75	60,4	1,49	56,7	2,66	71,7	4,97
100	42,1	1,25	65,7	2,13	81,4	3,66
125	29	0,94	65,8	1,74	85,2	2,55
150	22,5	0,85	72,2	1,58	89,9	1,98

Table 5.4 Variations on initial user count for GSM simulation scenario

As the boundary condition for I is approached in each, the initial drop of users in the community, and the time until the tipping point occurs for a particular I can vary considerably. For example, during repeated simulation runs for I = 50 in the WLAN hotspot case the initial user drop until $\bar{s} \geq \bar{f}$ varied between 5-10 users, and the time of the tipping-point to occur roughly 1-3 weeks worst case. In the GSM case this effect was observed to be much less drastic. The increased standard deviation along the boundary conditions can again be explained due to the balance between borderline frustrated and satisfied users within the community. Furthermore, the different probabilities on location service usage, recruiting and user mobility incurred by each user contribute to these variations as well.

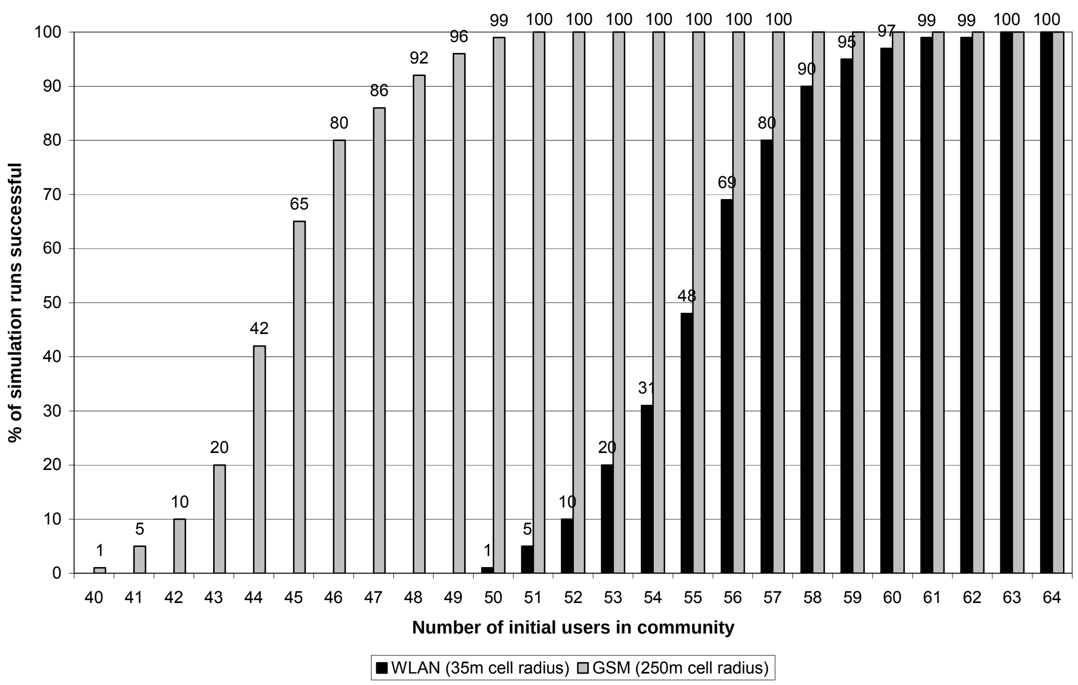


Figure 5.6 Probability of community survival given initial user count (GSM and WLAN)

[10] This graph has been omitted in Figure 5.5 for presentation reasons

The boundary condition concerning initial user count for the WLAN hotspot and GSM scenario was found out to at I = 59 and I = 49 users, respectively (both 95 percentile region). Figure 5.6 shows the probability of community survival tested for both usage scenarios for various simulation runs with varying I. Despite the different network cell radii used the incurred initialization times and community growth behavior become very similar for common I values being greater than 100 users. At lower initial user base values, however, the respective community growth behavior and probability of survival drastically changes. In the GSM case an initial user base of 50 users ensured an almost 100% survival probability of the community, whereas on the same I for the WLAN hotspot scenario the community was almost certain to die. This can be best explained by the fact that due to the larger cell radius in the GSM case the probability of one or more users using a location service more than once, and having to contribute manually only once whilst remaining in the same cell is much higher than in the WLAN hotspot scenario. Thus due to the smaller cell radius in the WLAN hotspot case, the probability of a user cell change is much higher, hence also the initial probability of having manually to contribute.

5.3.3 Effect of Average Usage per Week on Initialization

Having chosen an initialization strategy where users only contribute to the community-based location service on actually using the service itself, we need to look more closely at the effect the average usage per week has on the location service development. In the previous analysis on determining the minimum required initial user base on ensuring a successful launch of the community the average usage per week (U) was set to 7, yielding an average usage of once a day and a usage probability of roughly 4.2% on each simulated hour. This value is feasible in that it represents an eager usage of the community members due to location service's novelty and the curiosity of each user. By choosing the boundary condition value of I for a certain usage scenario determined in the previous section we will vary the average usage per week and analyze the respective outcome.

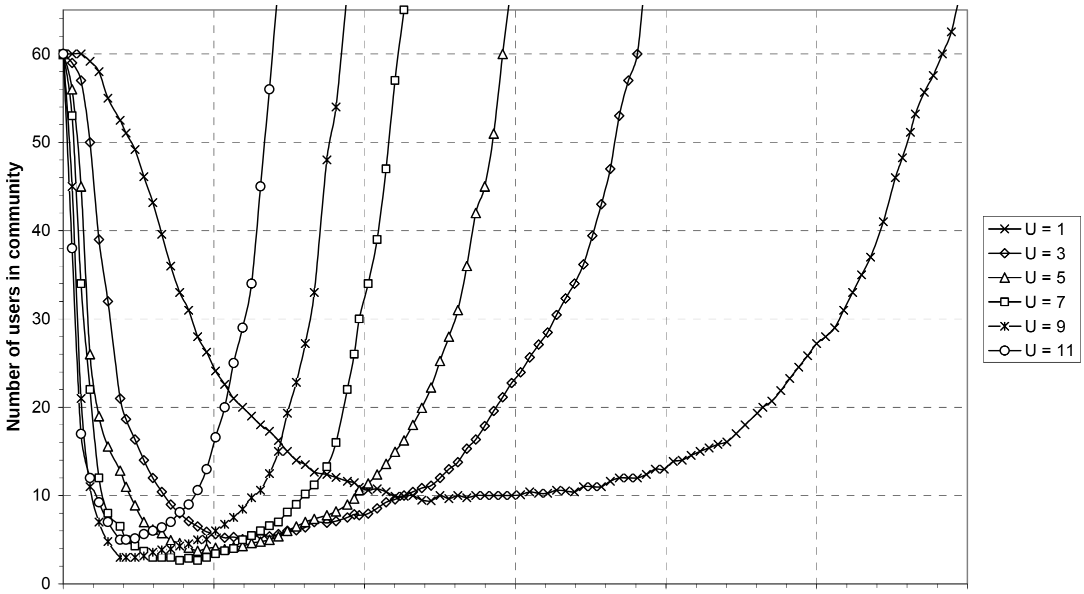


Figure 5.7 Varying usage per week for *I* = 60 WLAN hotspot scenario

The analysis was restricted to the WLAN hotspot scenario taking *I* as 60, indicating the boundary condition where the probability of survival is more than 95% (see Figure 5.6). The value of *U* was originally set to 7, and will now be evaluated for cases of *U* = [1, 3, 5, 9, 11, 14], simulating a usage of only once a week (*U* = 1) and up to twice a day (*U* = 14). All other settings remain the same as set out in Table 5.2. The simulation outcome is shown in Figure 5.7 and a summary of the key simulation results for each *I* variable setting is given in Table 5.5.

Decreasing the average usage per week has clearly the effect of prolonging the time where the average satisfaction starts to overtake the average frustration, the time when the tipping point occurs, and the time where finally 95% of the cells are mapped. Hereby the times almost increase exponentially, whereby the latter reference point extends to almost 40 weeks when the average usage is only once a week (i.e. *U* = 1). An explanation for this can be that the evolution rate starts to have an impact when the average usage per week drops to a similar value i.e. in this case *E* = 1 indicating an average cell failure of once a week. Hence the capturing of new cells is counteracted by the rate at which cells fail.

On the other hand, increasing the average usage per week has the effect of decreasing the observed time intervals proportionally, i.e. doubling *U* causes the respective time intervals to virtually half. Other simulation tests of increasing the average usage per week up to 28, 56 and 112 support this trend (i.e. an average usage of 4, 8, and 16 times a day), yet pose a not very realistic usage scenario.

Usage per week (U)	**Usage probability per hour (%)**	**% of users lost until $\bar{s} \geq \bar{f}$**	**Time when $\bar{s} \geq \bar{f}$ (weeks)**	**Time until tipping-point (weeks)**	**Time until 95% of cells mapped (weeks)**
1	0,6	83,5	12,21	15,47	39,47
3	1,8	91,6	5,95	6,85	22,03
5	3,0	93,7	4,46	5,71	16,94
7	4,2	95,6	4,44	5,06	12,52
9	5,4	95	1,89	2,67	11,58
11	6,6	93,3	1,88	2,38	8,83
14[11]	8,3	91,7	1,88	2,08	7,85

Table 5.5 Summary on usage per week variation using *I* = 60

Changing the average usage per week does also have an interesting effect on the number of initial users lost until $\bar{s} \geq \bar{f}$. A decrease in usage tends to results in a more gradual loss of user in the initial launch period, whereas a steady increase results in a stronger drop. Increasing the frequency of usage increases the probability of contribution, hence equally the probability of a user encountering a previously mapped or uncharted network cell. Since the latter case is true especially during the launch phase the user drop is more severe. Increasing the usage per week even further tends to counteract this initial heavy user drop again in that $\bar{s} \geq \bar{f}$ is reached even earlier.

In order to see whether the average usage per week can influence the probability of survival the tests have been repeated for *I* = 55, indicating a mere 50% chance of survival (see Figure 5.6). The obtained results where similar but the probability of success remained virtually the same. Therefore, in the development of the community-

[11] This graph has been omitted in Figure 5.7 for presentation reasons

based location service changing the average usage per week has the only effect of increasing or decreasing the rate of growth of the community but does not influence the probability of success as the initial user count does.

5.3.4 Evolution Rate of Access Networks and System Sustainability

Having evaluated the user-based factors the most critical environment related factor determining a successful launch of the community-based location service is the network evolution rate. This is factor relates especially to scenarios where the environment is highly dynamic, that is, an almost 100% coverage and/or network reliability cannot be ensured. Since the WLAN hotspot situation within inner-cities has the characteristic of multiple privately or publicly operated hotspot locations, it has been chosen for investigation on this manner.

Using the same simulation setup as used in section 5.3.2 the simulation for each value of I = [60, 75, 100, 125, 150, 200] has been carried out multiple times and repeated for evolution rate factors of E = [1680, 168, 48, 24, 12, 1] in each case, reflecting no cell failures up to one cell failure per hour. The average user satisfaction, frustration and percentage of cell covered outcome have been recorded after 10 weeks of simulation. The results are shown in Figure 5.8 and a detailed results listing given in Table C.3 of appendix C.

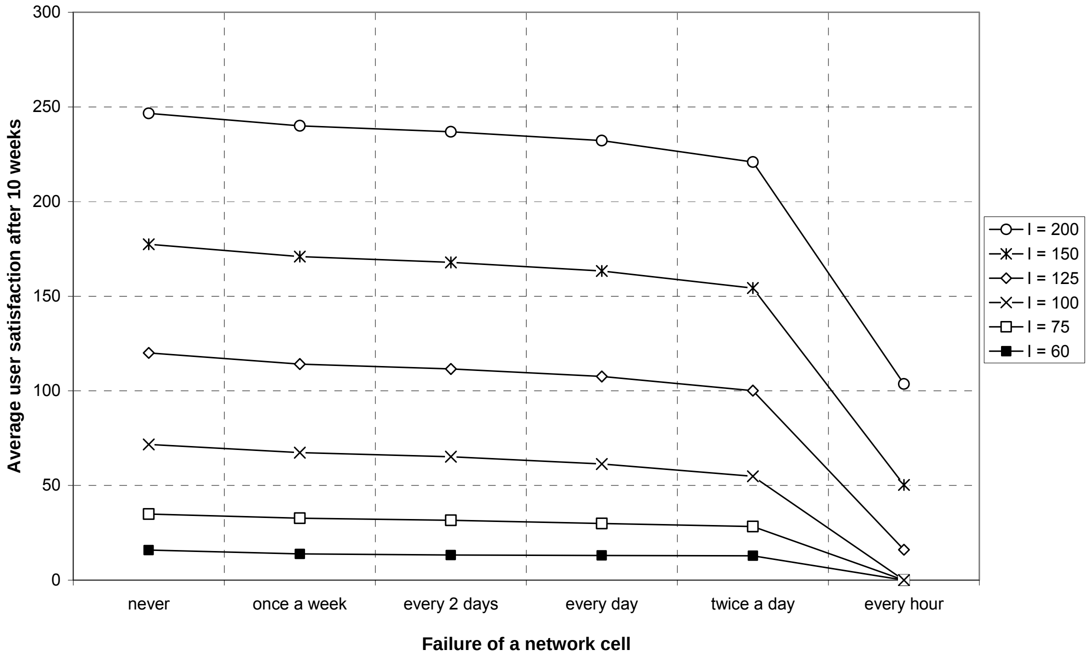


Figure 5.8 Evolution Rate behavior in relation to initial user count

Increasing the probability of a network failure per hour steadily over various initial user base numbers has the effect of reducing the average incurred user satisfaction, and increase the user frustration causing it to fail during the launch period of the community at a certain evolution/failure rate limit. Hereby the evolution rate effects have greater impact on the community development at lower initial user base cases, but have less impact the higher the number of available users in the community. As the initial user base is increased a higher number of users virtually compensate the evolution rate effects keeping a balance between cell failure events and participating members discovering cell changes and contributing them to the community. Furthermore the community is successfully launched and survives in most evolution rate cases, delaying the time of reaching $\bar{s} \geq \bar{f}$ and the community tipping point, and reducing the percentage of covered cells by the community only slightly, if not at all. Having the evolution rate of more than a day and up to a cell failure every hour causes the community to fail during launch period in cases where $I \leq 100$. In all other cases a constant drop of the number of mapped cells during the lifetime of the community is incurred up to the point where the maximum theoretical addressable target group size (T) has been long reached and beyond the observed simulation time of 10 weeks. The user community base equivalent to T was reached in cases for $I > 100$ before the observed simulation time of 10 weeks thus indicating the constant drop of covered cells as shown in Table C.3 of the appendix section.

Theoretically a large enough user base in the community can virtually compensate network cell evolution even beyond $E = 1$ to a sufficiently large enough average user satisfaction in order for the community location service to survive. Practically, however, being able to address a large enough initial user base for the community location service launch, and having more users at ones disposal than on average actually living in the designated usage environment (i.e. user density) is highly unlikely.

The average usage per week does have an impact on the average number of cells mapped in the community data pool respective to the incurred evolution rate. For example, in the case of $I = 100$ and E = 1 (i.e. one cell failure every hour) the community was successfully launched since the average usage per week was doubled from 7 to 14 (i.e. from once to twice a day). In every case for $I \geq 100$ the % of cells mapped after 10 weeks was higher as expected, but also a higher percentage level sustained during the lifetime of the location service. These effects, however, where not discovered for cases of $I < 100$, indicating that the initial user base is still the predominant factor for the community survival during the location service launch phase.

5.4. Summary and Conclusion

Having analyzed the boundary conditions for a community-based location service to be launched and sustained successfully for the proposed usage scenarios the following main observations can be summarized:

- The required initial user base is proportional to the usage environment area. It is the most dominant factor in determining whether the community-based location service will be successfully launched.
- This user base is essentially required to compensate the large initial user drop in the community due to unsuccessful location service executions and hence contribution effort required by each user during the launch phase.
- The more users participate at the beginning the shorter the time will take for all the network cells covering the environment to be captured.
- Assuming virtually 100% coverage of network cells a higher cell radius enables an on average higher user density per cell thus reducing the overall likelihood of users encountering an unmapped network cell. Therefore, the average frustration is lower enabling smaller initial user bases.

- The other dominant factor is the average usage per week which determines the rate of growth of the community. A higher usage rate results in a higher contribution frequency and faster community growth rate but increases the average incurred frustration and satisfaction equally.
- The evolution rate of network cells has a significant effect on whether the community is successfully launched. A high enough failure rate causes communities with certain initial user base to fail, or exhibit a reduced percentage of average network cells covered during the lifetime of the community. However, the effects of the network cell evolution rate on the community development can be counteracted by a high enough usage per week.

Although we have focused our analysis on these factors within the scope of this thesis, further evaluations on the remaining factors were carried out [ZD06], the results of which are briefly summarized within this section as well. Table 5.6 gives a summary of all relevant community factors and their impact on the community-based location service development.

The effect of user mobility as an indicator for spatial distribution of the location service usage within the community has little effect relative to the other factors. A high number of (initial) users within the community seems to provide a high enough spatial distribution already.

The recruitment and discouragement threshold are also important factors in the community development whereby the latter is much more critical. A higher discouragement threshold has only a moderate positive effect, whereas a lower threshold than what was used in the simulations (i.e. having $d_t < 3$) causes an overall 40% higher chance of community failure [ZD06]. Therefore, the users within the community must accept on average two more frustrations than satisfactions before abandoning the system in order to keep the community and its location service running for the proposed scenarios. A higher recruitment threshold just slows down the community growth but does not have a serious impact on the community survival.

Last but not least, the redundant location service request probability has only a negligible negative effect on the community development, tolerating a request redundancy of up to 20% [ZD06].

In our scenarios we have chosen a very neutral, worst case community model scenario disregarding the different motives and community user type natures. In reality the initial user base could be drastically reduced by offering incentives as previously discussed in this chapter. Using a small group of highly motivated lead users or dedicated service personnel during the community launch phase may clearly allow a similar and equally fast initialization and maintenance of the community-based location service. Such a scenario could have been simulated as well but would not be considered as an interesting research objective.

Mapping a network Cell ID to the community data pool provides the simplest proximity-based localization method as discussed in section 2.2.2, but is only capable of providing very poor localization accuracy (i.e. location quality) especially when bigger network cells seem to require less initial user bases. Concepts on how to provide much more accurate localization from multiple heterogeneous community user sensor streams will be covered in the next chapter.

Effect on *increasing* … factor on →	**Successful community launch and sustainability**	**Effect on rate of growth on community**	**Time of reaching $\bar{s} \geq \bar{f}$**	**Time of reaching tipping point**	**Important observations**
Initial user base (I)	+ + +	+ +	+ +	+ +	Relative to environment area and cell coverage
Average usage per week (U)	++	+ + +	++	++	A higher usage can compensate evolution rate
Probability of recruitment (p_r)	+	+ +	+ +	++	A higher p_r ensures faster community growth
Average cell radius (R)	++	+	0	0	Density of users /cell higher, probability that user encounters un-mapped cell reduced
Discouragement threshold (d_t)	+	+	0	0	Having $d_t < 3$ causes 40% higher chance of community failure
Average mobility (m)	0	+	0	0	Opposite effect to increasing cell radius, but users cover area quicker
Max. target group size (T)	0	0	0	0	Enables a higher user density in environment area, hence potential larger community
Environment area (A)	0	–	–	–	Provided network coverage remains the same
Redundant location service request probability (p_{rls})	–	–	–	–	A higher p_{rls} increases average user frustration correspondingly
Recruitment threshold (r_t)	–	– –	–	–	Affects community growth rate but not average user satisfaction
Network evolution factor (E)	– –	–	–	–	A high enough E causes higher average frustration
Key: + + + very strong positive effect, + + strong positive effect, + positive effect, 0 neutral, – negative effect, – – strong negative effect, – – – very strong negative effect					

Table 5.6 Summary of community factors and their relative effect on community development

Chapter 6
A Flexible Framework for Initialization and Positioning in LACBA

Following the analysis of the location service building and usage processes using the set out community principles, it is now crucial to define a suitable framework for a community-based LP, realizing the collaborative initialization and location information generation from multiple heterogeneous infrastructure sensor data captured by users from the designated usage environment. Hereby the selection of appropriate initialization and positioning models suitable for the collaborative location information generation and self-optimization is a crucial issue.

At first, the LP model concept will be further developed considering the requirements which were set out in chapter 4. Further discussing the requirements on processing multiple sensor data streams and potential error factors, a component model view of the respective LP LES component will be given. Once the appropriate LES framework has been established, possible operational modes, corresponding initialization and positioning models on processing sensor data as well as location models on representing the environment information will be discussed. Suitable initialization and positioning models will be chosen for evaluating the collaborative, community-based localization approach and possible positioning performance.

6.1. Developing a Flexible LES Framework

Due to the common functional components and nature of positioning systems and their corresponding LP solutions as described in sections 2.2.3 and 2.3.3 respectively, the goal is to develop a LP which can be deployed and adapted to various usage environments. Hereby the LES part is clearly the most important component providing the best possible location information having used best possible initialization methods providing very accurate reference locations, and using suitable localization methods for positioning in the designated environment – be it indoors or outdoors. Clearly, a community-based solution as developed by this work provides another level of complexity and challenge.

A generic all-in-one approach has the following features extending on the requirements originally specified in section 4.1.3 and as exemplified in Figure 6.1:

- In order to support localization in various usage environments, i.e. indoor or outdoor, a repository of appropriate initialization and positioning models need to be provided. Selection of respective suitable module pairs usually happens manually and is carried out by power users but the possibility of automated approaches should be investigated further.
- The initialization and positioning processes are independent of each other, each of which can be carried out automatically or manually depending on the mode of operation as discussed later on (e.g. distinct processes or "continuous initialization", i.e. dynamic adaptation of location reference data during the lifetime of a LP)
- Contributions as depicted by (1), (2) and (3) and previously in section 5.1.1 should be supported. For each positioning technology single and multiple sensor data streams (independent of volume) should be able to be processed accordingly for generation of suitable location reference data.
- Every sensor data contribution needs to be treated anonymously, verified and integrity checked before initialization occurs via suitable filter functions. Furthermore, although considering each sensor technology individually, sensor data errors and differences caused by hardware differences and configurations, as well as environment fluctuations need to be accounted for as well.
- Fusion and smoothing (i.e. by means of interpolation) of multiple and/or heterogeneous sensor technology positioning outcomes should be supported but is seen as optional.
- Manual user and automated feedback on positioning for location reference data improvement should be supported but seen as optional.
- The LES location service should at least provide an accuracy estimate along with the location information generated within its location model (i.e. LQoS). The area location model as discussed in section 2.2.1 and 4.5.1 was chosen as the most useful model and will hence be used in the evaluation of suitable initialization and positioning models for the LACBA LES framework later on.

Apart from these functional requirements the generic LP should exhibit the following (non-)functional requirements as set out in section 2.3.2 and 2.5:

- High location service availability and reliability
- Platform compatibility and openness to existing location service provisioning standards [OGISa][OGISb]
- Further privacy and security measures to protect and restrict access on tracked users, e.g. implementing proposed solutions as in [Leo03] or readily available AAA mechanisms [Rot05].
- In the case of proprietary coordinate formats typically used in indoor environments localization is usually three-dimensional whereas outdoor localization is two-dimensional (i.e. floor levels in buildings). Thus concerning initialization and later positioning a respective indoor LP should provide additional information and/or mapping between the local proprietary and a globally used outdoor format to support interoperability, i.e. by use of a suitable coordinate format such as WGS84 [ZDS04]

The remainder of this chapter will focus on the functional requirements on developing a suitable LES framework. Some of the non-functional requirements will be covered later in chapter 8 dealing with the implementation aspects of this work.

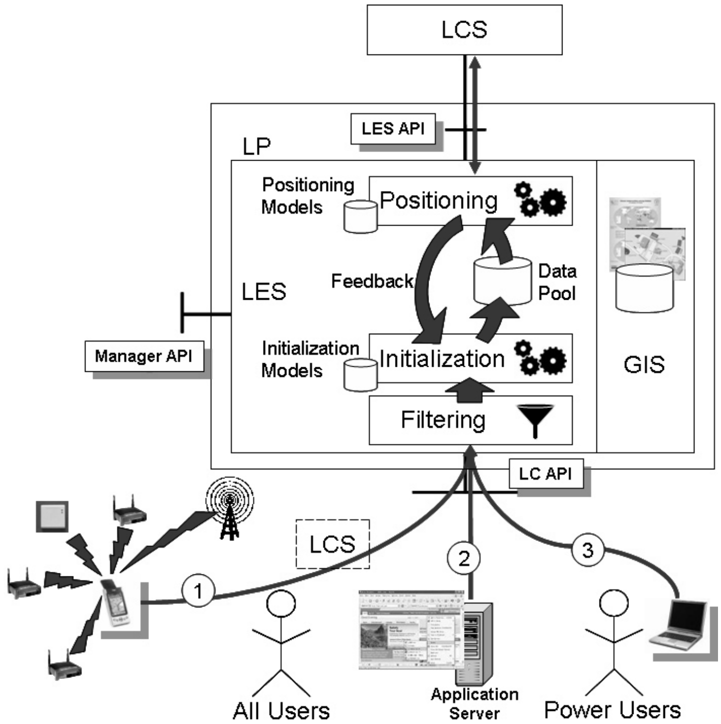


Figure 6.1 Overview of the LACBA location platform principle

6.1.1 Precautions on Processing Collaboratively Captured Sensor Data

The reference location data generation process referred to as initialization in positioning systems introduces some error factors as explained in section 2.2.3. Furthermore, collaboratively capturing wireless network information and location reference data generation by multiple mobile clients as depicted by this work introduces additional potential error factors, hence requires some additional practical precautions when deploying a system providing a common location service based on such a principle. All in all, the relevant factors influencing and providing potential sources of error can be summarized as follows: hardware, environmental, measurement, and heuristics-based influencing factors.

The ***hardware-based*** factors are a result of the heterogeneity in hardware, i.e. varying configurations/implementations of certain common technologies used in infrastructure as well as client based components. On the infrastructure side, for example, WLAN access points implementing a common standard can be configured differently concerning their transmission power used, whether to transmit broadcast beacon packets announcing their presence, or whether to respond to broadcast probe packets sent out by mobile clients in *active* scanning mode. In the latter case, the access points seem to be invisible to mobile clients. Using clients in passive scanning mode (i.e. sniffing traffic on all available Wi-Fi channels) allows such "cloaked" access points not transmitting their network ID to be detected.

On the client side the heterogeneity of different hardware chipsets used and use of internal or external antennas can considerably affect the observed signal-strength values and number of received traffic packets by the client. Larger, external, and special types of antenna can potentially detect much weaker and overall stronger access point signals over larger distances compared to smaller, often device-internal, standard configurations. The differences in raw signal-strength variations across chipsets can be compensated by simple linear cross-correlation as shown by [HFL04] and verified by [CCK05]. The same is true for the antenna variations if the respective antenna gains are known. Knowing the unique MAC address, chipset manufacturer and antenna documentation can be obtained freely from certain online databases [Wig06]. Apart from varying raw signal-strength interpretations across different chipsets, the rate at which Wi-Fi channel hopping can be carried out varies as well (e.g. from a few milliseconds up to 1-2 seconds [Ipp06]). Thus some chipsets may miss important traffic beacons on a drive-by scanning session compared to others.

Apart from the hardware variations to be considered wirelessly transmitted signals suffer ***environmental*** factors such as multipath signal fading and reflection effects. Due to the dynamic nature of moving objects and people in both indoor and outdoor environments signal-strength and timing-based measurements will always fluctuate to some degree. Some localization approaches as described in section 2.2.2 are more suited than others in compensating for these effects to a reasonable extend.

Measurement aspects play another important influencing role in both initialization and positioning processes. In the reference location data generation during initialization (see section 2.2.3) the reference location source used as ground truth for geo-coding respective signal-strength profiles or infrastructure components influences and determines the maximum achievable localization accuracy the most. For example, using GPS as ground truth for initialization of a WLAN hotspot-based outdoor LP in an inner-city scenario potentially already imposes several meters of error. As some experiments have shown, this still results in sufficiently enough localization accuracy for WLAN hotspot-based positioning in such scenarios, on average between 30 to 150m, which is the equivalent of usually less than a block of houses [CCK05]. On the other hand, indoor LP initialization usually requires “room accuracy” or better in most cases, thus demanding more sophisticated manual user, laser- or radar-based referencing methods [ZDS04]. Apart from the ground-truth source used the physical distribution of measured sensor data points used for geo-coding a target reference is critical, too. For example, an even distribution of sensor data measured close to and around the true location of a WLAN access point will result in a more accurate geo-coded position than the distribution being more spread out and/or biased to certain locations nearby (Figure 6.2).

Concerning positioning processes the window of observed sensor data readings used for localization in relation to the tracked target traveling speed is important. Using more observed sensor data for localization may result in a more accurate and stable position estimate provided the targets true position does not change much.

Appropriate initialization and positioning models need to be chosen for particular environment situations in order to minimize ***system heuristics-based*** influences as much as possible. Some initialization methods used are more suitable regarding sensor data volume, spread in measured sensor readings and physical sensor data point distribution handling than others when it comes to creating location reference data as accurately as possible. This is especially the case when the LP requires “adaptive” and “continuous initialization” such as in the community case. The same is true for positioning methods. Different localization approaches work best with different base station densities, respective radio coverage and sensor data fluctuations caused by dynamics in the environment. We will look and evaluate initialization and positioning models suitable

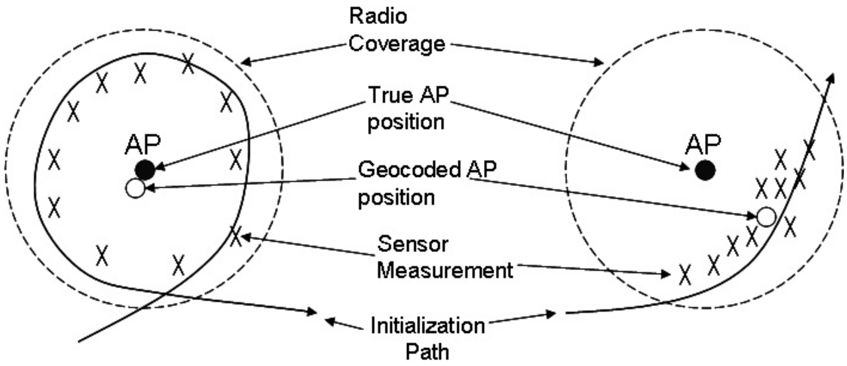


Figure 6.2 Distribution of sensor measurements for geo-coding during initialization

for our community scenario later on in this chapter. All of the introduced factors influence the sensor data for both initialization and positioning procedures. Pre-processing and validation of the data by means of suitable filter-sets as well as choice of respective models for each particular usage scenario can account for more or less all of these factors.

6.1.2 LES Component Model

Coming from the LACBA LES principle which has been illustrated in Figure 4.4 of section 4.1.3 we can also define a component model view as we have done previously concerning the LACBA location service. Figure 6.3 shows the defined LES control areas and respective components fulfilling the LES requirements as set out by this work.

The LACBA Client Control

The handling of multiple sensor data streams as well as data preparation for initialization and positioning is carried out by the *LACBA Client Control*. The tasks can be subdivided into the LACBA client session management, validation, and filtering as well as sensor data adaptation to the respective initialization and positioning module requirements used by the LES. These functions are carried out independently of the actual initialization and positioning processes. The *Sensor Session* established as described in section 4.7.2 delivers the appropriate authentication and LES positioning technology specific sensor parameters from the corresponding LACBA Client via the *LACBA Client API*.

Components of the *LACBA Client Control* include the LACBA Client Manager (LCM) controlling the LACBA Client access to the LES, a LACBA Client Proxy (LCP) which is initiated for every active LACBA Client managing the sensor data extraction, adaptation and necessary filtering, as well as the LACBA Client Database (LCD) holding LACBA Client hardware configuration and handling information.

The *LACBA Client Manager* handles all client session requests with the LES on the Sensor Session initialization with the UP of the LCS. Upon successful access rights and LACBA Client validation the LCM generates a LCP instance and directs all sensor signaling issues to this entity. The LCM also signals state information of active clients currently connected to the MM necessary for handling and control of the initialization and positioning processes.

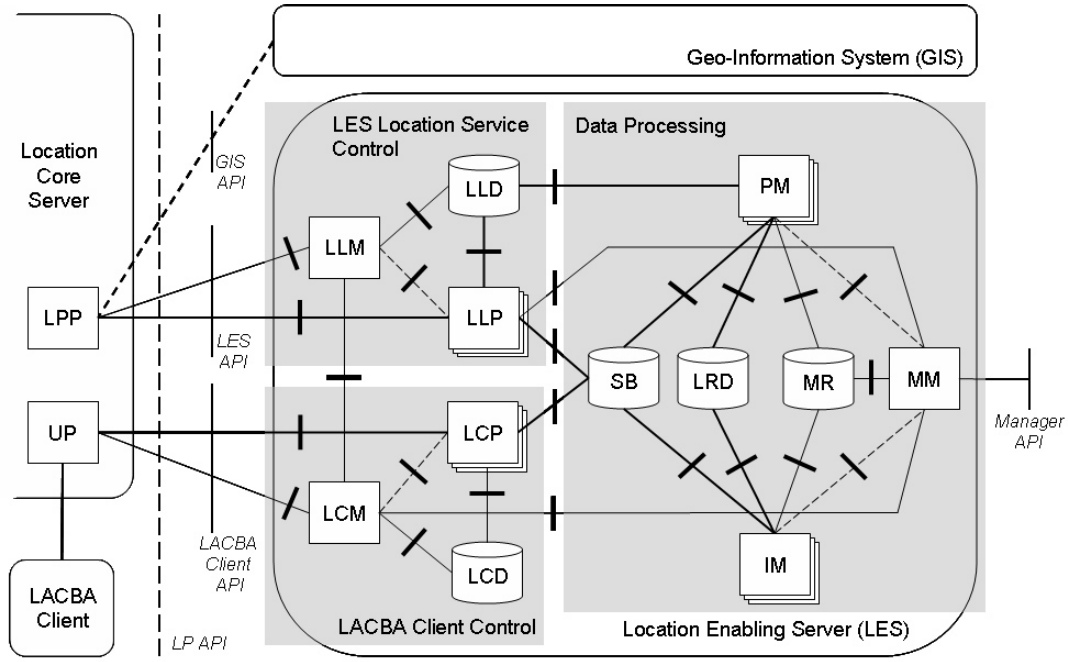


Figure 6.3 The LACBA flexible LES Framework component view

Key:

UP User Proxy	LLD LES Location Service Database
LPP Location Platform Session Proxy	SB Shared Buffer
LCM LACBA Client Manager	MM Module Manager
LCP LACBA Client Proxy	MR Module Repository
LCD LACBA Client Database	LRD Location Reference Database
LLM LES Location Service Manager	IM Initialization Module
LLP LES Location Service Proxy	PM Positioning Module

A *Location Client Proxy* handles the LACBA clients active sensor session performing the following functions:

- **Data extraction,** e.g. decompression and decryption as well as anonymization.
- **Sanitizing** and **integrity** checking, e.g. is the data complete and/or in the right format/sensor data range? This helps to prevent malicious attacks or system initialization/positioning errors.
- **Transformation** of address and physical coordinate input formats
- Additional optional **filtering** and **normalization** functions, e.g. to account for different hardware chipset and/or antenna configurations [HFL04], but also black/white-list enforcement determining the degrees of which users are allowed to initialize and/or perform positioning.

The pre-processed sensor data is then stored into the SB and referenced using the common UID provided via the *Sensor Session*.

The *LACBA Client Database* holds LACBA client hardware profiles and active client information being accessed by both LCP and LCM entities.

Data Processing

The heart of the LES is the *data processing* plane performing the actual initialization and positioning tasks. The following processes can be mapped and carried out by the respective components:

- Location reference data generation from the provided LACBA client sensor data.
- Positioning from sensor and location reference data.
- Fusion of multiple localization results.
- Feedback on provided localization results and location reference data improvement.

In the *data processing plane* the Module Manager (MM) coordinates the usage of both initialization (IM) and positioning models (PM) from the Module Repository (MR). The Shared Buffer (SB) holds all captured and pre-processed sensor data from all currently associated LACBA clients ready for initialization and positioning. Last but not least, the Location Reference Database (LRD) holds all generated location reference data from initialization for the respective environment and positioning modules to use for localization.

The *Module Manager* has the knowledge about all initialization and positioning models available and used for the current usage environment. The selection and use of respective models is either configured manually via the MM API and/or automatically using the sensor type, input sensor parameters and state of LACBA Clients delivered via the LCM. Similarly, the MM handles location requests via the LLP as explained further on. In both cases the input parameters are compared against the module specifications stored in the *Module Repository*. Even more importantly, the MM regulates the initialization and positioning module access to the LRD during localization and initialization respectively, i.e. ensuring that there is no location reference data modification during a localization requests.

The *initialization* and *positioning modules* themselves are kept generic as sensor data adaptation to ensure module compatibility which occurs at the LCP. An initialization module uses one or more LACBA Client sensor data instances from the SB and already generated location data from the LRD for reference location data building, storing the results again in the LRD. Furthermore, an initialization module can have an active initialization as well as passive role in performing monitoring, integrity checking, and other statistical analysis on the generated reference location data. This is crucial for implementing some of the feedback and self-learning features as explained in the next section. A positioning module uses the current sensor data of one client from the SB and validated reference location data from the LRD for localization. The localization result for this client (again using the UID) is then stored in the LLD. For fusion and estimation/interpolation purposes appropriate PM can use the LLD localization results and current LACBA client sensor data from the SB. In order to support adaptive initialization several initialization and positioning modules can be run in parallel supporting multiple reference location data buildup for a stepwise refinement of the data set used by the positioning modules for localization.

The *shared buffer* holds the prepared sensor data of all actively associated LACBA clients with the LES as well as feedback sensor data from the LES location service via the LLP. The *Location Reference Database* holds the generated location reference data and information on which data-set has been validated/released for localization purposes

The LES Location Service Control

The *LES Location Service Control* handles multiple location service sessions providing location information on currently associated LACBA clients with the LES, and provides a feedback channel on localization results. The tasks can be subdivided into location service management, location information provisioning and feedback mechanisms. These functions can only be provided if an active sensor session for a to-be localized client with the LES exists. As described in section 4.7.2, the *LP Session* provides the required location data delivery information as requested by a LBS via the *LES API* (e.g. automated location update intervals).

The *LES Location Service Control* components include the LES Location Service Manager (LLM) responsible for controlling the location service access to the LES, a LES Location Service Proxy (LLP) which is initiated for every LP location service request from the LCS providing location information access and feedback on localization as well as the LES Location service Database (LLD) which holds the localization results generated for the respective clients.

The *LES Location Service Manager* handles the LP session requests with the LES on the LP Session initiation with the LPP of the LCS. On verifying that the Sensor Session for the corresponding LACBA client exists, the LLM generates or terminates a LLP instance. On generation of a new LLP, all location service requests and handling issues are directed to that entity.

A *LES Location Service Proxy* handles the active LP Session with the LCS providing the requested location information and additional LQoS as specified by the location model using the localization results from the LLD (using the UID). The LLP triggers the localization requests on the currently available PMs via the MM. In doing so the LLP also forwards requested LQoS information to the MM for appropriate PM selection and handling. Similarly “re-initialization” can be triggered on appropriate IMs via the MM when feedback information obtained on LES location service usage is provided and stored in the SB by the LLP.

Finally, the *LES Location Service Database* holds the localization results on various actively associated clients provided and accessed by PMs (for localization, fusion, and estimation purposes respectively) as well as LLP agreed LQoS provisioning information.

6.1.3 LES Data Processing Principles

The functionality of different LES initialization and positioning processes supporting fusion, estimation, and feedback/learning possibilities can be resembled by various LES component configurations and inter-working between the various control parts. In order to illustrate each of these LES data processing concepts we will use the previously introduced component view showing corresponding signaling principles.

Single and Multiple User and Adaptive Initialization

The LACBA LES framework can support single-user administered- and multi-user un-administered initialization common to indoor WLAN RSS-based [Eka06] and community-based approaches respectively, as depicted in Figure 6.4. In both cases, independent initialization and positioning phases as illustrated in section 2.2.3 are possible to be implemented.

In order for initialization to occur appropriate sensor data needs to be available in the SB and/or at least one LACBA client needs to be actively connected via a Sensor Session to the LES. Single and multiple LACBA client sensor sessions can be supported, each having common or different contribution approaches as depicted in Figure 6.1. Considering, for example, the following collaborative inner-city WLAN hotspot geo-coding community scenario:

- LC1 and LC2 provide WGS84 coordinates from a GPS receiver, as well as WLAN MAC address and RSS traffic beacon information
- LC1 is a LACBA client performing a single manual bulk sensor log upload
- LC2 performs automated incremental sensor data uploads every 3 seconds
- LC3 is a single manual user contribution of a WLAN MAC address and a street address as location input via a local client application.

In each case, the respective LCP units filter and harmonize the time-stamped sensor data contributions. In the case of LCP3 a street address to WGS84 location format conversion is carried out. The corresponding sensor data and processing information is then stored in the SB.

The availability of new sensor data and sources is signalled to the MM by the LCM triggering an initialization process in this way. Alternatively the initialization process is triggered manually by an administrator via the Manager API from a corresponding initialization monitoring and management application. Hereby the MM provides IM module access and handling as well as MR, SB and LRD database access respectively. On triggering an initialization event the IMs carrying out the reference location data building process the available sensor data and store the results in the LRD to be used for positioning. The respective outcomes can be evaluated and selected by the external management application via the MM. For example, the IM1 entity can perform a *point-of-maximum RSS* calculation storing the WGS84 coordinate of the strongest RSS beacon discovered of a unique WLAN MAC address in the LRD. IM2 on the other hand could register *fingerprint profiles* of multiple WLAN MAC addresses seen at a particular WGS84 coordinate instead [DZ02]. Depending on the amount of sensor data entries captured and provided one or the other dataset produced could be more useful.

In the case of LCP3 the converted street address entry is taken instead. Weighting of available LC sources can also be provided via the LCPs and considered in the initialization process (e.g. trustworthiness of users etc.). Existing WLAN MAC address LRD entries are overwritten and replaced by the newly generated references. A more intelligent “merging” and updating of the existing data is shown next.

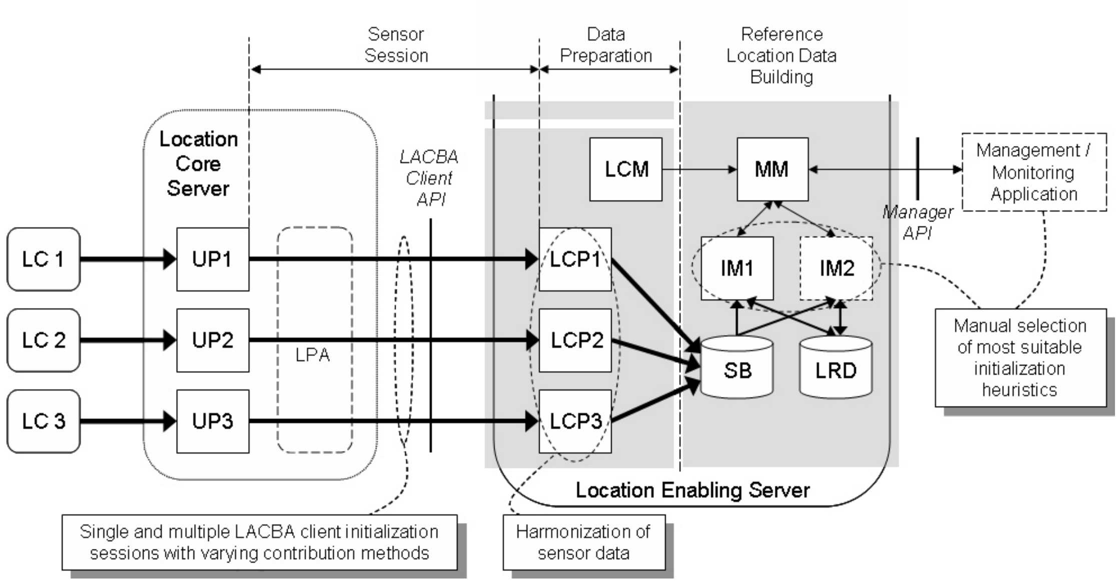


Figure 6.4 Single and multiple user initialization LES configuration

Adaptive and Continuous Learning

Having presented a fairly static approach for single and multiple user initialization in the previous example the LES framework can also support more dynamic and adaptive methods providing support for continuous and automated learning of infrastructure changes in the environment. The tangible tasks are as follows:

1) Automatic adaptation of the initialization process due to changed initialization conditions, e.g. types of sensor data and/or volume; number of LACBA clients
2) Improvement of reference location data quality in the absence of sufficient sensor data
3) Integrity checking and validation of generated reference location data triggering task 1) and/or a complete re-initialization of the LRD data pool upon discovering (sufficient) infrastructure changes in the environment.

Figure 6.5 shows a similar LES situation as before with multiple connected clients contributing sensor data as before as well as IM1 and IM2 resembling the *point-of-maximum RSS* and *fingerprint profile generation* initialization methods. This time, however, the selection of the appropriate IM generated LRD dataset is determined solely by the MM module using other indicators provided by additional passive IM modules (IM3 and IM4).

Integrity checking and validation could be carried out several ways. For example, module IM3 could represent a nearest-neighbour heuristic running independently in parallel to the other currently active IM modules. Available sensor data in the SB could be checked on whether neighbouring WLAN Access Points in the same radio coverage of identified WLAN MAC Addresses in the LRD are still physically present (i.e. network beacons are still being received by the LACBA clients). If they are missing, the appropriate LRD dataset references are flagged appropriately and potentially removed or marked as invalid/unreliable for localization estimates after repeatedly not being discovered. Furthermore, IM4 could represent a *data-aging* heuristic determining the

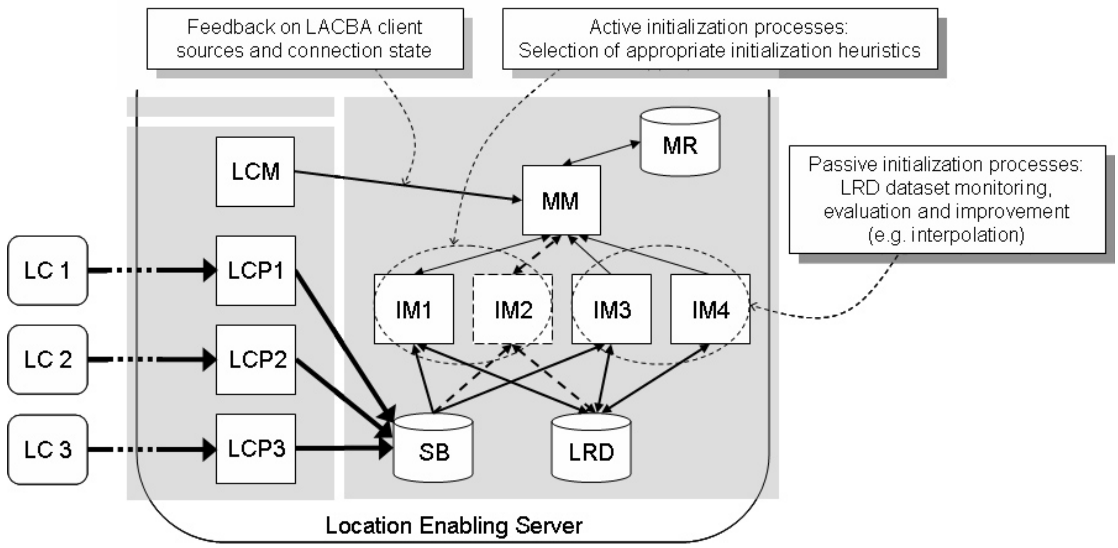


Figure 6.5 LES component configurations for automated adaptive and continuous optimization

reliability of LRD datasets on how long ago the physical presence of a particular WLAN Access Point entry was not verified by recently captured sensor data. The MM can thus use client connection state information and respective IM entities for dataset validation, i.e. to be used for automated, most suitable LRD dataset selection for localization, selection of appropriate IM entities for initialization, and triggering of respective (re-)initialization processes once certain boundary thresholds have been met.

Instead of representing a *data-aging* heuristic IM4 could also represent an *interpolation algorithm,* e.g. TPS [DZM04], interpolating between available datasets and generating additional ones if only sparse sensor information has been provided and thus only few reference location datasets are available in the LRD. Such interpolation approaches are especially useful as system heuristics during the initial launch phase of community-based LPs for improving initial localization performance.

Positioning, Fusion and Estimation

On the positioning process aspects the LES framework provides equal flexible and dynamic configurations as with the initialization part. Provided that the corresponding *Sensor Session* of a to-be positioned LACBA client is present and the *LP Session* was initiated the following localization functions can be provided:

- Dynamic selection of available localization heuristics depending on required LQoS.
- Smoothing of positioning results from a particular localization heuristic
- Fusion of multiple positioning results
- Estimation and prediction of positions using history location data

Figure 6.6 shows two active location service sessions (*LP Session*) with the LCS. Each session has an agreed LQoS which the LES can provide on the corresponding positioned LACBA client (i.e. LLP1 and LLP2). Localization requests are triggered via the MM resulting in the available PMs to calculate the current position of the LACBA client from current sensor data, and available LRD datasets to be calculated and stored in the LLD. The corresponding LLP can then select the appropriate position result from the LLD fulfilling the required LQoS.

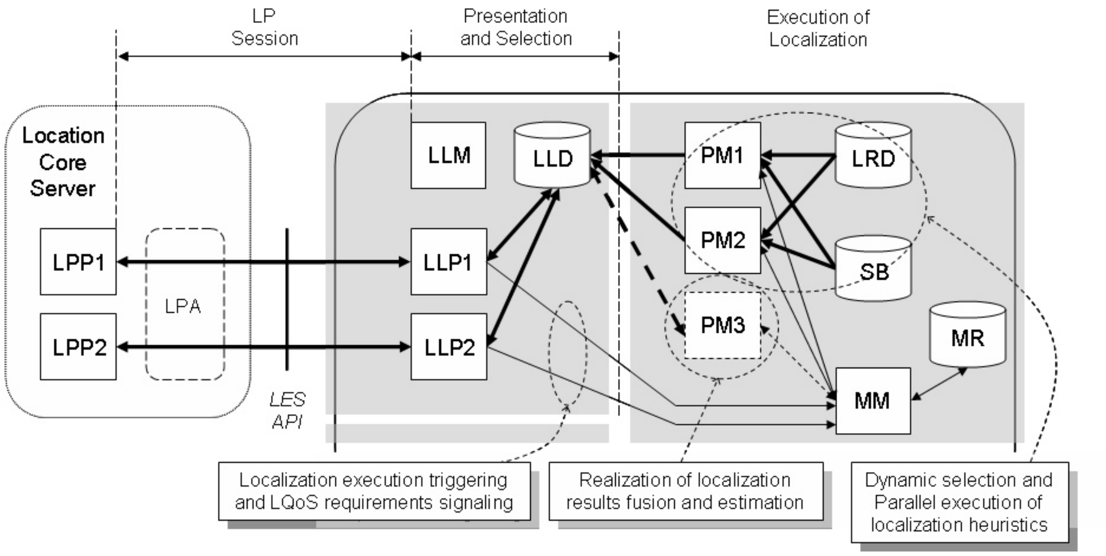


Figure 6.6 LES component configurations for positioning, additional fusion and estimation

Fusion can be implemented by having appropriate PM fusion heuristics access the computed position results of multiple PMs from the LLD (e.g. PM3 performing fusion on the position results of PM1 and PM2). Similarly, smoothing of positioning results of a particular PM can be carried out as well in case of strongly fluctuating sensor measurements due to dynamics in the environment [DZ02] (e.g. PM3 averaging several positions over a specified time window). Following the same example estimation functions represented by PM3 can predict potential future LACBA client positions calculating a direction vector from caches positions in the LLD.

6.2. Initialization and Positioning Models for a Community-Based LES

By having established the generic, flexible LES framework we now need to identify suitable initialization and positioning heuristics which can fulfil the originalities of the community-based location service concept in delivering a "competitive" localization performance. We will further use the WLAN hotspot-based localization in densely populated inner-city scenario as described in chapter 5. Although it clearly won't be possible to achieve close to or "better than" GPS performance the goal will be to reach on average "much better than" WLAN hotspot localization accuracy meeting current E112 and E911 regulatory standards [FCC06]. We will refer to this as sub-hotspot radio cell coverage or sub-cell accuracy for short from now on.

6.2.1 Determination of Suitable Initialization and Positioning Methods

The presented LES framework and respective pre-processing mechanisms account for much of the precautions set out in section 6.1.1 apart from the following issues which have to be handled by the initialization and positioning mechanisms regarding the community location service scenario:

1) A huge volume of sensor data needs to be processed. The contributions will be part WLAN RSS beacons with unique MAC-Addresses as labels with and w/o location reference information (e.g. GPS or street address). We will assume that all sensor data used for initialization included a location reference.
2) A very large usage environment hence large amount of reference LRD datasets. This has computational impacts in particular.
3) Only partial visibility and reachability of infrastructure components (hence biased distributions of measurement points, see Figure 6.2). LACBA clients will be mainly moving about freely accessible spaces in the inner-city picking up WLAN beacons of Access Points within buildings passing by outside on the street or at least going around a corner.
4) On average we can assume more than one visible access network at any point in time due to the nature of the environment and access point density.
5) Frequent changes in infrastructure need to be tolerated, i.e. (re-)moving and spawning of new access points. This calls for adaptive initialization methods.
6) Dynamic outdoor environment, i.e. signal fluctuations caused by people/objects moving about
7) Mobility of LACBA clients capturing data.

Having described the different principles of localization and initialization requirements in section 2.2.2 and 2.2.3 we know that initialization either determines the more or less exact position of the infrastructure access network base stations or generates sufficient reference location data using the measured raw sensor data. Knowing the advantages and disadvantages, and hence boundary conditions of possible localization methods from section 2.2.2, we can draw the following conclusions concerning the choice of appropriate initialization and positioning methods:

- Not knowing the locations of the access network base stations a priori and being provided with the type of sensor data as in 1), we can determine the base station locations accurately to a certain degree depending on the amount of measured sensor data provided (limited by the facts in 3)). Thus, we can possibly allow for geometric localization using RSS triangulations (due to 4)). But due to the weaknesses as pointed out in section 2.2.2 as well as points 5) and 6) this method can potentially be highly negatively affected.
- Localization by scene analysis e.g. fingerprinting, would be feasible due to 1) and 4) but vice versa not suitable due to the large usage area (point 2)) resulting in a huge storage space. Additional clustering of the datasets would need to occur in order to reduce computational overhead (see section 2.2.2). But the real “killer criteria” is the problem of self-learning without administration, updating, and filtering out redundant dataset entries (point 5))
- Similar to the scene analysis approach statistical approaches would also inhibit computational bottlenecks, but would be more robust/adaptable to changes in the environment provided they do not occur too frequently 5). As mentioned earlier, probabilistic methods either assume normal probability distributions (hence problem regarding 3) [SKS03][Kal60]) or need to convey the measurement point distribution respectively.

- Proximity-based methods matching client to the identified access network base station locations are always feasible but we are trying to aim at slightly better localization performance than such methods can offer.

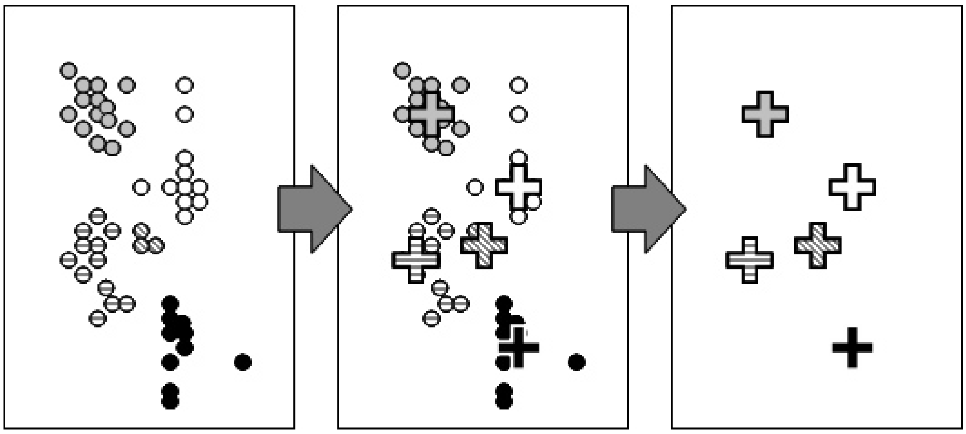
Concerning initialization purposes the reference location data generation from the huge sensor data amount in our scenario favours the geo-coding of access network base stations over the fingerprint RSS profiles required by scene analysis and/or statistical approaches. The information reduction in the former case is much greater resulting in lower storage and computational requirements. In order to account for the self-learning and adaptation requirements statistical methods relating to cluster analysis [NH94] can be best applied in both initialization and positioning processes. For the localization, simple heuristics being tolerant to 5) and 6) requiring low computational power are needed. Basic statistical localization methods such as the Centroid algorithm [Ng03] are suitable in this case. Probabilistic methods could potentially work better in both initialization and positioning processes but it is to be doubted that potential accuracy gains (if at all) justify additional computational requirements and complexity [Ipp06][LaM05]. Finally, the mobility aspect of LACBA clients relating to both initialization and positioning has to be accounted for e.g. by setting timeframe windows on observed sensor data for processing in each case.

6.2.2 The Sub-Cell Processing Model

The following sections will analyse whether the chosen approaches on cluster analysis, information reduction, and pattern recognition can cope with the huge sensor data volume generated by the collaborative capturing and contribution processes, and still provide the desired localization performance. The initialization and positioning process steps of the sub-cell processing model can thus be described as follows:

Initialization

The collaborative sensor data capturing and provisioning provides a large amount of measurement point locations. Each measurement point has a unique label, i.e. network identifier such as a MAC Address or Cell ID. Therefore we can generate clusters of measurement points as illustrated in Figure 6.7. Having a cluster of N measurement points the goal now is to extract certain amount of feature parameters X which are much smaller than the sum of the respective cluster measurement points, i.e. $X << N$. We call this process ***feature extraction,*** which is carried out after the sanitation and integrity checking and will be explained later on in more detail. Figure 6.7 shows the process of performing feature extraction on the raw sensor data producing respective ***cluster prototypes*** representing the original cluster of measurement points defined by the feature parameters used, e.g. point of max. RSS. Apart from the extracted cluster prototypes further feature information can be obtained on the points cluster such as the cluster size, its shape, its position in space, or even the cluster points density. All this information describes the point cluster cloud and could be used for reconstructing the original point cluster from the generated cluster prototype to a certain extend. The goal of the feature extraction, i.e. information reduction is to minimize the error between the original cluster point cloud and the “reconstructed” cluster cloud under the premise of using as little feature information as possible hence also minimizing required storage space. Apart from reducing storage space feature extraction and fewer parameters reduce the overall complexity and improving the speed of the later localization methods to be carried out.

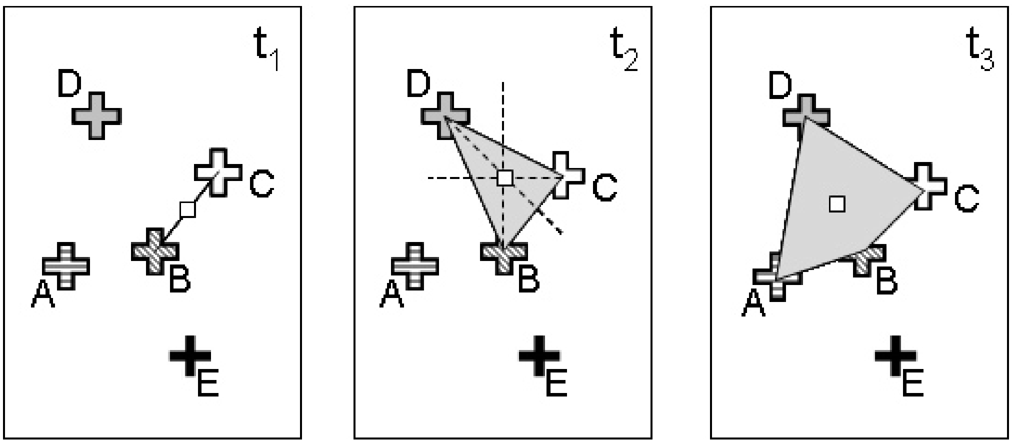


Raw Data **Cluster-Analysis** **Cluster Prototypes**

Figure 6.7 Sub-Cell Processing Model: Initialization

Positioning

The goal of the localization now is to determine a position using the cluster prototypes from initialization and knowledge of currently "seen" clusters. Figure 6.8 illustrates the positioning approach represented of the sub-cell process model. At an instant t_1 the LES identifies cluster prototypes B and C from the current sensor data, the location coordinates of which have been determined during initialization. In the simplest case the position of the LACBA client will be placed right in the middle of B and C (white square).



Using two cluster prototypes **Using three cluster prototypes** **Using four cluster prototypes**

Figure 6.8 Sub-Cell Processing Model: Positioning

At a later instant t_2 the system also identifies cluster D and positions the client in the centre of triangle B – C – D performing a triangulation using the arithmetic mean position of each prototype pair. If four cluster prototypes are identified the client is positioned in the resulting polynomial area. The positioning can be potentially further improved by considering additional cluster prototype features for a weighting of localization results, e.g. number and distribution of measurement points of particular clusters. Thus a cluster prototype position generated from many equally distributed measurement points should be better for localization than one with only few and biased distribution of measurement points.

6.2.3 Feature Extraction Principle

According to [RL02] the feature extraction tries to describe an occurring signal using a minimum number of parameters fulfilling the following requirements:

- The parameters have to be *deterministic,* i.e. two identical signals have to result in the same parameters.
- They have to be *de-correlated,* i.e. the information redundancy occurring in different signal features has to be minimal, if not zero. Therefore each parameter describes only one feature of the signal.
- The described parameters should only contain absolutely *relevant* information.

The features represented by the parameters can either be dynamic or static in nature. In the static case the signal can be represented by a single feature vector of parameters whereas a dynamic signal, a sequence of static feature vectors is used, i.e. small parts of the signal are extracted and arranged in sequence to each other. However, classification using dynamic features is more computationally intense and static features can only provide coarse descriptions of information-rich signals. Concerning initialization the rate of change in infrastructure is only very little in observed small area parts respective to the entire usage environment, e.g. WLAN-hotspot inner-city scenario. Thus we can assume a quasi-stationary approach, that is, in the time frame of observation during initialization new infrastructure is not being generated or removed in the observed environment part. Hence the degree of information change here is zero and static feature extraction approaches can be applied.

6.2.4 Cluster Analysis Methods

On realizing the proposed sub-cell processing model we will now present the statistical methods for the feature extraction as part of the cluster analysis for the reference location data generation (i.e. cluster prototypes) and position determination using this data.

Arithmetic Mean

A very interesting and simple method to determine the empirical middle position of a sensor measurement cluster is provided by an arithmetic mean. Considering M sensor reading vectors denoted by $\vec{x}_k$ where $k = 1..M$ the calculation is as follows:

$$\vec{m} = E(\vec{x}) = \frac{1}{M} \sum_{k=1}^{M} \vec{x}_k \tag{6.1}$$

In the simplest case the calculation of the mean would occur over the obtained GPS coordinates as part of each sensor reading of a uniquely obtained MAC address. However, this method would not generate a very accurate true location of the observed access point but rather a “virtual access point” position as seen by the majority of users from their possible residing locations in the environment. Clearly, the more measurement points are taken from possible locations encompassing the possible true location of the access point, the more accurate this method becomes (Figure 6.2). The biggest advantage of this method is the possibility of calculating the mean by means of recursion supporting the self-learning aspect as mentioned earlier. Every new sensor measurement captured for an already captured access point does not require a re-calculation of the mean over all previously captured sensor data. For calculating the new mean location of the access point, i.e. $\vec{m}_M$, only the previously calculated old mean value $\vec{m}_{M-1}$, the current sensor measurement data $\vec{x}$, as well as the number of overall captured measurement points are needed (hence information need to be stored for a particular cluster prototype):

$$\vec{m}_M = \left(1 - \frac{1}{M}\right)\vec{m}_{M-1} + \frac{1}{M}\vec{x} \tag{6.2}$$

Apart from the arithmetic mean method other similar related approaches can be used as part of the cluster analysis such as the geometric-, harmonic-, and quadratic-mean. As previous work has shown [Ipp06] the advantages (if any) compared to the standard arithmetic mean are only minimal in our described usage context hence will be disregarded within this thesis. A very promising improvement to the arithmetic mean could be a weighting of particular sensor measurement points using respective weighting factors, e.g. RSS, hardware issues, user reliability, etc. The weighted mean approach will be considered for future evaluation [NH94][Ng03].

Point of Maximum Received Signal Strength

An alternative to using the spatial properties of the gathered sensor measurement points would be to use the RSS properties instead. A very simple method of determining the possible true location of an access point from a sensor measurement cluster is to record the point-of-maximum RSS. The reliability of the determined RSS-based cluster prototype position increases with the number of measured sensor samples as well as effective physical “closeness” to the true access point position. However, multipath reflection and dampening effects need to be considered, especially in the outdoor street scenario and presence of natural obstacles such as building walls etc.

Median

Another suitable and simple method for both initialization and positioning is to use a median heuristic. The median cluster point position of a sorted sensor data vector (x_1, x_2, …, x_n) of n measurement datasets is calculated as follows:

$$\tilde{x} = \begin{cases} x_{(n+1)/2} & n \text{ odd} \\ \frac{1}{2}\left(x_{(n/2)} + x_{(n/2+1)}\right) & n \text{ even} \end{cases} \tag{6.3}$$

In relation to the arithmetic mean the median often provides a more meaningful cluster point value since it is much more stable concerning occasional sensor measurement outliers. But on the other hand, the median requires much more effort to compute since it requires the storage of all previously captured sensor measurements as datasets upon recalculating the median on sensor data updates (i.e. the median cannot be described by means of recursion).

Probability-Density Functions

As stated earlier, using standard probability density functions (PDF) can also possibly help to generate very good cluster prototypes and be used for positioning under certain conditions and assumptions. The PDF function itself can be generated from the datasets, or respective PDF parameters can be obtained and stored as reference location data by applying a commonly known distribution forms. In each case the determination of the most probable position which determines the centre location (i.e. most probable physical access point location) of the described cluster of sensor readings is of our main interest.

In the first case, histogram methods can be used to generate a unique probability distribution profile. A critical aspect here is the choice of appropriate measurement intervals. If the measurement intervals are equally spaced only the determined values and interval boundaries have to be stored. Furthermore, if the interval is chosen too big a very coarse quantisation of the observed space, i.e. access point signal profile occurs, and if chosen too little there is the danger of frequent and unwanted/unnecessary “zero” value estimates. The later aspect can be minimized by using the Parzen window method [Rus04] or [RL02].

The other case of assuming a commonly known distribution of measurements requires additional known properties of the measured and to be processed sensor data as already previously mentioned in section 6.2.1, e.g. characteristic of the sensor measurement distribution. The more information is known the more efficiently and accurately known distribution forms can be applied. In the simplest case, a normal (Gaussian) distribution of measurement points can be assumed [Hau03] which is rarely the case on applying sensor systems in complex environments such as offices and outdoor inner-city scenarios. Especially in the latter case the obtained sensor clusters will follow more elliptical distributions (i.e. conical structures). In order to obtain a more accurate position estimate the additional information on how the elliptical cluster lies in space is to compute its covariance [Ipp06].

As related research has shown [CCK05] it is again to be doubted that the overall localization performance gains compared to the simple statistical methods introduced in this section justify the additional efforts and complexity involved in applying probability-based heuristics in our proposed scenario. Hence our focus on the further analysis will be on the arithmetic mean, median, and point-of-max RSS methods as described above.

6.3. Analysis of the Sub-Cell Processing Model

Following the selection of appropriate statistical methods for both initialization and positioning, we will now define an experimental setup to test the raw sensor data capturing, feature extraction and cluster prototype building, as well as performing localization using these prototypes. Hereby we will also consider the particularities of the capturing and processing of raw sensor data potentially influencing the initialization and positioning methods, e.g. window of observation. Different combinations of methods will be evaluated in order to determine the best possible initialization and positioning combination. A real-life inner-city test environment with a characteristic WLAN hotspot situation has been chosen for testing.

6.3.1 Experimental Setup for Raw Sensor Data Generation

Picking up on the WLAN hotspot inner-city scenario from chapter 5 several independent "wardrives" (i.e. sensor capturing runs) have been carried out using different hardware configurations. Three test runs have been carried out on an identical route (approx. 13,8 km in length) over the duration of four months within a selected district in the inner-city part of Munich. Equally mixed residential and commercial infrastructure is found on the selected test route. Each test run was driven three times completely in alternating directions taking roughly 2 hours during varying traffic conditions. A map of the test route taken within the city can be seen in appendix section D5.

For the data generation two identical laptops equipped with the LACBA client software have been used (LC1 and LC2), each having identical WLAN cards with commonly available Orinoco[12] chipsets and GPS receivers (SiRF3[13] chipset), but different WLAN antenna configurations (i.e. LC1 has a build in internal antenna of the WLAN card and LC2 an additional externally connected antenna extension). The standard configuration and platform of the LACBA client prototype is explained in appendix section F4.

The external antenna increases the sensitivity of the WLAN card thus picking up more distant access points and weaker WiFi beacons, thus resulting in a larger number of captured access points. Before each run, the time clocks of each system were synchronized via a time server. The captured sensor data contains WiFi beacons RSS (signal and noise) values of currently visible and "cloaked" access points, their MAC addresses and ESSID, time stamp, as well as GPS (longitude, latitude, altitude), speed (mph), and GPS-Fix location information. The latter two values can be used to determine the mobility of the roaming clients as well as quality of the GPS reference signal used at a particular point in time. Sensor data readings were recorded every 100ms, cached on the client during the entire test run and uploaded into the community-based LP afterwards.

6.3.2 Feature Extraction

Before carrying out the reference location data building from the collected datasets using the chosen statistical heuristics, we have performed a statistical analysis on the captured sensor data in order to extract additional information on the quality of the raw sensor data itself. Table 6.1 gives an overview of the sensor measurements captured during the three test runs carried out at different points in time. The test drives were carried out during busy and relaxed traffic hours at different times of the day, also accounting for the phenomenon of people only switching on their WLAN access points in the evenings for internet access at home. In each case, the average speed was calculated using the GPS data included in the sensor measurements. The fairly low

[12] Avaya WLAN 802.11b PC Gold card, Lucent Orinoco chipset: http://www.avaya.de
[13] Navilock GPS NL-302U USB mouse: http://www.navilock.de

speed of roughly 12 km/h going by car is due to the incurred traffic conditions and frequent traffic light stops on the 13,8 km test route. In our scenario the varying traffic conditions and travelling speeds can be ignored if multiple drive-bys are carried out. Our tests have shown that performing at least two drive-bys at particular locations can compensate for a higher mobility on the amount of sensor data captured per access point, i.e. capturing by foot vs. capturing by car [Ipp06].

Date & time of test run	**Laptop setup**	**Av. speed (km/h)**	**# of collected sensor data points**	**# of unique APs**	**# of new APs**	**# of APs missing**	**Av. expansion, cluster diameter**	
							Lat. (m)	**Long. (m)**
2005-10-11, start @ 10:30	LC1	12,4	59050	1720	1720	N/A	48,2	33,3
	LC2		105676	2359	2359	N/A	53,3	41,1
2005-12-14, start @ 18:00	LC1	11,7	60532	1730	19	9	47,1	35,3
	LC2		111070	2373	25	11	54	41,9
2006-01-30, start @ 15:00	LC1	12,9	61041	1734	11	7	46,2	31,5
	LC2		113441	2377	13	9	55,1	46

Table 6.1 Statistical overview on sensor data captured during each test run

Clearly, by using an external antenna much more access points are discovered whose signal is normally too damped, i.e. its location is too far into the building and only weakly detected on the outside. In particular, this results into much more sensor data points being generated and a respective higher longitude/latitude expansion, i.e. bigger elliptical sensor clusters.

Altogether, a general increase of available access points was detected at each test run although a few previously captured ones were lost. The latter case could be due to the intermittent availability of access points as mentioned before, or occurring environment dynamics causing the respective access point signal to be dampened below detection.

Considering the average number of sensor data points available for each access point, a histogram has been generated from all test run datasets (see appendix section D1). A majority of access points are represented only by a few sensor data measurements, despite traffic light stops and multiple drive-bys. In 90% of all cases there are less than 100 sensor data packets per access point, and in 50 % of all cases there are less than 10 sensor data packets. The histogram representing the average number of access points seen respective to a set observation window over all test runs is illustrated in Figure 6.9. Knowing that the area underneath each curve adds up to 100%, 1-4 access points are visible 60% of the time (i.e. at window size = 0, one access point was seen ~17%, two at ~16%, three at ~ 14% and four at ~10% of all cases etc.). Extending the observation time to up to two seconds (i.e. win = 2) increases the average number of visible access points. This will become important again when determining the optimal window of observation and positioning method later on. There is an additional dampening factor whether the WLAN antenna (be it internal to the device or an additional external extension) is external of the travelling car or not, i.e. even more WLAN beacons are picked up. Experiments have shown that the dampening/shielding by the car results in roughly 20% decrease in picked-up WLAN access point beacons [Kar06]. Since we assume less than ideal/worst case conditions for evaluating the sensor data capturing, initialization, and positioning cases (i.e. heterogeneous hardware configurations and respective sensor measurement data spreading etc.) we can also assume that a majority of users might not possess external additional WLAN antenna extensions or external car

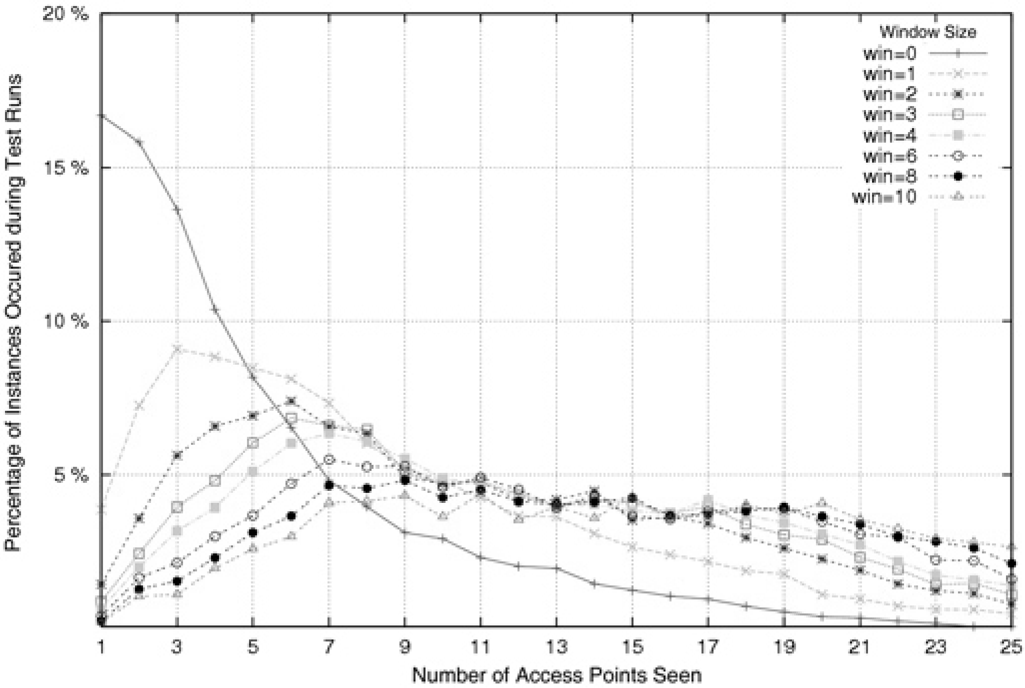


Figure 6.9 Observed access points at respective observation times during test runs

based hardware configurations. We can, however, assume that common mobile devices being carried along by people within vehicles or backpacks perform data capturing whilst for example moving to/from home and office locations.

6.3.3 Reference Location Data Building

For the cluster prototype generation and localization, the arithmetic mean, median and point-of-max. RSS methods have been used and implemented accordingly (see Perl script code in appendix section D2). Since the exact location of the WLAN access point is not known, we can assume that the point-of-max RSS method represents the nearest position to the actual physical location of the access point but not the centre location on how the user perceives it (due to multipath reflection, fading, etc.).

Furthermore, the user cannot occupy every coordinate position in space i.e. as in our scenario only the outdoor street positions. Bearing this in mind we can try and describe the access point from the user's perspective. This leads to the illustrated median and arithmetic mean approaches. Figure 6.10 shows a common case where the unique WLAN access point cluster prototype centre position has been generated from sensor data collected by all test runs. The results of the three chosen cluster prototyping methods are shown with respect to the deviation in longitude and latitude distances (in meters) on the computed arithmetic mean. We can see that the median and mean cluster centre positions deflect quite considerably from the point-of-max RSS determined position. Knowing the variations in how many access points are seen at a particular instant as well as computing the standard deviation of the sensor data measurement points within a prototype cluster itself can help determine the best possible initialization (i.e. cluster prototype) and positioning module combination for the best possible localization result.

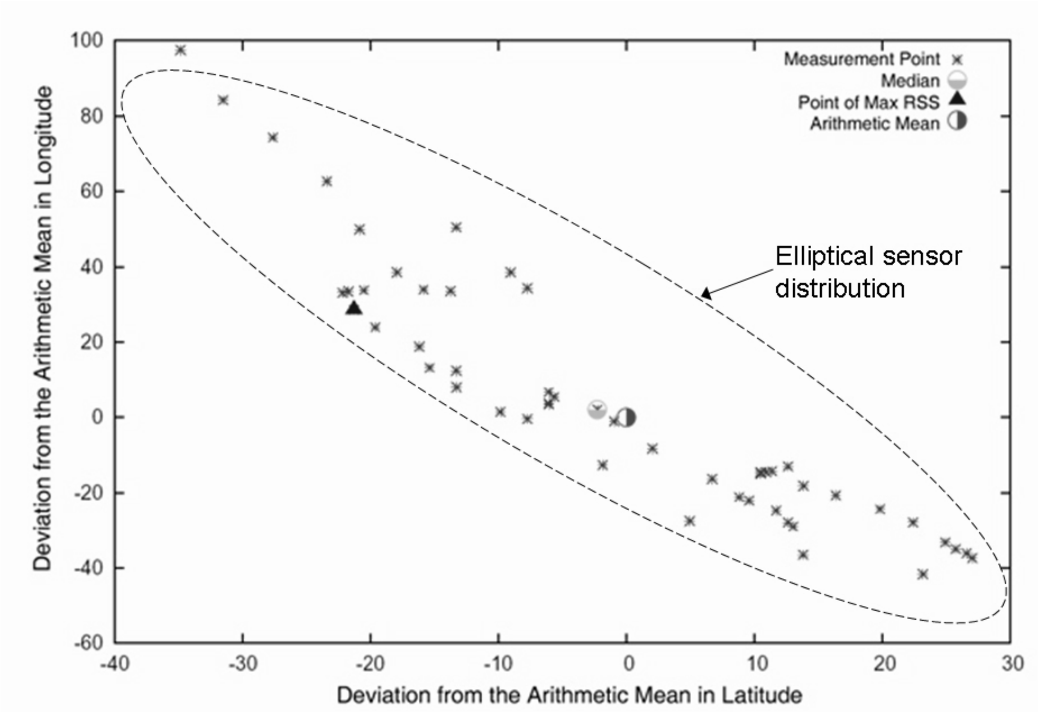


Figure 6.10 Example of a cluster prototype generated from collected sensor data

6.3.4 Evaluation of Different Initialization and Positioning Module Combinations

Following the sub-cell processing model as illustrated earlier we now have to look at the localization possibilities on the basis of the reference location data generated by the three cluster prototype principles. Considering the number and types of cluster prototypes used for positioning (see Figure 6.8), several localization possibilities arise but resulting in a discrete location coordinate with corresponding localization accuracy in the end. In the simplest and worst case only one cluster prototype is visible and discovered resulting to a classical proximity-based localization i.e. the position and coordinates of the user are that of the cluster point centre having an (in)accuracy of the determined respective cluster cloud area. The sensor measurement data expansion in longitude/latitude (see Table 6.1) determines the roughly elliptical area location model. If more than one cluster prototype is available, the user location is computed at the "centre-of-gravity" [Ng03] of all seen prototypes. The resulting intersection of cluster-cloud areas can be used for determining the relative accuracy of this localization.

Clearly, with a growing number of cluster points used for localization the number of localization possibilities increases, but using the centre-of-gravity approach the localization accuracy should increase as well as later analysis will show. The most suitable heuristics for performing this localization approach are the previously introduced arithmetic mean and median. The weighted mean which was introduced in section 6.2.4 could also be applied very effectively here by considering additional cluster prototype information such as the number of sensor measurement data points used in generating the cluster prototype i.e. use that as a "quality" weighting factor. For the evaluation within the scope of this thesis the focus will be on the arithmetic mean and median.

The Initialization–Positioning Evaluation Matrix

In the following we will evaluate six different initialization and positioning methods as indicated by the matrix in Table 6.2. The goal will be to identify the most optimal combination for our set outdoor inner-city scenario using the three types of cluster prototypes generated from the data of the three independent test runs.

Init. method… / Pos. method…	Point-of-max RSS	Arithmetic Mean	Median
Arithmetic Mean	1) PomRSS–Mean	3) Mean–Mean	5) Median–Mean
Median	2) PomRSS–Median	4) Mean–Median	6) Median–Median

Table 6.2 The evaluation matrix for initialization and positioning methods

For the subsequent evaluation the sensor data (i.e. WLAN RSS and MAC address, timestamp and GPS signal) of all test runs has been used to generate the cluster prototypes, and from it, the localization has been carried out with each LC test run dataset independently (Table 6.1). The obtained localization results were averaged and carried out for each combination listed in Table 6.2, hereby considering the GPS signal as ground truth. The average GPS error was 5,8m (CEP) and 28,1m (2dRMS) using a Garmin[14] GPS device.

The Optimal Window of Observation

The localization process starts with the selection of appropriate sensor data information. Depending on the speed of the tracked LACBA client it might be an advantage to not only use the current sensor data but also history data within a certain timeframe. This windowing function has the advantage of potentially having more cluster prototype information available for positioning hence improving the localization accuracy. However, the observed timeframe has to be within a certain limit before the “motion

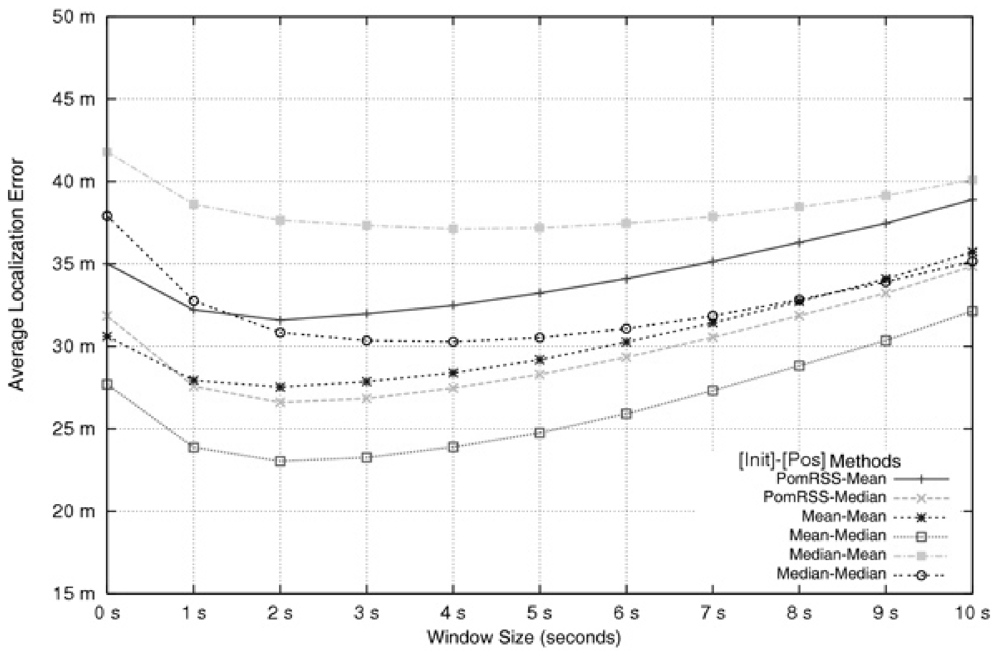


Figure 6.11 Average localization error vs. observation window size of init./pos. combinations

[14] Garmin: http://www.garmin.com

blur" increases too much, i.e. the position of the tracked LACBA client having changed too much within the chosen timeframe. Figure 6.11 shows the optimum window timeframe of sensor data observation for all possible initialization and positioning methods listed in Table 6.2. as well as mean localization accuracy obtained throughout a test run with respect to the GPS reference signal used. The average speed was determined to be 12,9 km/h. The window size $t - n$ has been varied from $n = 1 \ldots 11$s, looking at the resulting average localization error using only the currently valid sensor data (i.e. $n = 1$), up to including several seconds of cached sensor data (i.e. $n = 11$s). As can be seen from Figure 6.11 a minimum window size of two seconds has been identified for the average mobility of 12,9 km/h whereby the Mean-Median combination seems to perform best. In every case using observed sensor data of less than three seconds seems to improve the localization accuracy by roughly 15-26%. In other words, using sensor data over a small observed time interval increases the potential number of cluster prototypes seen, i.e. information available for localization hence improves the localization performance. The motion blur effects become more dominant and start to increase on having a window size greater than three seconds. We will use the chosen optimum window size of two seconds for our further analysis.

The Minimum Access Network Density

As discovered correctly not only the window size of observation can improve the localization but consequently also the amount of information available, i.e. cluster prototypes used for localization. Similarly, we should be able to determine the minimum required cluster prototypes ensuring an overall satisfactory localization performance. Using the same test approach as used for obtaining the ideal window size and a respective window size of two seconds the dependency on the number of visible access points and resulting localization performance can be seen in Figure 6.12.

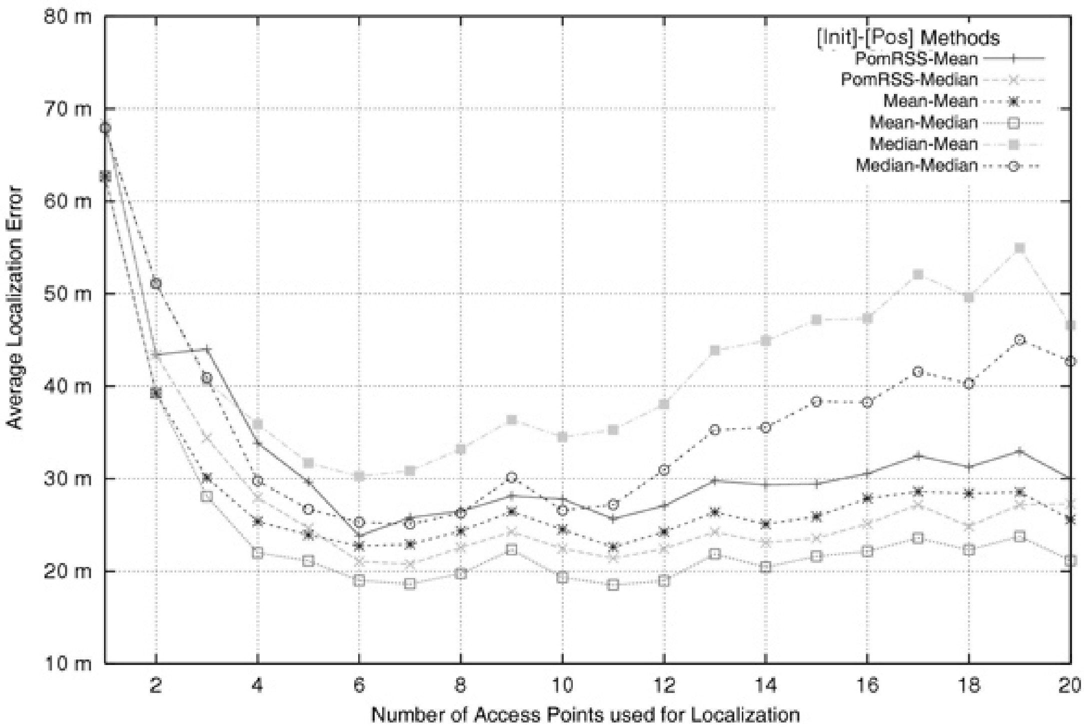


Figure 6.12 Average Localization Error vs. number of APs used for positioning

The results show that a localization error minimum is reached when the number of access points used for positioning is greater than three. As already mentioned at the beginning of this section, a single cluster prototype occurrence provides a cell-based accuracy of roughly the expansion of its composed sensor measurement data. On the condition of two or more cluster prototypes the cell area can be approximated to be circular in nature. Under this assumption the Mean-Mean combination in particular describes the average localization error produced by the superposition of currently seen cluster prototypes.

In general, apart from the Median-Mean and Median-Median combinations, the minimum average localization error occurs in the region of four to nine visible access points which increases steadily as more and more access points are seen. This exemplifies the problem of channel interference and increased signal noise as more and more access points are within radio coverage of each other. This could also be a likely reason why the Median-Mean and Median-Median combinations perform particularly bad as more access points are visible. It is also possible that the initialization error accumulates when using the Median filter for initialization. However, these are only hypotheses and need to be verified with ongoing research.

Summary and Conclusion

According to Figure 6.12 the proposed ideal combination of initialization and positioning methods for our chosen scenario is not Mean-Mean but Mean-Median instead. Looking at the statistical analysis represented by Figure 6.9 the minimum requirement of on average 3-4 visible access points at any point in time is more or less given. It can be assumed that all tested combinations perform similarly well in the given scenario since the occasions of having more than 13 visible access points are very rare. A further note should be given that the initialization and positioning combinations have only been experimentally tested and proven, but not mathematically. This can be made up for as part of ongoing research.

Using the arithmetic mean for initialization provides a very simple and adaptive approach for implementation in such an inner-city scenario. The achievable data compression and respective storage of reference location data is very high (i.e. only three parameters need to be stored from all generated sensor data, see section 6.2.4)

The positioning using the Median filter is also a very elegant approach, since only two median values are needed for the localization process (i.e. the longitude and latitude information). For the Median generation, the needed to be sorted list is very short and is composed of only the cluster prototypes currently identified. Assuming a window size of two this results on average to nine access points for each localization event [Ipp06]. The overall average localization performance computed for each initialization and positioning combination is summarized in Table 6.3, hereby showing that the Mean-Median performs best with an average of roughly 23m localization error with respect to GPS. The source code for the corresponding initialization and positioning methods of each combination is given in appendix section D3.

The initialization and positioning processes for the evaluated combinations have been visualized using Google Earth[15]. Hereby, the generation of the cluster prototypes from

---

[15] Google Earth: http://earth.google.com

the provided test run data as well as the localization performance with respect to GPS are shown using screenshots in appendix section D4.

Initialization – Positioning Combination	**Overall average localization error (m)**
pomRSS – Mean	31,1
pomRSS – Median	25,9
Mean – Mean	26,3
Mean – Median	23,1
Median – Mean	38,2
Median – Median	30,5

Table 6.3 Summary of localization error for each init./pos. combination

6.4. Summary

In this chapter we have introduced a flexible LES framework enabling the realization of various LP solutions. A component model view was given and every processing step for the data preparation, initialization, and positioning explained giving example configurations for particular LP solutions.

Continuing our community-based WLAN hotspot scenario for inner-city locations from chapter 5 suitable initialization and positioning methods were introduced and evaluated. Cluster analysis methods were chosen for the initialization and positioning, which enable a lightweight and adaptive solution fulfilling the requirements for our proposed scenario. A sub-cell processing model was introduced by capturing transmitted WLAN beacon traffic, geo-coding the access point locations using a common location reference signal (e.g. GPS) as well as performing localization using the generated access point cluster prototypes (i.e. virtual access point locations). The superposition of multiple cluster prototypes and the resulting intersection represents the localization error. Several combinations of initialization and positioning methods have been experimentally evaluated achieving an average localization error of roughly 23m, with respect to the GPS reference used as ground truth using a Mean-Median setup. In all, the WLAN hotspot-based localization and introduced sub-cell approach provide a competitive and cheap localization method alternative to GPS for a multitude of mobile (and legacy) devices using the widely distributed WLAN communication interface. The FCC standard [FCC06], also known as the E911 directive, specifies 100m for 67% of calls and 300m for 95% of calls for network-based localization solutions as well as 50m for 67% of calls and 150m for 95% of calls for handset-based solutions. Therefore, considering the imposed GPS location reference error the proposed sub-cell approach for the inner-city scenario clearly fulfils the E911 directive.

Chapter 7
Location Service Discovery

In the previous chapters we have described the structure and functionality of LACBA, showing the feasibility and potential performance of the innovative community and collaborative value generation approach for location service provisioning. In this chapter an equally novel and innovative location service discovery mechanism will be presented which provides a powerful and essential feature of the LACBA concept.

At first, several location service discovery mechanisms will be discussed by highlighting the peculiarities governing the LACBA concept. Following this, a novel peer-to-peer based location service discovery mechanism and hierarchy model will be introduced. Hereby, the processes relating to the location service registration, sensor data processing, and location service discovery using LACBA client sensor data will be explained using corresponding examples.

7.1. Evaluation of Location Service Discovery Mechanisms for LACBA

The LACBA principle is a radically different approach in that it efficiently integrates mobile clients in the environment data capturing, reference location data building and location service provisioning processes. This principle should be able to be applied in all possible usage environments in question, i.e. indoor and outdoors, where there is an abundance of wireless networks present. For every usage environment, a dedicated LP offering a suitable location service should be present as shown in Figure 7.1. As we have also specified in our LACBA principle, the LP in a particular usage environment is responsible for the location information generation and provisioning in its target usage environment.

In the scenario as described in Figure 7.1, the heterogeneous wireless network landscape will consist of various publicly and privately hosted WLAN hotspot locations. A community-based LP as described in chapters 5 and 6 can geo-code this wireless network landscape and use it for localization purposes. Similarly, indoor areas such as the TU Munich, a museum or airport can use their local WLAN infrastructure to offer a location service via an indoor-based LP. In many cases WLAN access points at the edge inside of buildings are registered in an outdoor community-based LP as well. Such access points can function as “geographic boundaries” (so called “edge networks”) when going from outdoors into buildings e.g. a museum or a university campus. The

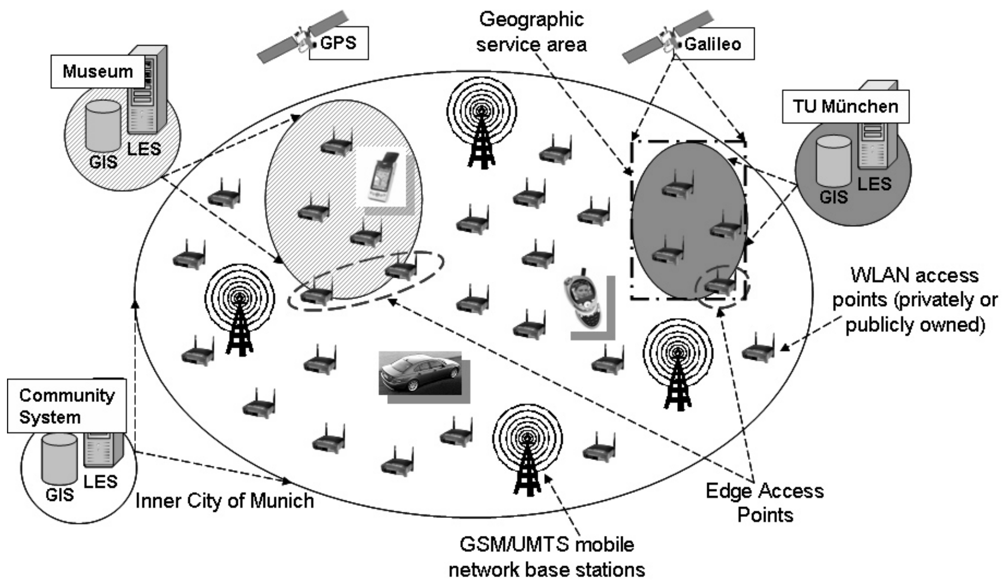


Figure 7.1 Location platforms in heterogeneous wireless usage environments

community system at the city level might contain several thousands of geo-coded WLAN hotspots in its database. Groups of these WLAN networks could belong to shopping malls, museum or university campuses. If referenced accordingly, they can be used to indicate the presence of a respective LP for this geographic area. The uniquely determined network IDs can be used for identification purposes, as we will show later.

Now, the critical questions arising in such a scenario are as follows:

- How do we determine which LP is the most suitable at any point in time? And more importantly:
- How do we discover their presence?

Concerning the first question, the service areas of the LPs need to be known and a hierarchy modelled. The hierarchy is necessary for handover and fallback mechanisms due to the availability of LPs in the environment (see Figure 4.8, section 4.4.2). If available to the user, an indoor WLAN RSS-based LP should provide more accurate location information than an outdoor GSM Cell-ID based LP. Thus, if the indoor system is not available to the user, the outdoor LP needs to be used as a fallback solution. The hierarchy itself needs to be known by all clients or needs to be made available via a centralized lookup directory e.g. UDDI.

The service areas itself can be determined by physical location and location inference indicators, e.g. unique network identifiers. The respective geographic service area can be determined by the IP address of the mobile client (e.g. IP-localisation), the country code identified from GSM network (via CCNMC, LAC, or even GSM Cell ID), or by identification of currently “seen” WLAN hotspots. Physical location indicators could for example rely on GPS/Galileo-based sensor sources. Geographic areas and boundaries can be defined (see Figure 7.1) and handoff signalled to appropriate LP entities using GPS receiver information from the mobile client. Manual user input in forms of an address could be also considered a physical location indicator.

The second aspect on how to discover the presence of the LP is even more critical. Hereby, it is vital to know that clients might not have access to the local network providing the location service in the first place. Furthermore, it is crucial to decide whether to rely on existing network service broadcasting and advertisement methods and protocols (e.g. DHCP, Mobile IP, SLP), or network independent mechanisms. Possible approaches will be evaluated in the next sections.

7.1.1 Requirements on Location Service Discovery in LACBA

Looking at the scenario as described in Figure 7.1, the location service discovery in LACBA should fulfill the following requirements:

- ***Network independence:*** The location service discovery should be independent of particular access networks and network functions (e.g. LP advertisement or client solicitation broadcasts). The client does not necessarily have access to the local network infrastructure resources
- ***Global availability:*** The location service discovery mechanism should be uniformly available on a global scale.
- ***Location platform independence:*** LPs should not be interdependent on each other. They should be able to join and leave the "location service resource network" dynamically.
- ***Flexible location platform discovery using sensor data from environment:*** The location service discovery should be performed based on sensor data captured by mobile clients, i.e. using both physical location and location inference indicators as described above.
- ***Location platform flexibility:*** At least one LP must be available to a mobile client at any point in time, serving as a fallback if necessary. This also implies knowledge of respective LP hierarchies.
- ***Location platform scalability:*** The location service discovery mechanism should scaleable in supporting a global network of distributed LPs, though focusing location service discovery to local living environments.

The location service discovery mechanism which was partly portrayed and modelled in chapter 4 has to fulfil these requirements on the nature of the sensor data captured by the mobile clients in LACBA. In the following we will concentrate solely on the issues involving the location service discovery mechanism, describing the coherences to the other previously explained LACBA processes where applicable.

7.1.2 Means of Performing Location Service Discovery

There are several approaches to allow for the discovery of a location service suitable for the local environment. They can be categorized as follows:

1) The local infrastructure or LP can advertise its presence via beacon broadcasts over the local network.
2) The mobile client broadcasts service solicitation messages via the local network.
3) The mobile client performs a query via centralized root directory.
4) The LPs know about their hierarchy and inter-work accordingly to provide a location service without the mobile clients' interaction.

In 1), the LP can send out proprietary beacon messages [Rot04] or rely on existing standardized methods (e.g. Mobile IP [TW04]) in order to advertise its services in the local network. Using a piggybacking approach via Mobile IP requires additional infrastructure in the respective networks, i.e. Foreign and Home Agents, which normally solve the purpose on supporting network roaming. Using such an approach might be feasible in local network scenarios, but will be problematic in a global, homogeneous scenario when depending on suitable communication networks and cross provider solutions, e.g. additional cell broadcast in mobile operator networks.

The same is more or less true in the case of 2). Here, the local LP responds to the solicitation message of a mobile client (e.g. UDP or Multicast IP). Each client request must contain the location information of the mobile client (e.g. GPS) from which a candidate LP can deduce whether to be able to serve the client or not.

In 3), the client performs a form of broadcast or lookup query to a centralized directory agent, which transfers additional information to the client on the network and services available there (e.g. using DHCP or SLP [TW04]). Apart from the local network connectivity information, DHCP can for instance deliver additional information on local print or other server address information. In SoA any client or server can query a central registry (the UDDI) for requested services, provided all relevant location services have registered with it. Still, the client must provide a location indication and the service area information of each location service must be accessible by the directory as well or known by all participants requiring a location service [Leo03][Rot05]. These hierarchies of location services must be frequently updated due to potential selected availability.

In order to potentially solve the dilemma of the mobile client needing to know up-to-date location service hierarchies and LPs providing at least an initial point of contact, LPs itself can exchange their service area information so that each LP knows which LPs share potential service area overlaps. In [Rot05], a respective inter-LP protocol was specified for this purpose. Although this approach supports network independency of the mobile client, it increases LP interdependencies.

7.2. A Novel DHT-based Peer-to-Peer Approach for Location Service Discovery

Coming from the special requirements of the LACBA location service discovery and possible means of realization, an approach was devised organizing distributed LP resources in a hierarchical DHT-based peer-to-peer (P2P) system using an arbitrary number of independent Chord [SMK01] rings, structuring the corresponding location services by mirroring the resolution/granularity of provided location information, i.e. country, city, building and room level. The concept was adapted and extended from [ZET06] and was first conceptualized by us in [ZI06b]. The original concept behind [ZET06] was to introduce topic spaces, clustering participating nodes according to particular content topics, thus optimizing routing and lookup queries to these specific groups, therefore improving overall system stability and scalability. Figure 7.2 illustrates the location service discovery principle using a single Chord ring structure. We have previously introduced the P2P-based overlay network architecture as the ERN in chapter 4. The basic principle is that location service queries are performed on keywords using the sensor data to appropriate nodes within the overlay network structure. Nodes containing content keyword references related to target reference location data stored by corresponding LPs return this information to the requesting peer, which then can use this information to lookup the appropriate LP offering the needed location service.

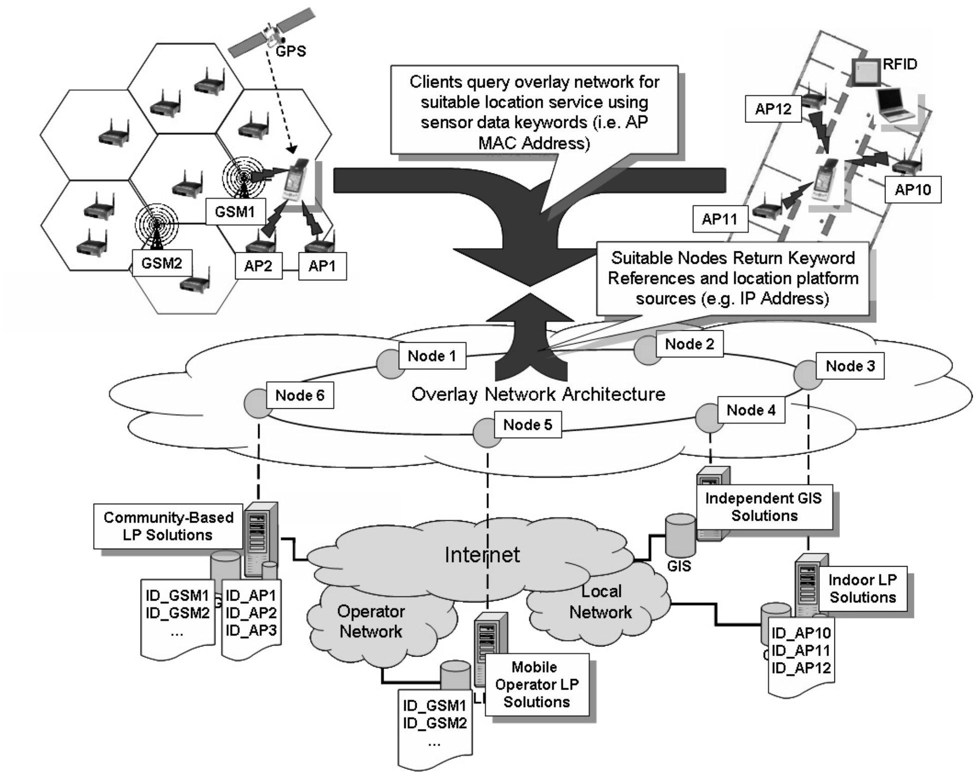


Figure 7.2 The location service discovery principle based on a P2P overlay Chord ring structure

In this section, we will show how this approach solves the requirements as set out in section 7.1.1, illustrating its detailed workings. We will give an overview of the hierarchical LP ring model, showing the registration and deregistration of LPs and how unique reference location data is created from sensor data captured by LACBA clients in corresponding usage environments. Finally, we will show location service discovery examples on how a mapping of captured sensor data (e.g. WLAN ESSID or GSM Cell ID) to candidate LP entities storing the respective unique reference location is performed.

7.2.1 DHT-based P2P Systems and the Chord Protocol

Classical P2P systems and applications are distributed systems without any centralized control or hierarchical organization, treating all participating nodes as equal (unlike classical client-server relationships). They provide a self-organizing overlay architecture independent of underlying communication networks with the goal of finding and using distributed resources. Two main categories of P2P systems exist: ***structured*** (e.g. Chord protocol [SMK01]) and ***unstructured P2P systems*** (e.g. Gnutella [KM06]). A discussion on the advantages and disadvantages of respective systems can be found here [Hof06].

The structured P2P system approach using the Chord protocol is best suited fulfilling the location service discovery requirements set out in LACBA. Contrary to unstructured

P2P, the network topology and position of specific content are determined by the protocol using Distributed Hash Tables (DHT). Hence the position of a node and its desired content are known a priori, and thus can be contacted directly (i.e. routing instead of flooding). This provides a resource efficient location service lookup mechanism keeping signaling overhead to a minimum, which is especially important if frequent location service lookups are to be performed e.g. during a navigation session.

On the downside, effort is required to manage the structure of the P2P network and content references need to be built and shared upon entry and leaving of a node. Thus high node fluctuations (or churn rate) increase signaling overhead potentially causing instability [Eic05]. LPs can join and leave the overlay network dynamically. But the expected churn rate as well as total number of potential participating LPs is seen as fairly low compared to typical P2P file sharing scenarios having several thousand and high fluctuations of participating peers.

The Chord protocol distinguishes itself from other P2P lookup protocols in its simplicity, provable correctness and performance. It is fully distributed and especially addresses load balancing, scalability, availability (i.e. actuality of content) and a flexible naming scheme. Actuality is very important in the context of initialization and self-optimization i.e. availability of cluster prototype references and distribution of location reference updates. The Chord approach was chosen based on these underlying features (i.e. mainly its simplicity) and purely on the aspects of being capable of realizing the LACBA location service requirements. Other structured P2P protocols such as Kademlia, CAN or Pastry have different approaches on structuring their ID space or routing, but show similar performance to Chord [SMK01]. An evaluation of different P2P approaches and performance aspects are not scope of this work and left to future research. In all, the scalability of such structured P2P system protocols was shown to be very high [ZZ03].

7.2.2 The Chord-based Location Platform Hierarchical Ring Model

The basic principle behind Chord is a fast distributed computation of a hash function mapping keys to nodes which are responsible for them. By means of *consistent hashing* [KLL97] (e.g. using SHA-1 [NIS95]), each node and key is assigned a unique m-bit identifier, generating a node identifier by hashing the nodes IP address and a key identifier hashing the key itself. The keys and nodes are ordered in an identifier circle organizing participating nodes in a ring type structure. We will show the hashing principle and assignment later on in more detail. In effect, a single ring structure would be sufficient to accommodate all nodes and possible content keys, provided the identifier length m is large enough to make the probability of two nodes or keys hashing to the same identifier virtually impossible. As mentioned before, considering an N node system each node maintains routing information on $O(\log N)$ other nodes and resolves lookups using $O(\log N)$ messages to other nodes [SMK01]. Considering a world-wide network of LPs, employing ring hierarchies in the context of organizing the services areas. i.e. geographic responsibilities of participating nodes (see Figure 7.3), increases scalability and provides the following advantages:

- Location service lookup is restricted to certain sub-rings, hence LPs relating to the current locality of the LACBA clients (e.g. city or building level). Hence this allows for a faster lookup and relieves other nodes from processing search requests, i.e. reduction of signaling overhead in that the number of nodes per ring is drastically reduced.

- Grouping LPs representing an environment granularity level (e.g. country, city, and building) and resolving of the geographic relation in the respective ring hierarchy (e.g. Germany → Munich → TU Munich) helps to realize vertical handovers between ring hierarchies as we will see later.

In Figure 7.3 we have chosen a three layer ring model representing the most logical environment-based grouping for LPs. The root ring represents the LPs providing a location service (location information and GIS content) on a country wide level, e.g. GPS and mobile operator-based solutions. The second layer classifies the LP relating to city environments, e.g. community-based systems using WLAN hotspot or GSM Cell ID-based approaches, or even mobile operator solutions providing more city relevant information.

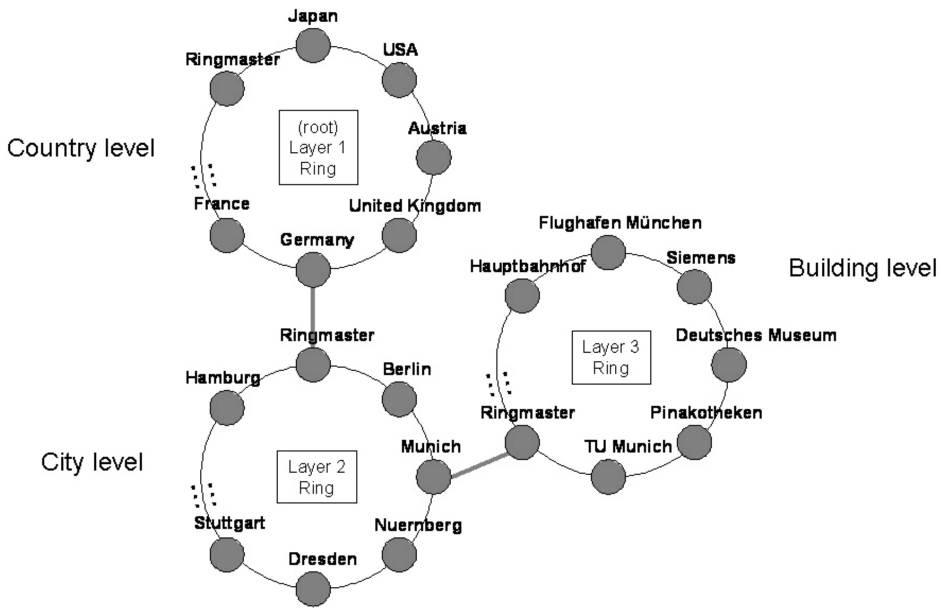


Figure 7.3 DHT-based P2P hierarchical Chord organizational model for location platforms

The lowest ring layer refers to building-based environments, e.g. indoor WLAN RSS-based LPs at airports, factories or museum sites, etc. It is possible to define even lower sub-rings providing logical context resolution on a building or even room level when thinking about allocating proximity-based sensor information, e.g. RFID tags. The potential of this approach will become more apparent later on.

In order to allow for location service lookups across different ring hierarchies, additional *ringmasters* are introduced within the chord protocol [ZET06]. Every new chord ring is initiated and managed by its own ringmaster. This role is taken on by a node initiating the new sub-ring thereby retaining its original function as a conventional Chord ring node in the new rings parent ring. New ringmasters send respective association messages to their parent ringmasters, thus each ringmaster is aware of all its child ringmasters and therefore enables cross ring hierarchy routing.

The LCS defined in the LACBA framework also assumes the role of a node participating in the Chord ring performing the sensor data keyword queries. The position in the hierarchy can be assumed at the root/country ring level, although it can also be positioned at an arbitrary ring level. More importantly, the LCS assumes the role of a static *supernode* [Zha02] to which all mobile LACBA clients connect to the hierarchical Chord ring structure. Similar to the LCS the use of a dedicated bootstrap server is proposed as an entry point for new LP peers and general search queries in [ZET06], performing AAA and billing for both LPs and mobile clients.

The namespaces of each Chord ring are independent of each other. Mapping of keywords can either be performed using “qualified” and “unqualified” name keys [ZET06]. Unqualified name keys refer to the standard keyword and content lookup rules of Chord. Qualified naming considers the precise path and naming within the hierarchy tree structure, thus performing content/keyword and node insertions as well as lookup at the appropriate ring hierarchy. This precise routing of INSERT and QUERY messages at and across several ring hierarchies is realised using the XPath language. We will explain this more thoroughly later on.

7.2.3 Location Platform Registration

The conventional Chord implementation was extended by several independent Chord sub-ring hierarchies representing certain geographic regions, i.e. service areas. However, each Chord ring can be treated as a closed Chord system, hence node participation can be handled in the conventional manner as specified by Chord. LPs are assumed to know their designated service area a priori and join ring hierarchies at the appropriate level. In the case that an appropriate ring level is not yet present, new Chord rings can be spawned respectively as mentioned earlier. This section will focus on the LP registration process at an existing Chord sub-ring level.

In order for the Chord ring to maintain a stable state, each node in the ring must know about its direct predecessor and successor nodes and for every content key k generated, the node with the node ID closest to the respective key IDs is responsible of maintaining the content keys, i.e. node *successor(k)* is responsible for k. While the former aspect ensures that the routing within the ring functions, the latter aspect ensures that the content can be found within the ring. To ensure that the predecessors and successors of each node are up-to-date, each node periodically runs a *stabilize* function querying its neighbor nodes on their predecessors respectively.

As mentioned earlier each node generates its unique node ID key hash from its IP address. These node IDs are ordered clockwise in ascending order in the ring in modulo 2^m (m being the identifier length). Having computed the unique identifier key hash for LP node Q (hash ID = 1), the registration procedure is as follows. Figure 7.4 illustrates the LP node registration at the city level where $m = 3$:

1) Assuming the identity of an arbitrary other node Z in the ring is known to the new node Q, node Q asks node Z to find its successor node, i.e. the node whose ID succeeds node Q’s ID the closest.
2) Node Z queries the ring network and determines the closets node successor to be R (hash ID = 3) returning the result to node Q.
3) Node Q notifies node R to update its predecessor routing neighbor information.
4) Node P runs its periodic stabilize function querying node R for its current predecessor.

5) Node *R* returns its current predecessor to be node *Q*.
6) Node *P* updates its successor information to be node *Q* and notifies *Q* of itself being *Q*'s predecessor.

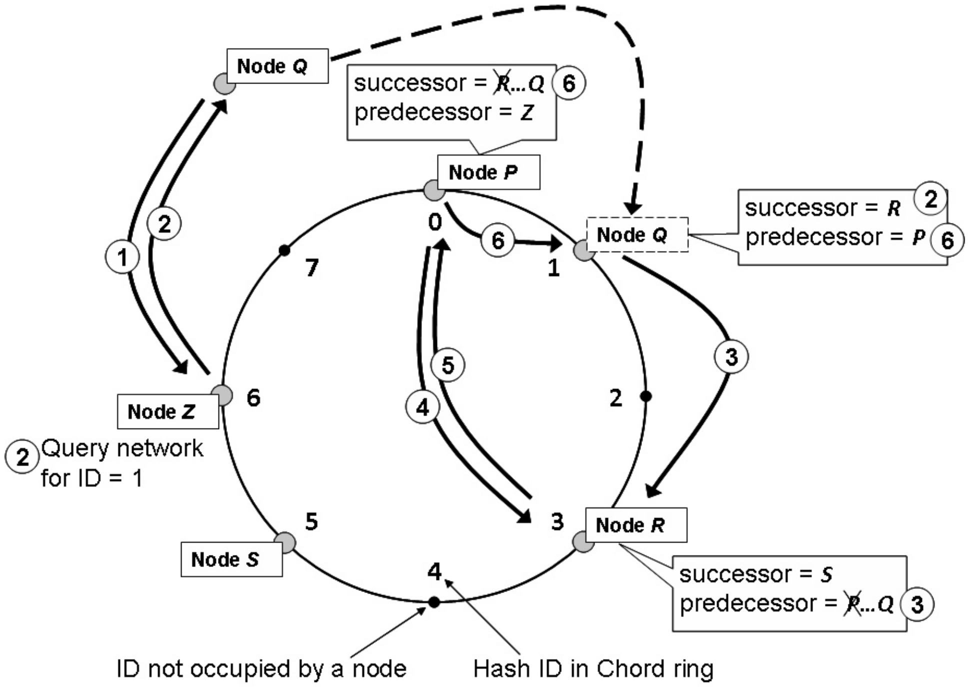


Figure 7.4 Location platform node registration in pre-established Chord ring network

Once the registration is complete the content keys are redistributed between the nodes accordingly. The content key (i.e. reference location data reference keys) distribution will be explained in the next section.

A node leaving the ring notifies its predecessors and successors accordingly, redistributing the content keys respectively. An unexpected node failure is discovered during a stabilize operation. Hereby the nodes maintain successor lists contacting follow-up successors after respective stabilize timeouts. During a lookup operation the node can find alternative routes via its finger routing table entries as we will show later. Other problems regarding failures and possible remedies during join operations are discussed here [BSH05].

7.2.4 Initialization: Generating and Allocating Reference Location Data Keywords

Content references in Chord are generated by performing the same hashing function which is used when generating the node ID hash from its IP address. As already previously mentioned the content references in LACBA refer to the cluster prototype reference location data generated during initialization of a LP using the client sensor data captured by the LACBA clients. Hereby the content hashing is not limited to the cluster prototype datasets of corresponding LP implementations, but also other client/user identification information needed by certain LP solutions (e.g. the MSISDN used by mobile operators), or even user access credentials of a particular authentication directory (e.g. LDAP authentication via an active directory system). Other types of sensor information such as fingerprint, facial or body temperature profile information are supportable as well. In essence many possible user identification and locating themes can be realized in this way provided the participating systems support the mapping process and discovery mechanism of the hierarchical Chord ring structure. Figure 7.5 illustrates the content key generation using the content and node information, of which being capable of processing the content data and/or providing additional services respectively.

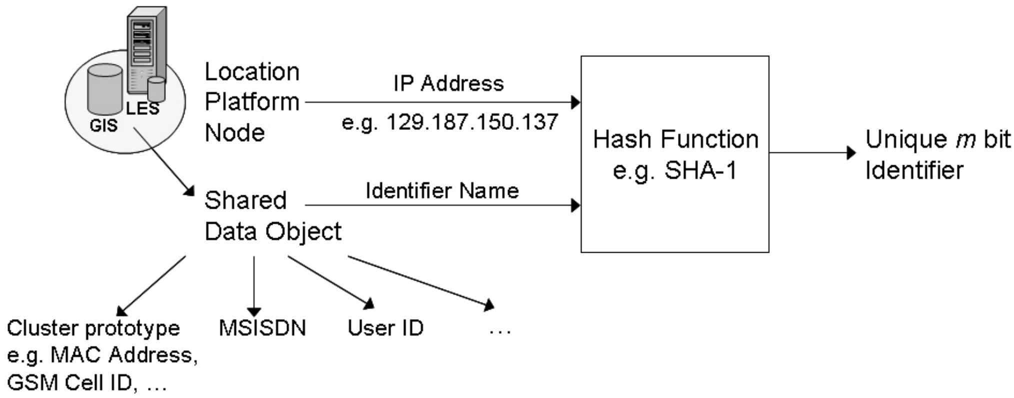


Figure 7.5 Content key generation according to the Chord protocol approach

A particular problem on using DHTs and the Chord approach is the use of sensor data which cannot be easily mapped to an explicitly defined keyword e.g. GPS sensor data. Basically, determining a LP from a single 2D longitude/latitude coordinate set without additional radius definition restricting and defining a service area would not work properly without at least an iterative query on a very large set of coordinates. We have developed a solution to this problem by describing 2D service areas using unique keys by indexing uniform geographic area segments similar to the UTM- coordinate system [BVV02]. This transformation function allows for the mapping of a (Latitude, Longitude) $\mapsto$ ID and ID $\mapsto$ (Latitude, Longitude) pair, hence processing of GPS coordinates in the proposed DHT Chord solution [Ipp06][Pue06]

The content key generation and mapping in the respective Chord ring structure can be performed after the LP node registration. Similar to new file sharing peers participating in a file sharing P2P network, a one time effort of hashing and mapping of all shared data objects (e.g. the entire reference location data pool) has to be performed until all shared data objects can be found by all participating peers in the network via keywords. Any incremental changes and additions of objects to the shared data pool are hashed and mapped on the fly as they are being created.

Using the community-based LP example providing sub-cell localization based on WLAN hotspot information, the hashing and mapping of respective cluster prototypes in a city level Chord ring is illustrated in Figure 7.6. On a ring using 3-bit hash record identifiers, hash records are generated for the shared data object (i.e. cluster prototype dataset) using the LPs IP address as well as keyword hashes referencing that particular shared data object hash record, i.e. a unique MAC Address and its ESSID. According to the mapping rules and procedures of the Chord protocol [SMK01], the respective keyword and shared data object hash key record (KR) are associated and stored with the node having the closest succeeding hash node ID.

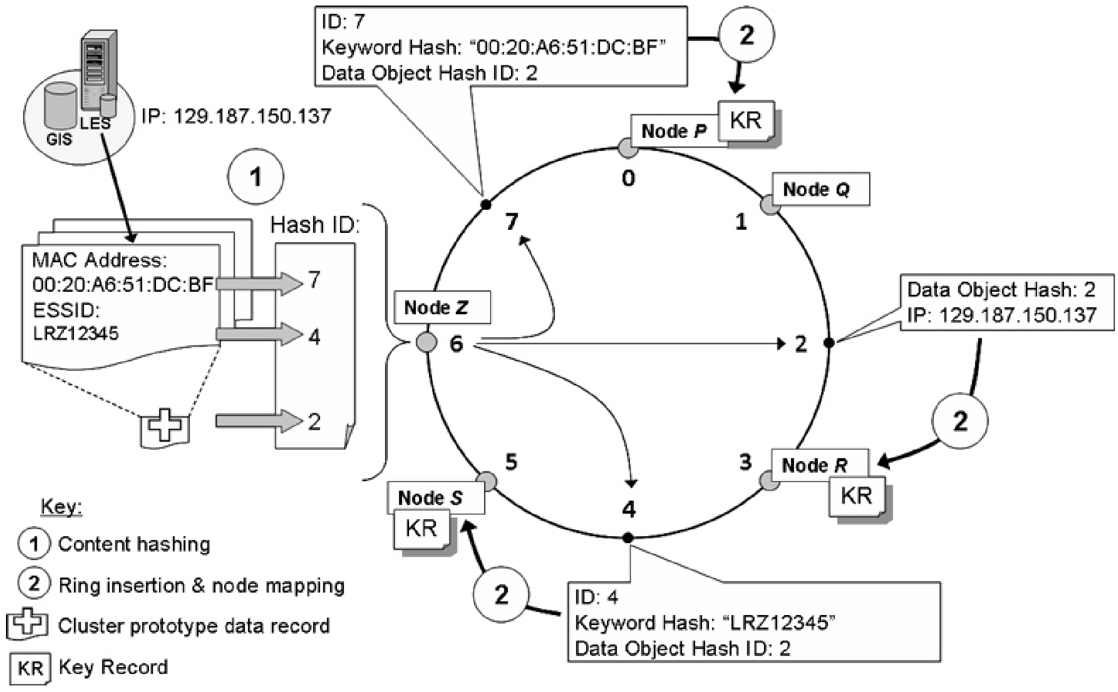


Figure 7.6 Reference Location data key record generation, ring insertion and node mapping

7.2.5 Location Service Lookup Using Client Sensor Data

Performing a location service discovery search using sensor data which resemble a MAC address and/or ESSID as keywords results in the nodes holding the corresponding KR reference to respond and return the appropriate LP source information (i.e. content link). Consequently, each node in the ring would be queried sequentially until the appropriate KRs are discovered. However this would not scale well for very large rings resulting in an average lookup of *n* nodes taking ½ *n* hops. Therefore Chord uses so-called *finger routing tables* providing lookup shortcuts through the Chord ring which are computed and stored by each node, resulting in an average lookup in $\frac{1}{2}\log_2(n)$ hops [SMK01]. Using *m*-bit hash record keys, each node has *m* finger entries in its routing table. By using the nodes ID in the ring, each finger entry shortcut *i*, where $i = (0 \ldots m - 1)$, to another KR or node ID in the ring is computed by:

$$(ID_{Node} + 2^i) \bmod 2^m \qquad (7.1)$$

Figure 7.7 continues the example from Figure 7.6, showing the location service lookup performed by node *Q* (i.e. a super node) using the LACBA client sensor data for the keyword search. Since we have a 3bit key length every node in the ring has three finger shortcut records ($i = 3$) to other nodes in the ring stored in its routing table. For each *i* the shortcut to an appropriate ID is computed using formula 7.1, and the node column indicates the appropriate node successor in the ring responsible for this ID value.

Using the captured sensor data information of a WLAN hotspot with the MAC address "00:20:A6:51:DC:BF" the location service discovery within the ring would be carried out as follows (Figure 7.7):

1) Node *Q* has computed the ID hash of "7" for the keyword MAC address "00:20:A6:51:DC:BF" and performs the Chord lookup operation. The routing table of node *Q* indicates the furthest shortcut hop to ID "5", i.e. node *S*, thus

passes the query to that node. Node *S* has a routing table shortcut entry to ID "7", i.e. node *P* having ID "0", thus forwards the query to that node respectively.

2) Since node *P* holds the KR for the keyword hash of the MAC address, it returns the appropriate data object hash ID "2" to node *Q*.
3) Node *Q* looks up the shortcut entry for ID = 2 in its routing table and queries the target node R for the data object KR.
4) Node *R* returns the source IP address information to the requesting node *Q*.
5) Knowing now that a WLAN hotspot cluster prototype reference for the MAC address "00:20:A6:51:DC:BF" exists at node *Z*, node *Q* can finally contact this node and obtain the appropriate location service information as explained in section 4.7.2 (i.e. how to establish the Sensor and LP sessions appropriately).

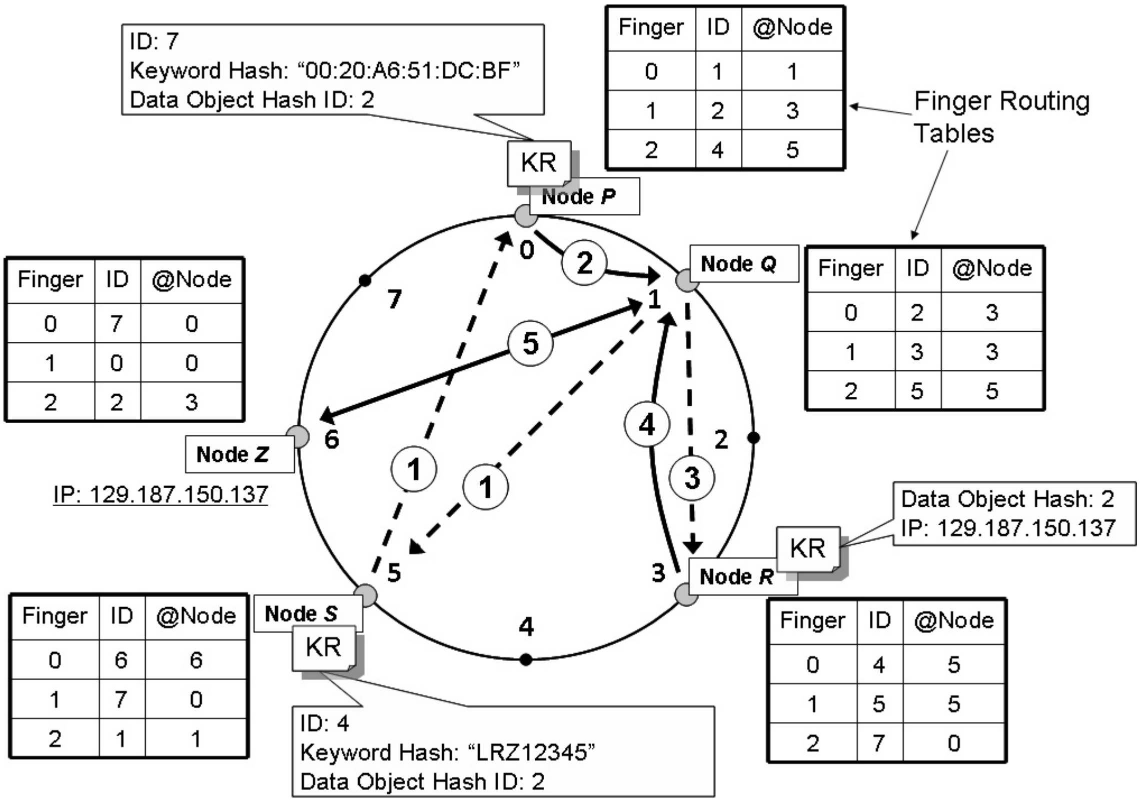


Figure 7.7 Location service lookup using the client sensor data using Chord

As explained earlier, it is very likely that LP service areas naturally overlap, e.g. an outdoor community-based LP picking up an access point outside a building belonging to an indoor LP providing a location service inside that particular building. Assuming that this is the case for access point "00:20:A6:51:DC:BF" multiple KRs of this keyword identifier with the same hash ID would exist on node P and be returned to node Q, but each KR referencing a unique data object KR with a unique ID in the ring. Therefore node Q would have to perform multiple data object KR lookups, respectively, and select the most suitable LP offering a location service as specified in section 4.7.2.

Location Service Discovery using the Hierarchical Chord Ring Structure

If the "entry ring level" cannot be determined from previously cached location information or sensor data, the location service QUERY using the LACBA client sensor data can potentially span the entire hierarchy tree and all nodes potentially resulting in a higher search effort than with a conventional Chord structure. If, however, certain location indicators or ring hierarchy entry points can be assumed, the location service QUERY can be restricted to a particular ring level. Using XPath a precise location service QUERY statement using a qualified path would look as follows:

```
path="/root/germany/munich/"
axis="self-or-decendant"
keyword="MAC Address 00:20:A6:51:DC:BF"
```

This results in a location service query for a reference location data entry/cluster prototype having the WLAN access point MAC Address "00:20:A6:51:DC:BF" in the Chord ring city level "/root/germany/munich/" (e.g. the target LP being community-based) and all of its child rings i.e. including the building level ring, assuming its an access point belonging to an indoor LP. A qualified path from just the sensor data can have a high lookup success probability due to the nature and usual availability of positioning systems in the environment. For instance:

- LPs using GPS coordinates or GSM country code (i.e. CCNMC, LAC) and Cell ID information are most likely found at the country level
- GSM Cell ID and WLAN hotspot MAC Addresses are most likely servable at the city or building level
- RFID sensor tag information etc. most likely at the building level.

The XPath given in the location service query refers to the LP hierarchy path mentioned in chapter 4. Storing this path information as explained would allow for the LP backup handover feature described in section 4.4.2. Further information on the XPath syntax for allowing very flexible QUERY operations can be found here [Eic05][Lee02].

7.3. Summary

In this chapter we have introduced a critical link in the LACBA concept supporting the client-based sensor data capturing and allocation to responsible LP sources, hence also determination and discovery of the most suitable location services on a particular client.

We have discussed several network dependent location service discovery mechanisms relating to the requirements specified by the LACBA concept. Coming from these requirements a structured DHT-based P2P system approach using Chord was introduced using a hierarchical tree structure of several Chord rings representing suitable geographic usage environment domains for LPs and their respective service areas. Furthermore, the processes on LP registration, reference location data indexation and distribution within the Chord ring structure as well as location service discovery using LACBA client sensor data where shown.

A demonstrator illustrating the location service discovery based on the DHT-based P2P system concept using Chord discussed in this chapter was implemented showing the feasibility of this powerful approach. Details on this demonstrator implementation are given in appendix E.

Chapter 8
Prototype Realization

Parts of the LACBA system have been implemented within the scope of this work. The goal was to realize the seamless provisioning of distributed location services, using outdoor community-based and indoor LPs as well as readily available client-based positioning systems (e.g. GPS), showing the feasibility of the LACBA concept. Figure 8.1 gives and overview on the so far carried out work and implementation of LACBA. It shows all major components implemented whereby the control components of the LPs have been omitted for the sake of simplicity.

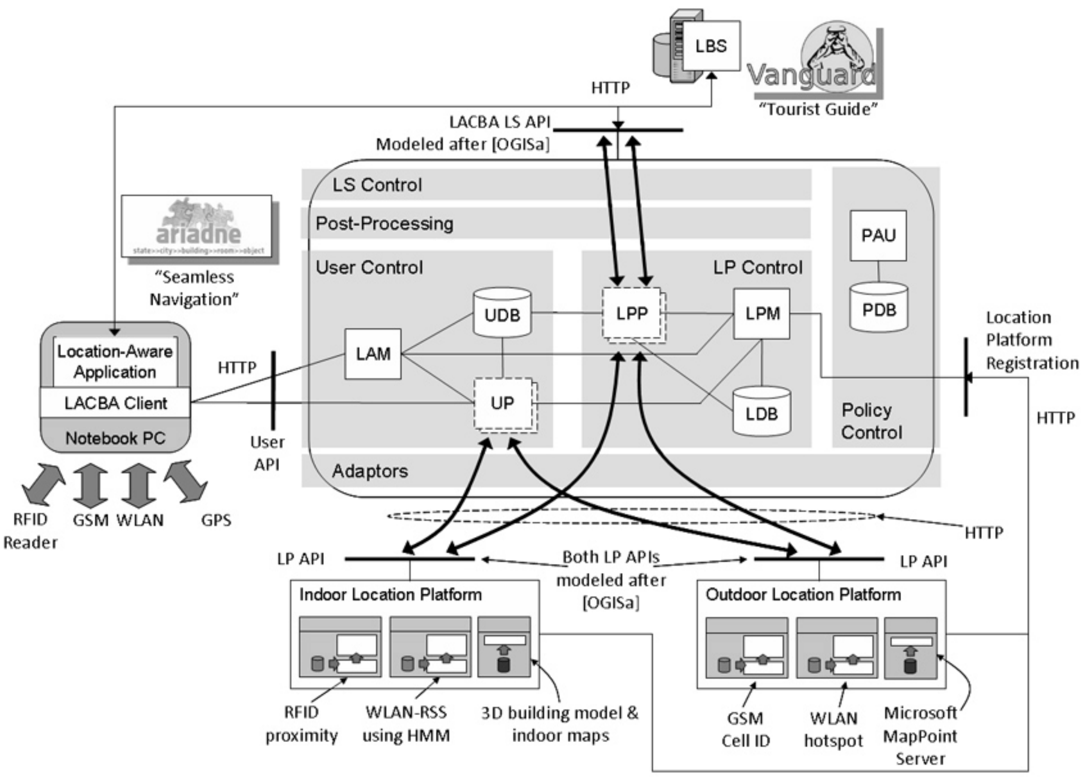


Figure 8.1 Overview of the LACBA prototype realization

Due to the lack of suitable simulation environments regarding LBS, positioning systems, and LP concepts the realization of usable prototypes and localization test-beds in accordant usage environments using real life captured sensor data and performing actual localization on the processed data has been a key requirement on developing a feasible location service architecture concept. Hence LPs and positioning systems where implemented in respective usage environments using the naturally occurring heterogeneous communication network infrastructures. Corresponding web services providing location services and reference location databases have been mainly implemented using Java and readily available open source software systems. The underlying client and server systems providing the necessary interface flexibility and corresponding access to sensor data have been also implemented on open source operating system platforms (i.e. Linux). Commercially available notebook systems have been used as the target LACBA client systems again regarding the communication and sensor interface flexibility and support as well as on usability aspects concerning prototype LBS implementations. Furthermore, notebook client platforms provided the necessary mobility for respective environment specific testing. All data communication is done over HTTP.

As already mentioned in chapter 5 the community and collaboration aspects regarding LACBA in particular usage environments have been evaluated using the StarLogo 2.0 simulation package which provided the necessary flexible simulation environment and software modules to describe and test particular usage environments as well as corresponding conditions very quickly.

The LACBA location service discovery mechanism described in chapter 7 was not implemented in the prototype LACBA system. Due to the only few LPs available for testing a much simpler solution was chosen in the prototype implementation. Instead a Java-based demonstrator implementing the Chord protocol and XPath was adapted from [ZET06] to illustrate the P2P system based LACBA location service discovery mechanism (see appendix E).

8.1. The LACBA Prototype System

The LACBA prototype implementation was divided into several development stages realizing the most critical features first on proving the LACBA concept. The stages were roughly categorized as follows:

- *Stage 1:* Client based sensor data capturing and transmission to appropriate LP entities for position calculation and provisioning of a location service.
- *Stage 2:* In addition to stage 1 initialization and reference location data generation using multiple clients in a particular usage environment (i.e. building of a community-based LP using WLAN hotspot information)
- *Stage 3:* Provisioning of multiple location services to LBS performing handover using various handover triggers.
- *Stage 4:* The network independent location service discovery mechanism.
- *Stage 5:* LQoS matching and negotiation for LQoS-aware LBS using multiple location service sources.

Within the scope of this work only stages 1 to 3 were implemented since they resemble the core features of the LACBA concept. The remaining LACBA release stages are seen subject to future work. Stage 4 was still illustrated using a demonstrator application but not within the LACBA prototype. Stage 5 still required some additional research effort and is clearly seen as a future work item.

As seen in Figure 8.1 the LACBA system prototype comprises of the LACBA client, the LCS and two LPs dedicated for particular usage environments i.e. an indoor and outdoor community-based system, which will be explained in more detail later on. The open source Debian Linux operating system platform was used for all server implementations (i.e. LCS and LP). The various control areas and features handling LACBA clients, their sensor data and location service access were implemented using web services running on an Apache Tomcat application server using the Apache Axis Web Service[16] engine. The web service instances (i.e. proxy components specified in LACBA) are automatically generated by applying the Java2WSDL and WSDL2Java tools which are part of the Axis engine. MySQL was used to implement the LACBA database components and reference location databases of the LPs. A complete description of the implemented methods and program packages presented here can be found in [Laq05] and in the source code JavaDoc documentation.

8.1.1 The Location Core Server

In realizing stages 1 to 3 the User and LP Control areas need to be implemented. For validating user access rights to the LCS and various location services the policy control area as shown in Figure 8.1 is required as well. The generic location service interface definition provided by the OpenLS framework [OGISa] was used for providing the LACBA location service and both LP implementations, hence removing the need for Adaptor components. The location information formats and content of each system were treated individually and were left unmodified by the LCS requiring the LBS to handle the provided information appropriately since stage 5 of the LACBA prototype development was not met. Thus, the LQoS negotiation and adaptation features of the LS control as well as the post-processing functions needed not to be implemented. Furthermore, in the chosen web service implementation approach the web service client stubs (i.e. proxy component definitions) provide the necessary information on accessing the LP services, hence remove the need of the adaptor components.

Table 8.1 gives an overview of the individual source code packages and their functions comprising the LCS prototype implementation, mapping each to the control areas and respective component functions of the LACBA model developed in chapter 4. The database scheme representing the UDB, LDB, and PDB components including their relations and class diagram of the implemented LCS are given in appendix section F1.

LCS source code package	Control area	Component(s)	Description
`LSManger`	User, LP and Policy	LAM, LPM, PAU	Handling of client connections to the LCS, LP discovery and provisioning, as well as access right
`LSManager.template`	LCS global function	Web service management	Holds templates for LCS web services (see LSManager)
`LSManager.gls`	User and LP	UP, LPP	Automatically generated client stub files that are needed to connect to the respective indoor and outdoor LP components
`LSManager.ggis`	User and LP	UP, LPP	
`LSManager.lls`	User and LP	UP, LPP	
`LSManager.lgis`	User and LP	UP, LPP	
`LSmanager.test`	LCS global function	Web service management	Test routine checking web service methods in terms of proper functionality

Table 8.1 LCS prototype component mapping to LACBA component models

[16] Apache Axis Web Service Engine: http://ws.apache.org/axis2/

The LCS provides the basic web service methods as described in Table 8.2 for each LACBA client and LP instance. The actual web service source code is contained in `LSManager` and every instance is created using the `LSManager.template` by running the Java2WSDL tool. The `logOn` and `logOff` methods initiate and destroy the Access session of the user, enabling him to use the services of the LCS. The functions enabling the user to manipulate his personal information and device configurations as described in chapter 4 have not been implemented in the current LACBA prototype, but can be manipulated using provided database management tools such as phpMyAdmin[17]

`LogonResponse`	`logOn(LogonRequest request)` Logs the user on establishing the Access session
`boolean`	`logOff(String check)` Logs the user off and destroys the Access session
`LSResponse`	`getLSList(LSRequest request, String check)` Gets a list of all available LPs using the sensor data of the user
`boolean`	`setLS(LSPair lsPair, String check)` Sets the current LP (i.e. LES and GIS pair) for a user initiating the appropriate LACBA client (UP) and location service (LPP) instances

Table 8.2 The web service methods provided by the LCS

Location Service Discovery

The `getLSList` method performs a simple LP discovery. Every LP is registered manually a priori in the LP pool (i.e. LDB) with appropriate sensor data identification markers (e.g. WLAN MAC address, GSM country code or Cell ID) and IP addresses. The function thus performs a remote database search in the respective reference location databases of the LPs using the current LACBA client sensor data, i.e. the outdoor LP is identified using the GSM country code and WLAN hotspot MAC addresses, and the indoor LP using just WLAN MAC address information. Appropriate matches indicate the LP availability for the user and are returned for selection. This location service discovery implementation has been chosen for simplicity reasons and due to the few LPs used.

Sensor and Location Platform Session Initiation

Once the suitable LP has been selected, the `setLS` method initiates the respective Sensor and LP Session to the selected GIS and LES components as shown in the simplified signaling diagram in Figure 8.2. A LP handover is performed by destroying the previously active GIS and LES associations (i.e. Sensor and LP Sessions) after the new Sensor/LP Sessions have been successfully established in effect performing the LP handover.

The initiation of the Sensor and LP Session to the appropriate LP components creates the corresponding UP and LPP entities handling the communication between the LACBA client and the LP (i.e. Sensor session) as well as between the LCS and LP regarding location service access (i.e. LP Session). In the current LACBA prototype implementation two web service client stub pairs provide the necessary information in each case. The `LSManager.gls` and `LSManager.ggis` stubs provide the information for the "global", outdoor community-based LP, and the `LSManager.lls` as well as the

[17] phpMyAdmin: http://www.phpmyadmin.net/

`LSManager.lgis` for the “local”, indoor LP, whereby the `gls` and `lls` provide information for both Sensor and LP Sessions. The `lgis` and `ggis` stubs provide information regarding the GIS content access. Upon session initiation, the LCS will authenticate and generate appropriate new LC-LES/GIS session IDs using his previously generated LCS-LC session ID, and terminate the old LC-LES/GIS session IDs accordingly. The new session IDs are then returned to the requesting LBS for connecting to the new LES/GIS pairs, enabling the location service and content access accordingly.

Figure 8.2 relates the seamless navigation application implementation which will be introduced in more detail later on in this chapter. It exemplifies the registration process and handoff from the global community-based (GLES) to the indoor LP (LLES) triggered by the user from the application running on his LACBA client.

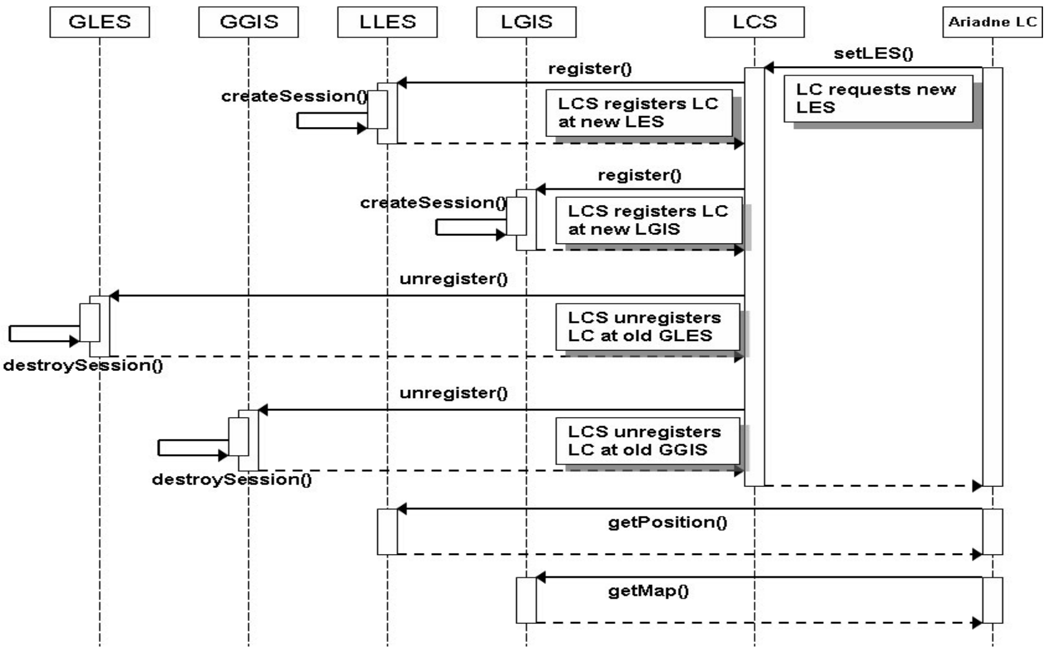


Figure 8.2 The location platform Sensor and LP Session initiation and handover process

Policy Control Enabling User and LP Access Rights

Access rights for LPs can be defined for each user registered with the LCS, i.e. having a corresponding user profile in the UDB. The access settings can be manipulated by means of the previously mentioned database management tools (i.e. phpMyAdmin). User access to the LCS is checked during the logon process as well as access to LP entities when performing the `getLSList` procedure. The policy database profiles of the PDB can be found in appendix section F1 (Figure F.1 and Figure F.2).

8.1.2 The Location Platform Systems

The current LACBA prototype implementation features two LPs using communication network infrastructure of their respective usage environments. Both LP implementations follow the LES model developed in chapter 6 as shown in Figure 6.3. Each LP has its own dedicated GIS system providing the necessary map, routing and/or building model information for its usage environment. Table 8.3 shows the mapping of the implemented

LP components to the model developed in chapter 6. From now on the implemented community-based outdoor LP will be referenced as global LES and GIS (i.e. GLES and GGIS), and the indoor WLAN/RFID-based solution as local LES and GIS (i.e. LLES and LGIS) pairs, respectively.

Location platform source code package	**Control area**	**Component(s)**	**Description**
`gls`	LES Location Service	LLM, LLP	Provides the location service for user localization and user registration (Sensor session initiation) at the GLES
`gls.template`	LES global function	Web service management	Holds templates for GLES web services
`gls.test`	LES global function	Web service management	Test routine checking web service methods in terms of proper functionality
`gls.pl`	LACBA Client	LCM, LCP	Handles the sensor data from the LACBA client and performs sanitation
`lls`	LES Location Service	LLM, LLP	Provides the location service for user localization and user registration (Sensor session initiation) at the LLES
`lls.template`	LES global function	Web service management	Holds templates for LLES web services
`lls.test`	LES global function	Web service management	Test routine checking web service methods in terms of proper functionality
`lls.pl`	LACBA Client	LCM, LCP	Handles the sensor data from the LACBA client and performs sanitation
`lls.utilities`	Data Processing	MM	Provides simple programs assisting in the initialization of the LLES
`lgis`	–	–	Provides the LGIS services for LBS
`lgis.template`	LGIS global function	Web service management	Holds templates for LGIS web services
`lgis.test`	LGIS global function	Web service management	Test routine checking web service methods in terms of proper functionality
`lgis.utilities`	–	–	Provides simple programs assisting in the initialization of the LGIS

Table 8.3 Location platform implementations and component mapping to the LACBA model

The LES location service control (i.e. `gls` and `lls` components) has also been implemented in the form of web services as it was the case with the LCS web services explained in the previous section. The LACBA client control (i.e. `gls.pl` and `lls.pl` components) was implemented using CGI Perl scripts, dumping the sensor data received from clients into a MySQL database (i.e. shared buffer, SB) performing simple sanitation functions as described in chapter 6 (e.g. integrity and format checking).

The data processing part of each LP implementation features dedicated, manually configured PM and IM models suitable for the usage environment. All PM and IM modules are implemented as CGI Perl scripts accessing appropriate location reference data in the LRD and sensor data of the SB in a MySQL database. Information on the PM and IM modules used will be given later on. The management and configuration of the modules used (e.g. role of the MM) is done manually via a phpMyAdmin database tool and via the Apache `cgi-bin` directory in the GLES case. The LLES and LGIS implementations (i.e. `lls.utilities` and `gis.utilities`) feature a dedicated toolset assisting in the initialization and module configuration.

The GGIS system used in conjunction with the GLES is a commercial Microsoft MapPoint 2004 Europe[18] desktop application. Here the interfaces provided by the dynamic link libraries (DLLs) where used allowing access to map images and routing services, hence no special database system was required. On the other hand, the LGIS was custom built from scratch and will be briefly explained later on.

For each LP implementations the database schemes representing the LCD, LRD and LLD components including their relations as well as class diagrams of the implemented GLES, LLES, and LGIS servers are given in appendix section F2 and F3.

The Global Location Enabling Server (GLES)

The GLES features GPS, WLAN hotspot, and GSM Cell ID-based localization using the LACBA client sensor data, providing web service methods as described in Table 8.4 during a Sensor and LP Session. The actual web service source code is contained in `gls` and every instance is created using the `gls.template` by running the Java2WSDL tool.

`PositionResponse`	`getPosition(PositionRequest request, String Check)` Gets the users current position using the best possible and available positioning technology on the user
`boolean`	`registerUser(String data, String check)` Registers the user at the GLES via the Sensor session
`boolean`	`unregisterUser(String data, String check)` Unregisters the user at the GLES and destroys the respective sessions

Table 8.4 The location service methods provided as web services by the GLES

When a position request is performed via the LP Session using the `getPosition` method, the best possible location information is returned depending on the sensor data available from the user's LACBA client. The current GLES implementation features hard coded order in the following sequence: 1) GPS, 2) WLAN hotspot, and 3) GSM Cell ID. Localization results in each case are returned in WGS84 coordinate format.The `registerUser` and `unregisterUser` methods are triggered by the LCS initiating or destroying the Sensor and LP Session, respectively i.e. on a LP handover, LBS termination request, or localization error caused if the current sensor data does not contain the required sensor information needed for localization.

More than 2000 uniquely geo-coded WLAN access points have been captured during test runs in the inner-city of Munich. The *Mean* and *Point-of-max RSS* functions from the sub-cell processing model developed in chapter 6 have been implemented for initialization and localization. For the GSM Cell ID based localization geo-coded GSM base-station locations referenced using GPS for the inner-city of Munich were obtained from existing internet databases. Both GSM and WLAN cluster prototype datasets have been stored in the GLES LRD. When GPS was available in the outdoor environment, the GPS data from the LACBA client is also transmitted to the GLES and directly stored as available location information in the LLD user profile to be provided by the LES location service.

The average localization accuracies obtained outdoors in the inner-city environment using the various positioning technologies were as follows: Since we are also using GPS for the WLAN hotspot cluster prototype generation, this includes the inherent initial

[18] Microsoft MapPoint 2004 Europe: http://www.microsoft.com/mappoint/

error due to the GPS reference precision in the WLAN hotspot initialization model processing (see section 6.3.4). Hence for outdoor WLAN hotspot-based positioning 50% of the discovered WLAN hotspots have ≤60,1m (CEP) and ≤231m radius (2dRMS). The current GLES implementation does not include the WLAN sub-cell initialization/positioning modules as developed in chapter 6 for localization. The GSM Cell-ID-based positioning was rarely used and not very thoroughly tested since it was only used for backup. Expected localization accuracies for GSM are roughly 230m to 430m cell radius on average for the inner-city environment (see section 5.3.1).

The Local Location Enabling Server (LLES)

The LLES provides WLAN RSS-based localization using a Hidden Markov Model (HMM) [SKS03] passive RFID tags combination in an indoor environment encompassing the CDTM institute and staircase area from the ground up to the 2nd floor. The web service methods are described in Table 8.5 and are again provided during a Sensor and LP Session. Similarly to the GLES implementation the actual web service source code is contained in `lls` and every instance is created using the `lls.template`.

`PositionResponse`	`getPosition(PositionRequest request, String check)` Gets the users current indoor position
`FingerprintResponse`	`getFingerprints(FingerprintRequest request, String check)` Gets all WLAN fingerprints for a building
`boolean`	`registerUser(String data, String check)` Registers the user at the LLES via the Sensor session
`boolean`	`unregisterUser(String data, String check)` Unregisters the user at the LLES and destroys the respective sessions

Table 8.5 The location service methods provided as web services by the LLES

Again, when a position request is performed via the LP Session using the `getPosition` method, the LLES system returns an indoor location in a simple 3D coordinate format (x, y, building floor) for mapping the location onto a 3D building model and corresponding map data provided by the LGIS. Additionally the LLES returns a 3D location when RFID information of a passive RFID tag is provided with the LACBA client sensor data which has a mapping onto the indoor building model in the LRD of the LLES as well. A brief illustration and explanation of the indoor building model is given in appendix section F3. The `getFingerprints` is a feature provided by the LLES enabling the download of the entire WLAN RSS fingerprint LRD dataset into the LACBA client cache which can have performance benefits on performing indoor localization in some situations, e.g. in case of faster tracking needing shorter location update intervals. It is only feasible in this case since the LLES LRD dataset is reasonable small. Lastly, the `registerUser` and `unregisterUser` methods provide the same functions as in the GLES implementation.

The package `lls.utilities` provides two programs assisting in the initialization of the LLES LRD database. The `WLANFingerprintCollector` tool is used for gathering WLAN fingerprints. Upon defining the coordinates of a point to be fingerprinted the tool starts a scanning process for 30 seconds. The obtained RSS values for each unique access point discovered are averaged and stored along with the coordinates in the LRD database. The second tool is called `WLANFingerprintFiller` which interpolates RSS values and generates respective fingerprint datasets at coordinates not covered in the initialization space, capable of reducing initialization effort considerably [DZM04].

On localization performance the WLAN RSS-based fingerprinting using HMM achieves an average 2~5m positioning accuracy [Laq05][SKS03]. The HMM approach is particularly useful since it is capable of providing very stable localization in static in indoor scenarios, and it is capable of coping with localisation ambiguities where other fingerprinting approaches face problems (e.g. very close RSS fingerprint values in the same service area when only few access points are present). More information on the HMM workings and its LLES implementation can be found here [Laq05].

The Local Geographic Information System (LGIS)

The LGIS system is closely tied to the LLES since it offers dedicated location content and services for a unique indoor usage environment situated at the CDTM using the web service methods illustrated in Table 8.6.

`LocationResponse`	`getLocation(LocationRequest request, String check)` Gets building specific location information
`MapResponse`	`getMap(MapRequest request, String check)` Gets the specific indoor map image
`POIResponse`	`getPOIList(POIRequest request, String check)` Gets a building specific list of points of interest (POIs)
`RouteResponse`	`getRoute(RouteRequest request, String check)` Calculates a desired indoor route
`boolean`	`registerUser(String data, String check)` Registers the user at the LGIS via the Sensor session
`boolean`	`unregisterUser(String data, String check)` Unregisters the user at the LGIS and destroys the respective sessions

Table 8.6 The location service methods provided as web services by the LGIS

On the web service view the `getLocation` method provides the location information relating to the building coordinate model and the `getMap` method provides the corresponding map section given the building coordinate. Upon invoking the `getPOIList` method a list of all the POIs and their building model location reference is returned. The `getRoute` method calculates the shortest indoor route given a start and end building model location reference. A building waypoint graph is computed using the A* algorithm [RN03] using RFID and WLAN RSS fingerprint markers classified as waypoints in the building model. The A* algorithm has been chosen since it can be implemented very easily. Its heuristics converge very quickly in the underlying graph if the tree branching factor does not exceed 4 (which is true in our application scenario).

Finally the `lgis.utilities` tool provides a simple program called `MapUploader`, assisting in the LGIS initialization by helping with the uploading and management of map data and location content references in the LGIS database.

8.1.3 The LACBA Client

The role of the LACBA client part is to gather and cache the information of the various sensor devices, provide data compression/encryption, and handle the data communication to the LCS. Furthermore, the client has a local location service cache for holding GIS information. Caching building models and map content on the client improves usability and location-aware application performance on the mobile device, especially when using slow data communication networks (e.g. GPRS and in some cases even UMTS). This part was deliberately written in Java, kept lightweight and modular for potential future adaptation to other mobile device platforms.

On the software side the various communication and sensor devices are accessed via appropriate device drivers providing the necessary sensor data information. In case of the WLAN interface state-of-the art WLAN war-driving and network analyzing tools providing suitable C++ APIs are used i.e. Kismet[19]. The actual LABCA client part is written in Java and accesses the driver information via JNI, Perl scripts, and respective Java device APIs. In the current implementation the `ariadne.peripherals` code package handles all these currently connected peripheral devices and corresponding writing scanner routines providing the sensor data. Further information on the client configuration and LACBA client software part can be found in appendix section F4, and in the documentation of [Laq05] and [Ipp06].

8.2. A Seamless Navigation Prototype using LACBA

In order to prove the concept of LACBA, a seamless indoor-outdoor navigation service has been developed within the Ariadne project[20] at CDTM/LKN. The motive was to investigate the following concepts as illustrated in Figure 8.3. A deeper analysis on these issues can be found in [ZIL06b]:

- **Seamless LBS session strategies** i.e. advance session construction, pre-authentication/registration or pre-caching of location data if the route from A to D is known. An alternative would be ad-hoc location service discovery and session construction at the transition points between domains (i.e. B and C)
- **Hanover methods** e.g. a hard handover would be performed upon loosing the GPS signal when entering the building at B requiring a location service discovery after a certain timeout. A soft handover would detect the CDTM LP earlier on approaching from the outside and detecting the presence of the WLAN AP at point B, and prepare the likely impeding handover.
- **Handover triggers** e.g. performing a handover automatically could involve preemptive signaling to the CDTM LP on approaching and detecting the AP at point B, or via a RFID-based door sensor at B indicating a very likely building entry. A manual trigger could be performed by the user via the LBS application GUI once the presence of the CDTM LP has been detected and its availability been signaled by the LACBA system.

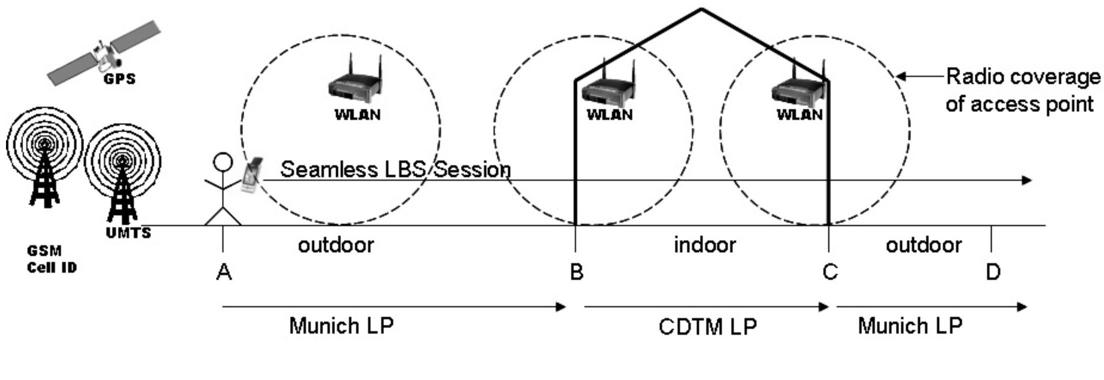


Figure 8.3 Seamless navigation prototype investigation issues

---

[19] Kismet open source software: http://www.kismetwireless.net/
[20] Ariadne Project: http://www.lkn.ei.tum.de/~max

A simple representation of the LACBA configuration used in the Ariadne navigation prototype application is shown in Figure 8.4. The previously explained outdoor and indoor LPs developed within the scope of this work where used providing their location service to a navigation application running on the LACBA client implementation. Outdoors GPS was mainly used using the street maps provided by the Microsoft MapPoint GGIS. WLAN hotspot and GSM Cell ID localization was used as a backup, whereby the GSM country code was used as the marker for location service discovery. The indoor system provided the localization via WLAN RSS-based HMM, using RFID as localization backup (i.e. signs incorporating RFID tags placed at key positions throughout the building as waypoints/checkpoints) as well as target/goal identification in the indoor navigation. The WLAN access point MAC address was used as the location service discovery marker here. Concerning handover triggers, the available LPs were presented to the user via the Ariadne navigation application, performing the handover upon the user's manual selection via the GUI.

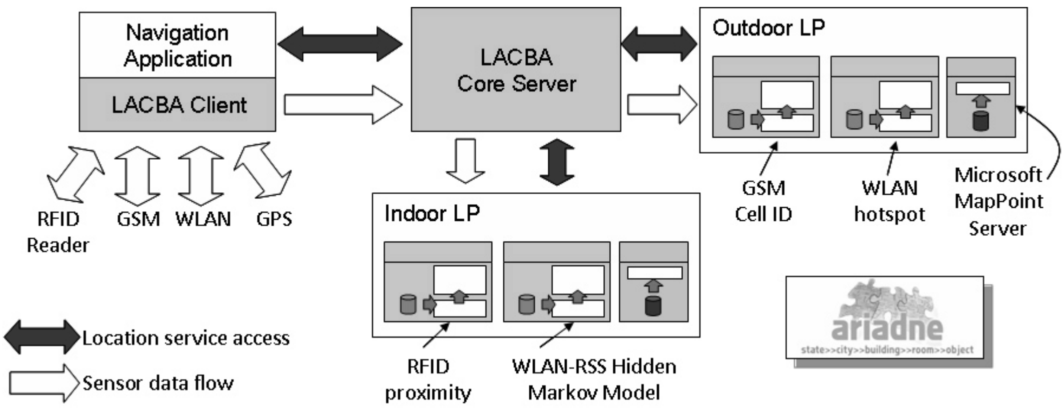


Figure 8.4 The LACBA system configuration of the Ariadne prototype

The building model, map data, and POI list of the LGIS were downloaded into the LACBA client cache when connecting to the indoor system due to performance reasons, but the position calculation was still performed at the LLES. The map and building model data adds up to roughly 1,2 MB in size and the entire download upon the initial indoor system connect took approx. 40 sec. Finally UMTS was used throughout the entire navigation session for data communication. Screenshots of the Ariadne GUI showing the handover from the outdoor to the indoor LP using a user triggered manual handover are shown in appendix F4.

8.3. Summary

The prototypic realization has followed certain goals in demonstrating the feasibility of the LACBA concept. First of all, parts of the LACBA component model developed and specified in chapter 4 have been verified and demonstrated by prototypic realization. Hereby it was essential to built fully working prototypes to prove the novel location service principles set out by this work. Since the complete LACBA framework is quite extensive, the prototype realization has been broken down into certain stages and implemented in the order of importance on demonstrating the novel LACBA approach. Some aspects could not be realized in a workable prototype due to timely restrictions and technical feasibility, potentially requiring additional research effort (e.g. the LQoS negotiation and adaptation as well as location fusion).

In order to demonstrate the location service delivery concept different LP solutions have been implemented and sample applications developed using the LACBA prototype system. On realizing sample LBS applications, a different perspective and additional insights on the usability and provisioning of distributed location services have been gained on improving the LACBA concept helping to define future work items. The seamless navigation application Ariadne was chosen to be implemented illustrating the location service discovery, handover, and handover trigger aspects using multiple distributed location services as required.

Within the scope of the work additional LBS were also implemented. An indoor asset tracking application called Immofind [21] using the indoor LP implementation, demonstrates the effective combination of RFID and WLAN RSS-based localization for inventory management. A LBS tourist guide named Vanguard[22] has been implemented using the outdoor LP, showing the possibilities of WLAN hotspot-based localization and location triggers in inner-city scenarios. Both examples are less suitable in demonstrating the LACBA concept and thus have been omitted in this thesis.

[21] Immofind: http://www.lkn.ei.tum.de/~max
[22] Vanguard: http://mpd.cdtm.de/projects/05WT/Vanguard/

Chapter 9 Resume and Outlook

The growing heterogeneity of positioning systems and future demand for ubiquitous and homogeneous location service delivery provides big challenges to novel distributed location service architectures. Hereby the most cost efficient and accurate positioning technology and GIS solution providing suitable location data for a dedicated usage environment, especially in localized domains such as museums, factories, airports, etc. must be used. Current research approaches and solutions still struggle to give a homogeneous solution on handling the heterogeneity of distributed location service solutions, their network independent discovery, provide a homogenous and generic location service, and, most importantly, react to and capture the changes in the environment and dynamic nature of location-based information. In this thesis a novel distributed location service architecture concept called LACBA is presented, which tries to combine the best-of-breed research solutions and especially address the latter aspect on capturing the dynamic nature of location information in the local usage environments. By combining the advantages of community principles and platform features with distributed location service architecture approaches a paradigm shift in the creation, provisioning, and management of location services was invented.

Developing the novel location service architecture concept required an initial evaluation of existing state-of-the-art approaches and concepts, defining the requirements and features of the new system architecture. The requirements from a service architecture point-of-view required a horizontal separation of the domain specific location services, the processing and harmonization of the distributed location services as well as single point of provisioning of a generic, ubiquitous location service for LBS applications. The requirements from the community and collaboration perspective looked at the vertical processes structure on location service initialization, provisioning, and self-optimization, allocating certain tasks and responsibilities in the horizontal planes of the developed service architecture. Several related research work provides part components and features, but does not integrate all aspects into one service architecture concept. Hereby the focus is either on the service discovery and provisioning (e.g. SoA and Nimbus approaches), location service fusion and harmonization (e.g. Location Stack), or deals with the location information generation from wireless networks and/or community sharing and collaboration aspects individually (e.g. PlaceLab, Location Trader).

Provisioning of a Location Service Using Multiple Heterogeneous Distributed Location Platforms

The LACBA concept describes a service architecture providing a centralized location service using multiple heterogeneous LP systems. A clear separation of functionality is made as follows: A centralized location service provisioning entity provides location service discovery, management of multiple location service sources of registered clients as well as intelligent provisioning of a ubiquitous, generic location service, handling, negotiating and adapting LBS application requirements. The location information capturing, generation, provisioning of localization and location content data as well as self-optimization on changes in the usage environment is kept at the respective responsible LP entities, whereby the LACBA clients capture and provide the necessary sensor data for generation, localization and self-optimization in each case. Adaptation and translation between the heterogeneous LPs and the centralized entity is done via respective Adaptor components.

Flexibility on the intelligence distribution regarding the centralized LACBA provisioning entity is given by having the role of this entity being assumed by a dedicated server (i.e. LCS) in the internet or allocating this functionality also on the LACBA client. In the former case this has advantages in LBS scenarios, optimizing the signalling towards the operator-based LBS provisioning and keeping the complexity and computational requirements on the clients low (i.e. suited for mobile devices with limited storage and processing power). The latter case has signalling and location service provisioning advantages for client-based location-aware applications, thus optimizing the signalling towards the client side, but increases mobile device complexity and requirements.

Modelling of the Service Architecture

All LACBA system aspects and properties have been described using different abstract modelling views by using RM-ODP and UML modelling techniques (e.g. data model, component, and business model views as well as UML class and sequence diagrams). This combination of modelling techniques has been chosen due to LACBA's similar resemblance regarding independent service and resource control in heterogeneous network environments, and successful application in [Kel02]. Six different modelling views were chosen to describe LACBA in both static (e.g. internal and external interfaces, data model) as well as dynamic system properties, i.e. communication behaviour and signalling between LACBA components.

Separated Control Areas for User, Location Platform and LACBA Location Service Provisioning

The requirements specified by the LACBA concept regarding the centralized LCS entity have been realized by distributing the tasks into three distinct, inter-working control areas resembling by the User, LP and LS control. The User control is responsible for handling user access to the LCS and manages the user profile. Once the user is registered he can initiate Sensor sessions to various LP entities as needed. Every Sensor session triggers the generation of a LP Session via the LP control. The LP control performs location service discovery using LACBA client sensor data and manages the location service access to the distributed LPs via LP Sessions for a particular user independent of the centralized LBS location service provisioning. Lastly, the LS control manages the LBS access to the LCS allowing LBS to register their LQoS requirements independently of the available LP sources of a to-be positioned user. The LS control provides the necessary negotiation and adaptation of LQoS as well as signalling of appropriate user position information and GIS content to the requesting LBS.

Collaborative Location Service Building, Usage Behaviour and Adaptation

Following the LACBA concept on developing the vertical location service initialization, positioning and self-optimization processes, the required user interactions, contribution, and location services usage behaviour was analyzed using common community modelling principles. The developed LACBA community model was tested using a suitable outdoor WLAN hotspot and GSM Cell ID community scenario, determining the boundary conditions for successful location service deployment and sustainability in each case. Several influencing factors were evaluated on carrying out the simulations using StarLogo 2.0. Each factor was then weighted and categorized on respective impact (i.e. positive, negative or neutral), indicating that the factors relating to the environment dynamics, i.e. infrastructure changes have a strong negative impact, whereas initial community member size and average usage per week have a strong positive impact. Other factors relating to the user behaviour within the community (i.e. probability of recruitment, discouragement threshold etc.) have been configured neutrally, but can tip the community development in both ways depending if the overall participation is rather “optimistic” or “pessimistic”. Having special interests groups within a community greatly influences the overall participation effort required in building the location service.

A Flexible LES Framework

In the LACBA concept, the location information generation, provisioning and self-optimization tasks are allocated with the LP (i.e. LES part) dedicated to a particular usage environment. A flexible and generic LES model has been developed supporting these processes and allowing for the implementation of various dedicated location service solutions (i.e. indoor and outdoor community-based LPs). Since a community-based LES requires all three tasks fulfilled, suitable initialization and localization methods were discussed and analysed. The localization performance was evaluated using different initialization and localization method combinations, using WLAN access point RSS and GPS information from several thousand hotspot locations captured from the inner-city of Munich.

Communication Network Independent Location Service Discovery

The LACBA concept aimed on providing a communication network independent and simple location service discovery mechanism working across heterogeneous wireless network and location service provider domains. Since LACBA uses the mobile client as sensor in the local environment, the discovery of a suitable location service should be performed on just the captured sensor data from the respective usage environment. The missing key component completing the LACBA concept was found in a DHT-based structured P2P system approach using a hierarchical Chord ring structure. The distributed location services have been allocated to certain ring locations representing their usage environment geographic scope of responsibility, i.e. country, city and building level. The vertical location search across and within the ring hierarchies is performed using the XPath technology. A Java-based demonstrator application from

[ZET06] was adapted to the LACBA location service discovery concept, illustrating node registration and ring generation, reference location data indexation and distribution within Chord rings as well as the LACBA location service query itself.

Prototypic Realization

The essential components and respective features of the LACBA model were implemented using open source software systems and Java web service technology demonstrating the feasibility of the LACBA concept proposed by this work. Two fully working LPs providing localization and GIS content for different usage environments were implemented using the wireless infrastructure present (i.e. an indoor LP providing WLAN RSS-based localization using HMM and passive RFID tags as well as a community-based LP using WLAN hotspot, GSM Cell ID and GPS information). A fully workable seamless navigation application prototype using LACBA and both LP systems was implemented as a proof of concept. The parts not implemented in the prototype implementation were the LQoS matching and negotiation, location information fusion, and the P2P-based location service discovery concept for various explained reasons.

Outlook

The work presented in this thesis provides many possible follow-up research issues:

The concept of LQoS-aware applications could be investigated further, i.e. extending the existing LACBA prototype implementation with the LQoS negotiation, matching and adaptation functionality. The signalling and QoS could be adapted from [ZD04] extending the LQoS concept in LACBA even further. Additional research could be carried out towards location information fusion [Hig03] of heterogeneous location sources in order to meet certain LQoS requirements (i.e. at the LCS level). These aspects were discussed within this thesis work but not seen as feasible at this point in time. Additional research into could potentially re-evaluate this issue.

The development of working LBS prototypes helped to evaluate possible seamless LBS session strategies, handover methods (i.e. soft and hard handovers) as well as handover triggers. Within this work, usability insights were gained on testing the navigation prototype considering certain handover triggers. Currently there is little to no research work done regarding the seamless service session issues in the scope of developing distributed location service architectures, although clearly being a next logical research step.

Community platforms and services are on the rise providing innovative collaboration and communication aspects in many domains (e.g. ad-hoc/spontaneous communities of interest, business collaboration services, etc.). Further emerging concepts on sharing location-based information within localized community groups could be evaluated and developed towards the LACBA concept, e.g. the Cooperative Cars project[23] on cars gathering local traffic information and distributing this information to other local cars, ad-hoc communication between cars of a certain vicinity distributing traffic or accident warnings etc.

[23] Cooperative Cars project: http://www.pt-it.pt-dlr.de/de/1677.php

An interdisciplinary research component could be provided by further analyzing location broker models when pursuing the location service provisioning via a centralized server component, i.e. the LCS. The allocation of the centralized entity functionality poses interesting economical questions regarding who controls the client location information and who would most likely assume this role in a heterogeneous location service provider environment.

The P2P concept introduced in the LACBA location service discovery is very powerful. Considering a LACBA configuration where the role of the centralized LCS entity is allocated on the mobile device as well, LACBA can be easily developed towards a completely P2P-based architecture realizing P2P-based LBS. Hence many more P2P research aspects could be introduced and investigated on the LACBA concept.

Finally the generic principle behind LACBA opens up many possible research fields and application scenarios. It can be virtually applied to any other collaborative context information acquisition, processing and provisioning domains. For instance, using mobile devices equipped with special sensor technology can detect local hazardous situations such as fires, chemical gasses or even function as a mobile health monitor.

List of Figures

List of Tables

Bibliography

Note: References which are formatted in bold are publications of the writer of this thesis being either author or co-author respectively. Pre-publications to this thesis have been marked with "*" additionally as requested at §4, Line 5 of the "Promotionsordnung" of the Munich University of Technology.

[3GP03] 3GPP, 3GPP S3-030744 Technical Specification Group Security WG3, Role of Authentication Proxy (AP-NAF) – Discussion and Pseudo-CRs to TSs on GAA/HTTPS and Presence Security, November 2003

[3GP04] 3GPP, 3GPP.23.271 Technical Specification Group Services and Systems Aspects; Functional stage 2 description of Location Services (LCS), June 2004

[Aer06] Aeroscout Visibility System Overview, http://www.aeroscout.com/

[ASS02] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E.Cayirci, A Survey on Sensor Networks, IEEE Communications Magazine, August 2002

[Bar02] A. L. Barabasi, The new science of networks, Perseus Publishing, Cambridge, USA, 2002

[BVV02] Bayerisches Landesvermessungsamt Muenchen, UTM - Abbildung und Koordinaten, Technical Report, 2002. http://www.geodaten.bayern.de/bvv_web/downloads/UTM-AbbildungenundKoordinaten.pdf

[BSH05] A. Binzenhoefer, D. Staehle, R. Henjes, On the Stability of Chord-based P2P Systems, Research Report No. 347, Dep. Of Computer Science, University of Wuerzburg, January 2005

[BB03] M. Brunato, R. Battiti. PILGRIM: A Location Broker and Mobility-aware Recommendation System, in Proceedings of IEEE PerCom, 2003

[BD00] B. Bruegge, A. Dutoit, Object-Oriented Software Engineering – Conquering Complex and Changing Systems, Prentice Hall, ISBN 0134897250, 2000

[Car99] L. Carotenuto et al, Community Space : Toward Flexible Support for Voluntary Knowledge Communities, Workshop on Workspace Models for Collaboration, April 1999

[CHT03] K. Channabasavaiah, K. Holley, E. Tuggle, Migrating to a service-oriented architecture, IBM DeveloperWorks, 2003

[Che04] H. Chen. An Intelligent Broker Architecture for Pervasive Context-Aware Systems. PhD thesis, University of Maryland, December 2004

[CCK05] Y. Cheng, Y. Chawathe, J. Krumm, Accuracy Characterization for Metropolitan-scale Wi-Fi Localization, IRS-TR-05-003, Intel Corporation, January 2005

[Con05] K. Connelly, A Toolkit for Automatically Constructing Outdoor Radio Maps, International Conference on Information Technology: Coding and Computing (ITCC’05), April 2005

[DA99] A. K. Dey, G. D. Abowd; Towards a Better Understanding of Context and Context-Awareness, GVU Technical Report GIT-GVU-99-22, Georgia Institute of Technology, 1999

[DZM04] P. Dornbusch, M. Zuendt, M. Moeller, Feasibility Study on Seamless Navigation, Center For Digital Technology and Management (CDTM), Technical Report, May 2004

[DNB05] L. De Nardis, M. G. Di Benedetto, Joint communication, ranging, and positioning in low data-rate UWB networks Wireless World Congress 2005, Joint 2nd Workshop on Positioning, Navigation and Communication & 1st Ultra Wideband Experts Talk 2005

[DO01] R. Descartes, P. J. Oscamp, J. Transactions on Discourse on Method, Optics, Geometry, and Meteorology. 2001

[DR00] H. Dietl, S. Royer, Management virtueller Netzwerkeffekte in der Informationsökonomie. Zeitschrift Führung und Organisation, 69, 324-331, 2000

[DFH04] P. Dornbusch, F. Fuchs, H. Huber, Simulating Collaborative Mobile Services – An Approach to Evaluate a New Location Enabling Service 4th International Conference on Electronic Business (ICEB2004) December 2004

[DH03] P. Dornbusch, M. Huber, Generierung von Ortsinformationen durch User-Communities, Proceedings of 6. Internationale Tagung Wirtschaftsinformatik 2003

[DM98] L. Downes, C. Mui, Unleashing the Killer App: Digital Strategies for Market Dominance, Harward Business School Press, 1st Edition, 1998

[DZ02] P. Dornbusch, M. Zuendt, Realisierung von Positionsortungen in WLAN, ITG-Fachtagung "Technologie und Anwendungen für die mobile Informationsgesellschaft" 2002

[DHN04] F. Duerr; N. Hönle; D. Nicklas; C. Becker; K. Rothermel,: Nexus--A Platform for Context-Aware Applications,: 1. Fachgespräch Ortsbezogene Anwendungen und Dienste der GI-Fachgruppe KuVS, Juni 2004

[DH05] P. J. Duffett-Smith, P. Hansen, Precise Time Transfer in a Mobile Radio Terminal, Cambridge Positioning Systems Limited, February 2005

[Eic05] M. Eichhorn, Inhaltsbezogene Hierarchien in strukturierten Peer-to-Peer Netzen, Diploma Thesis, carried out at the Institute of Communication Networks (TUM), October 2005

[ETZ05] B. Eissfeller, A. Teuber, P. R. Zucker, Special Report on: Untersuchungen zum GPS-Satellitenempfang in Gebäuden. Universität der Bundeswehr München

[Eka06] Ekahau; WLAN Positioning Solutions Provider, http://www.ekahau.com

[Eri04] Ericsson Mobile Positioning Protocol (MPP), Mobile Location Protocol 3.0.0, March 2004

[Eur03] European Commission. The Galilei Project-Galileo Design Consolidation. Produced by ESYS, plc, Guidford UK, 2003

[FL05] N. Faggion, S. Leroy, Alcatel Location-based Services Solution – A Key Fixed / Mobile Convergence Enabler, Strategy White Paper, September 2005

[FCC06] Federal Communications Commission (FCC), http://www.fcc.gov/

[Fin00] K. Finkenzeller, RFID Handbook: Radio-frequency Identification Fundamentals and Applications, John Wiley and Sons Ltd. 2000

[FDW04] G. Fischer, B. Dietrich, F. Winkler, Bluetooth Indoor Localization System, 1st Workshop on Positioning, Navigation and Communication 2004

[FT02] N. Freimuth, A. Tasch, Virtuelle und mobile Communities – Begriffsklärungen und Implikationen für Geschäftsmodelle, Arbeitsbericht Nr. 35, ISSN 0942-5098, December 2002

[FM01] H. B. Funk, C. A. Miller; Location Modelling for Ubiquitous Computing: Is This Any Better?, Workshop at Ubicomp 2001

[Gar05] Gartner Research, Maturing Open RFID Applications Will Reshape SCM, Research Note 30 January 2005

[IETF] IETF Working Group on Geographic Location/Privacy (geopriv), http://www.ietf.org/html.charters/geopriv-charter.html

[Gir05] L. D. Girod, A Self-Calibrating System of Distributed Acoustic Arrays, PhD Thesis submitted at the University of California, L.A., 2005

[GJK04] Y. Gwon, R. Jain, T. Kawahara, Robust Indoor Location Estimation of Stationary and Mobile Users, IEEE Infocom, March 2004

[Han05] T. Hanna, Symphonia – Architecture & Technology, Technical Report, Siemens AG , December 2005

[Har01] I. Harre, Calculation of Position Errors dRMS, 2dRMS, CEP95 and the Error Ellipse for p=O.95, January 2001

[HFL04] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, L. E. Kavraki, Pratical robust localization over large-scale 802.11 wireless networks, ACM MobiCom'04, September 2004

[Hau03] G. Hauske, Statistische Signaltheorie, Institute of Communications Engineering (TUM), 2003

[Hen00] J. Henkel, Was ist neu am e-commerce? In Marktstrukturen in der Internet-Ökonomie, Graduiertenkolleg, Juli 2000

[Hig03] J. Hightower: "From Position to Place," in: Proceedings of the 2003 Workshop on Location-Aware Computing, Oct. 2003

[HB01] J. Hightower, G. Borriello, Location Systems for Ubiquitous Computing, in IEEE Computer: Special Issue on Location Aware Computing, 2001

[HCL05] J. Hightower, S. Consolvo, A. LaMarca, I. Smith, J. Hughes, Learning and Recognizing the Places We Go, in Proceedings of Ubicomp 2005, September 2005.

[Hof06] Q. Hofstätter, Analysis of a Content Addressable Network in mobile Ad-Hoc Networks, Diploma Thesis, carried out at the Institute of Communication Networks (TUM), April 2006

[HM04] R. Huang, Y. Manoli, Phased Array and Adaptive Antenna Transceivers in Wireless Sensor Networks, pp. 587-592, Euromicro Symposium on Digital System Design (DSD'04), 2004

[Hu04] M. Huber, Kolloborative Wertschöpfung, Deutscher Universitätsverlag , ISBN: 3824481901, September 2004

[Hul04] W. Hulls W, IP-Based Assisted-GPS Solutions – The Technical Benefits, IIR's Sixth Annual LBS Forum; Budapest, June 2004

[ILM99] Y. Inoue, M. Lapierre, C. Mossotto, The TINA Book – A Co-operative Solution for a Competitve World, Prentice Hall Europe, London, 1999

[Ipp06] P. Ippy, Flexible Initialisierung für Indoor-Outdoor Lokalisierungssysteme, Diploma-Thesis, carried out at the Institute of Communication Networks (TUM), May 2006

[Kal60] R. E. Kalman, A New Approach to Linear Filtering and Prediction Theory, Trans. ASME J. Basic Eng. 82, 1960

[KLL97] D. Karger et al, Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web, 29th Annual ACM Symposium on Theory of Computing, May 1997

[Kar06] H. Karl, Analytische Auswertung geographischer Lokalisierungssysteme in heterogenen, community-basierten Netzen, Study Thesis, carried out at the Institute of Communication Networks (TUM), May 2006

[KBS04] D. Krafzig, K. Banke, D. Slama, Enterprise SOA: Service-Oriented Architecture Best Practices, Prentice Hall, ISBN 0131465759 November 2004

[KFS99] A. Kambil, G. B. Friesen, A. Sundaram, Co-Creation: A New Source of Value, Outlook 1999, No. 2, 1999

[Kel02] W. Kellerer, Serverarchitektur zur netzunabhängigen Dienststeuerung in heterogenen Kommunikationsnetzen, doctorial thesis, Institute of Communication Networks of the Munich University of Technology, 2002

[Koc03] J. H. Koch, Unterstützung der Formulierung und Analyse von virtuellen Communities, doctorial thesis, Institute of Computer Science of the Munich Universtiy of Technology, Peter Lang Verlag, 2003

[KTL06] A. Kuepper, G. Treu, C. Linnhoff-Popien, TraX: A Device-Centric Middleware Framework for Location-Based Services, IEEE Communications Magazine, Vol.44 No.9, September 2006

[Kue05] A. Kuepper, Location-based Services - Fundamentals and Operation, John Wiley & Sons, ISBN 0-470-09231-9, October 2005

[LKD03] G. Lachapelle, H. Kuusniemi, D. T. H. Dao, G. MacGougan, M. E. Cannon, HSGPS Signal Analysis and Performance under Various Indoor Conditions, Proceedings of GPS/GNSS 2003 Conference (Session C3), Portland, OR, September 9-12, 2003

[LaM05] A. LaMarca et al; Place Lab: Device Positioning Using Radio Beacons in the Wild; In proceedings of Pervasive 2005

[Laq05] B. Laqua, Development of a Seamless Navigation Systems based on a Scalable Location Service, System Entwicklungsprojekt (SEP), Computer Science Department for Software & Systems Engineering (TUM), October 2005

[Lee02] D. Lee, Query Relaxation for XML Model, PhD thesis, University of California, 2002

[Leo03] A. Leonhardi. Architektur eines verteilten skalierbaren Lokationsdienstes. Institut für Parallele und Verteilte Systeme, Universität Stuttgart, June 2003

[LES04] Location Based Services LES v3.5 Developer's Guide, Siemens September 2004

[Mer01] H. Merz, Praxis-Lexikon e-business, ISBN 3-478-24940-6, Moderne Industrie Verlag, 1st Edition, October 2001

[MK05] M. Mikalsen, A. Kofod-Petersen, An Architecture supporting Implementation of Context-Aware Systems, Workshop on Context Awareness for Proactive Systems (CAPS), 2005

[Moo99] G. A. Moore, Crossing the Chasm: Marketing and Selling High-tech Products to Mainstream Customers, HarperBusiness; Revised Edition 1999

[MUC06] Statistisches Amt der Landeshauptstadt München – Direktorium, Statistisches Taschenbuch 2006, ISSN 0173-0029, 2006

[Nav06] Navizon, Peer-to-Peer Wireless Positioning, http://www.navizon.com

[NS63] A. Newell, H. Simon, GPS[24]: A program that simulates human thought. In Computers and Thought, ed. Feigenbaum and Feldman. McGraw-Hill, 1963

[Ng03] J. Kee-Yin Ng, et al; A Study on the Sensitivity of the Center of Gravity Algorithm for Location Estimation, Department of Computer Science Hong Kong Baptist University, May 13, 2003

[NH94] R. T. Ng, J. Han, Efficient and effective clustering methods for spatial data mining. 20th VLDB Conference, pp. 144–155, 1994

[NME] The National Marine Electronics Association (NMEA), http://www.nmea.org/

[NIS95] FIPS 180-1, Secure Hash Standard, U.S. Department of Commerce/NIST, National Technical Information Service, April 1995

[Nok01] Nokia White Paper on "Mobile Location Services", August 2001

[OT05] A. Odlyzko, B. Tilly, A refutation of Metcalfe's Law and a Better Estimation for the Value of Networks and Network Interconnections, Technical Report, Digital Technology Centre at the University of Minnesota, March 2005

[OGISa] Open GIS Consortium, OpenLS Initiative, http://www.openls.org

[OGISb] Open GIS Consortium, Geography Markup Language (GML) Implementation Specification, http://www.opengis.org, 2003

[OM04a] OMA (2004) OMA-RD-SUPL-V1_0-20040930-C, Secure User Plain Location Requirements Candidate Version 1.0 30-September 2004

[OM04b] OMA-RD-LOC-2004-1118 Location Architecture Overview Requirements, Open Mobile Alliance, Version 1.0, November 2004

[24] GPS in this case refers to "General Problem Solver"

[OM04c] OMA Mobile Location Protocol (MLP) V3.1 - Status: Candidate Enabler - Release date: March 2004

[Par07] The Parlay Group, http://www.parlay.org

[PAM03] S. Panagiotakis, A. Alonistioti, L. Merakos; "An Advanced Location Information Management Scheme for Supporting Flexible Service Provisioning in Reconfigurable Mobile Networks"; IEEE Communcations Magazine; pp88-98; February 2003

[PHU06] S. Peschke, R. Haeb-Umbach, Particle Filtering of Database assisted Positioning Estimates using a novel Similarity Measure for GSM Signal Power Level Measurements, Proceedings of the 3rd Workshop on Positioning, Navigation and Communication 2005 (WPNC06), March 2006

[Pfe05] T. Pfeifer; Redundant Positioning Architecture; Computer Communication, Vol. 28 (2005) 13, pp. 1575-1585, August 2005

[PPZ99] T. Pfeifer, R. Popescu-Zeletin, A Modular Location-Aware Service and Application Platform; 4th IEEE Symp. On Computers and Comms, ISCC, 1999

[Pue06] M. Puerta, Location Server Discovery and Inter-working, Diploma Thesis carried out at the Institute of Communication Networks (TUM), June 2006

[Ree01] D. P. Reed, The Law of the Pack, Harvard Business Review, page 23 February 2001

[RL02] G. Rigoll, M. Lang, Mensch-Maschine Kommunikation 1, Volume 4, Institute for Human-Machine Interaction (TUM), 2002

[Rhe03] H. Rheingold, Smart Mobs: The Next Social Revolution, Perseus Books, ISBN-10: 0738208612, October 2003

[Rot05] J. Roth, A Decentralized Location Service Providing Semantic Locations, Informatik Bericht 323, Habilitationsschrift, Fernuniversität Hagen, January 2005

[Rou02] G. Roussos, Location Sensing Technologies and Applications, Technical Report published on JISC online library, http://www.jisc.ac.uk/, November 2002

[RBP01] J. Rumbaugh, et al, Object-Oriented Modeling and Design, Prentice Hall, ISBN-10: 0136298419, 1991

[Rus04] G. Ruske, Datenanalyse und Informationsreduktion, Institute for Human-Machine Interaction (TUM), 2004

[RN03] S. J. Russell, P. Norvig, Artificial Intelligence – A Modern Approach. Second Edition, Pearson Education Inc. 2003

[Sar02] T. Sarvanko. Positioning standards E911, E112 and UMTS, 2002 www.telecomlab.oulu.fi/home/Radiotekniikka/materiaali/Standardit.pdf

[SHG03] B. Schilit, J. Hong, M. Gruteser; "Wireless Location Privacy Protection", IEEE Computer Magazine, pp135-137, December 2003

[SV98] C. Shapiro, H. R. Varian, Information rules – A strategic guide to the network economy. Mc Graw Hill, Harward, USA, 1st Edition, 1998

[Ske83] D. Skeen, "A Formal Model of Crash Recovery in a Distributed System". IEEE Transactions on Software Engineering 9 (3) pp219-228, 1983

[Sna06] SnapTrack, gpsOne™ Position Location Solutions for UMTS/GSM/GPRS www.cdmatech.com/solutions/pdf/positionlocation.pdf.

[SKS03] T. Stamoulakatos, S. Kyriazakos, E. Sykas, Hidden Markov Modelling and Macroscopic Traffic Filtering supporting Location Based Service. Dept. of Electrical and Computer Engineering, Technical University of Athens, 2003

[SK01] P. Sties, W. Kellerer, A Generic and Implementation Independent Service Description Model. ICDCS-21W 2001, 21st IEEE International Conference on Distributed Computing Systems, April 2001

[SMK01] I. Stoica, R. Morris, D. Karger, M. Kaashoek, H. Balakrishnan, Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications, ACM SIG-COMM, 2001

[SDN] The Java ME Platform, Sun Developer Network, http://java.sun.com/javame/index.jsp

[TK05] G. Treu, A. Kuepper, Efficient Proximity Detection for Location Based Services, Proceedings of the 2nd Workshop on Positioning, Navigation and Communication 2005 (WPNC05), vol. 2005, March 2005

[TW04] U. Trick, F. Weber, SIP, TCP/IP und Telekommunikationsnetze – Anforderungen – Protokolle – Architekturen, Oldenbourg Verlag, ISBN 3-486-27529-1, 2004

[TSH05] M. Tüchler, V. Schwarz, A. Huber, Accuracy of an UWB localization system based on a CMOS chip, Joint 2nd Workshop on Positioning, Navigation and Communication & 1st Ultra Wideband Experts Talk 2005

[Var06] M. Varsavsky, A Dream Come True, "Les Blogs 2.0" conference, December 2005

[Vic75] T. Vicenty, Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of nested Equations, Directorate of the Overseas Surveys, Survey Review XXII, 176, April 1975

[Wol03] A. Wolf, Tipping Markets, PhD thesis, Martin-Luther-Universität, Wirtschaftswissenschaftliche Fakultät, Wittenberg, 2003

[W3C04] World Wide Web Consortium (W3C), OWL Web Ontology Language Overview Recommendation, http://www.w3.org/TR/owl-features/, February 2004

[WGS] World Geodetic System 1984 (WGS84), http://www.wgs84.com/

[Wig06] Wigle.net, Worldwide WLAN Hotspot Community, http://www.wigle.net/

[WNY05] W. H. Wong, J. K. Ng, W. M. Yeung; Wireless LAN Positioning with Mobile Devices in a Library Environment, 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05), June 2005

[X.901] ITU-T Recommendation X.901, Information Technology – Open Distributed Processing – Reference Model: Overview (ISO/IEC 10746-1) 1997

[YAS03] A. M. Youssef, A. Agrawala, A. U. Shankar; WLAN Location Determination via Clustering and Probability Distributions, First IEEE International Conference on Pervasive Computing and Communications (PerCom'03), March 2003

[Zha02] B. Y. Zhao et al, Brocade: Landmark Routing on Peer to Peer Networks, IPTPS, March 2002

[Zha05] T. Zhang et al, AC-CDS: Autonomous Collaborative Collection, Discovery, and Sharing of Information for Mobile Users and Devices; 3rd International ITRE'05 IEEE Conference, June 2005

[Zha02] Y. Zhao. Standardization of Mobile Phone Positioning for 3G Systems, IEEE Communications Magazine, July 2002

[ZZ03] L. Zhuang, F. Zhou, Understanding Chord Performance and Topology-aware Overlay Construction for Chord. Project Report, available at http://www.cs.berkeley.edu/~zl/doc/chord_perf.pdf, 2003

[ZET06] S. Zoels, M. Eichhorn, A. Tarlano, W. Kellerer, Content-based Hierarchies in DHT-based Peer-to-Peer-Systems, 2nd Workshop on Next Generation Service Platforms for Future Mobile Systems (SPMS 2006), January 2006

[Zue04]* M, Zuendt et al, Location Awareness in Heterogeneous Wireless Networks for Pervasive Communications Support, in Wireless Future Studies - Lectures on Pervasive Communications, Volume 3, Vodafone Group R&D Germany 2004

[ZD06]* M. Zuendt, P. Dornbusch, A Novel Approach to Ubiquitous Location-Based Service Architectures Using Mobile Communities, in Managing Development and Application of Digital Technologies, Springer Verlag 2006

[ZD05a] M. Zuendt, G. Deo, M. Naumann, M. Ludwig, Realizing Peer-to-Peer Location-Based Services in Mobile Networks, Joint 2nd IEEE Workshop on Positioning, Navigation and Communication 2005 (WPNC'05) & 1st Ultra-Wideband Expert Talk (UET'05), March 2005

[ZD05b] M. Zuendt, G. Deo, M. Naumann, M. Ludwig; De-centralized Location Management: Minimizing Privacy Concerns for Location Based Service; 3rd International ITRE'05 IEEE Conference, June 2005

[ZDS04] M. Zuendt, P. Dornbusch, T. Schaefer, P. Jacobi, D. Flade; "Integration of Indoor Positioning into a Global Location Platform"; 1st Workshop on Positioning, Navigation and Communication 2004, Hanover, March 2004

[ZI06a]* M. Zuendt, P. Ippy, B. Laqua, J. Eberspaecher; LACBA – A Location-Aware Community-Based Architecture for Realizing Seamless Adaptive Location-Based Services, In 12th European Wireless Conference (EW2006), April 2006

[ZI06b]* M. Zuendt, P. Ippy, B. Laqua, Seamless Adaptive Location-Based Services using Heterogeneous Positioning Technologies and Multiple Location Servers, In 6th International Workshop on Applications and Services in Wireless Networks (ASWN'06), May 2006

[ZTB03]* M. Zuendt, P. Tabery, C. Bachmeir: "Seamless Handoff in Community-Based and Location-Aware Heterogeneous Wireless Networks", MoMuC Conference 2003, ISBN 3-98088429-5

Appendix A

A1 Summary on Localization Method Principles

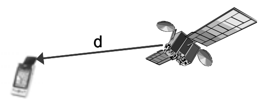
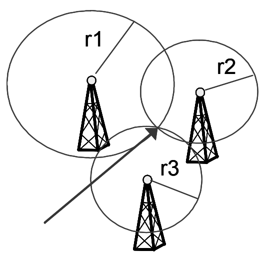
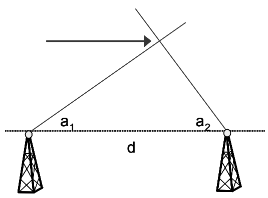
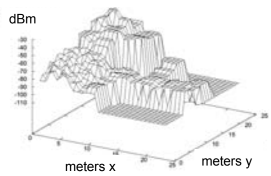
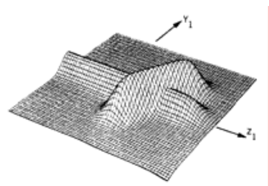
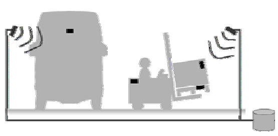
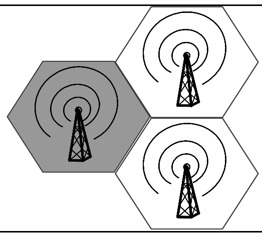
Localization Type	Advantages	Disadvantages	Illustration	Example
Geometric Lateration, timing-based	Very good localization accuracy and precision (LOS) Outdoor environments	Obstructed environments multipath reflections timing resolution and synchronized clocks		In 3G TOA, TDOA, E-OTD, U-TDOA [Zha02] GPS, Galileo [Hul04]
Geometric Lateration, RSS-based	Mediocre localization accuracy and precision	Dynamic environments erratic multipath fading		RSS WLAN-based triangulation [DZ02]
Geometic Angulation	Fairly good localization accuracy and precision in the order: TOA > AOA > RSS	Infrastructure cost and complexity multipath problems accuracy degrades with distance		AOA [HM04]
Scene Analysis Fingerprinting, signal-profiling	Very good localization accuracy and precision Indoor environments	Dynamic infrastructure Initialization cost directly proportional to area multipath fading		RSS WLAN profile table-based matching [DZ02], GSM [Con05]
Statistical Probability-based	Similar to fingerprinting but more “robust” Technology independent, fusion	Computational overhead Dynamic Infrastructure Initialization cost		Normal PDF [GJK04] [WNY05], HMM [SKS03] Particle Filter [PHU06]
Proximity Tag-based	Low cost retrofitting of target entities	Localization range Cost of long-range readers		RFID [Fin00]
Proximity COO-based	Readily available at low cost	Localization accuracy highly depends on cell size		GSM Cell-ID, WLAN-hotspot-based [ZTB03]

Table A 1 Summary of localization methods

A2 Summary on Positioning Technologies

Technology	Typical Application Environment, Availability	Expected Accuracy	Advantages	Disadvantages
GPS	Outdoor, globally and widely distributed	SPS civilian frequency ~4-10m CEP and 25-45m 2dRMS PPS military frequency ~1-5m CEP and 15-25m 2dRMS [ZI06a][Rot05][ETZ05]	Highly accurate, operator independence, user control at end device, low hardware (HW) costs, no initialization i.e. training cost	Satellite visibility, initiation period, increased battery consumption
AGPS	Outdoor, roll-out by selected operators in 2007/08	4-10m [ETZ05]	Highly accurate, operator dependence, low end-device hardware costs, no initialization i.e. training cost	Satellite visibility, access to location information only via mobile operator
Galileo	Outdoor, roll out expected in 2008	1-10m [Eur03]	Same as GPS	Same as GPS, market penetration of end devices
GSM *a) COO* *b) AOA* *c) TOA* *d) TDOA* *e) EOTD*	Outdoor, a) readily available, others rarely implemented	a) 300m to several km b) 150m to 250m c) same as b) d) 100m to 150m [Zha02][Sar02][Kue05]	a) readily available at no additional infrastructure and end device hardware or software (SW) cost	b)→ e) availability questionable due to high HW/SW costs for operators and end devices
UMTS *a) COO* *b) UTDOA*	Outdoor, a) readily available	a) 200m to several km b) Several meters [Zha02][Kue05]	COO and UTDOA potentially possible now	Accuracy still insufficient, breathing cells & cell hierarchy problems
WLAN IEEE802.11 a/b/g/n *(Timing-based)*	Indoor, implement-tations at an experimental stage	<1m [Aer06]	High accuracy, No training costs needed	Lack of conformity to existing WLAN standards, 802.11n might be suitable, high HW costs
WLAN IEEE802.11 a/b/g/n *(RSS-based)*	Outdoor and indoor, readily available	1-10m (indoor) [DZ02][Eka06][ZI06a]	Sufficient accuracy indoors, at least room accuracy, low SW cost	Dynamic environments, training and re-training costs
Bluetooth IEEE 802.15.1 *a) COO* *b) Timinig-based*	Indoor a) readily available b) experim. stage	a) COO-based, 10m b) Lateration, <1m on short range [FDW04]	a) Readily available, cheap b) high accuracy	Limited Coverage, b) hardware costs v2.0 standard more promising
UWB IEEE 802.15.3	Indoor, at an experimental stage	1cm to 30cm, short range, [TSH05][DNB05]	High accuracy, fairly high range ~40m	Interference problems, regulatory issues, costly HW
Siemens LPR	Indoor, readily available	<1m, short range [Zue04]	Very accurate and stable in dynamic usage environments e.g. factories	Specialized hardware, very costly

RFID	Indoor	Few cm to 3m, depending whether active or passive tag	Readily available, passive tags very cheap	Low range, high cost of active devices

Table A 2 Summary on positioning technologies

A3 Summary on Location Platform Approaches

Location platform Criteria	**Device centric**	**Decentralized middleware**	**Centralized**
Location service availability	Poor Only local on active client In case of GPS can serve global, outdoor location information	Mediocre Location service usually available in local network or domain. Usually available in local usage environment (e.g. building, local city [LaM05]) and if client part is active	Good Location service interface globally accessible by 3rd part application providers. Mobile operator ensures 24/7 availability in emergency situations (by EU regulation)
Heterogeneous positioning system support	Poor Only positioning systems based on self-positioning	Good LP framework allows for implementation of many positioning system types	Mediocre Only outdoor-based and few positioning systems supported. Availability and support varies on mobile operator strategy
Interoperability	N/A	N/A No standards specified yet on proprietary LP solutions	Mediocre Standardized but lack of roaming support between operators
Scalability	N/A Very poor in peer-to-peer example	Poor Needs optimization of location update signaling intervals	Mediocre/Good Mobile operator can handle location requests within its domain
Cost (HW/SW/Tx)	Mediocre Usually all in one paid solution, no follow-up costs during usage Tx costs in peer-to-peer configurations	Poor Customized S/W client on client, pre-processing of location information (e.g. battery power) and Tx cost on communication network	Good All location information processing on infrastructure, no user incurred costs on transmission and/or S/W modifications to device
Privacy and access control	Good 100% user control	Mediocre Proprietary solutions, if at all supported. Client can opt-out by deactivating location information Tx	Good/Mediocre Lacks transparency to the user and/or usability
Standardized interfaces and location formats	Good NMEA interface for GPS	Poor No standards yet, proprietary solutions mostly	Good OMA standard for location information

Table A 3 Summary on location platform approaches

Appendix B

B1 The Session Model UML Use Case Description

The session model of LACBA as illustrated in Figure 4.7 can be shown as a UML Use Case description as follows. The corresponding syntax description can be found here [Kel02].

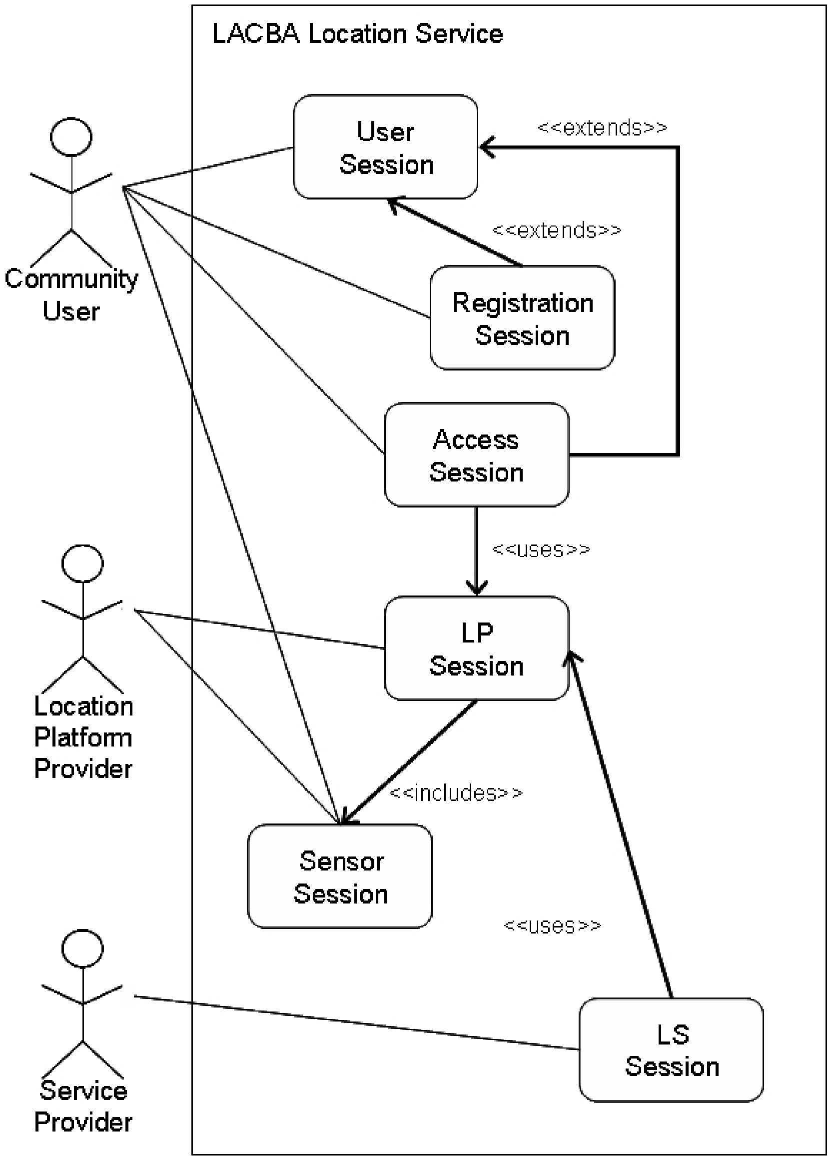


Figure B 1 UML Use Case Description of the LACBA session model

B2 The Location Service Mobility UML Description

The timeframes illustrated in Figure 4.9 describing the location service mobility can each be represented by an individual use case diagram as follows. For the time instance t_1, User 1 has a Sensor and LP Session association with Location Platform Provider 1. Service Provider 1 uses the corresponding LP Session in his LS Session with LACBA (Figure B.2).

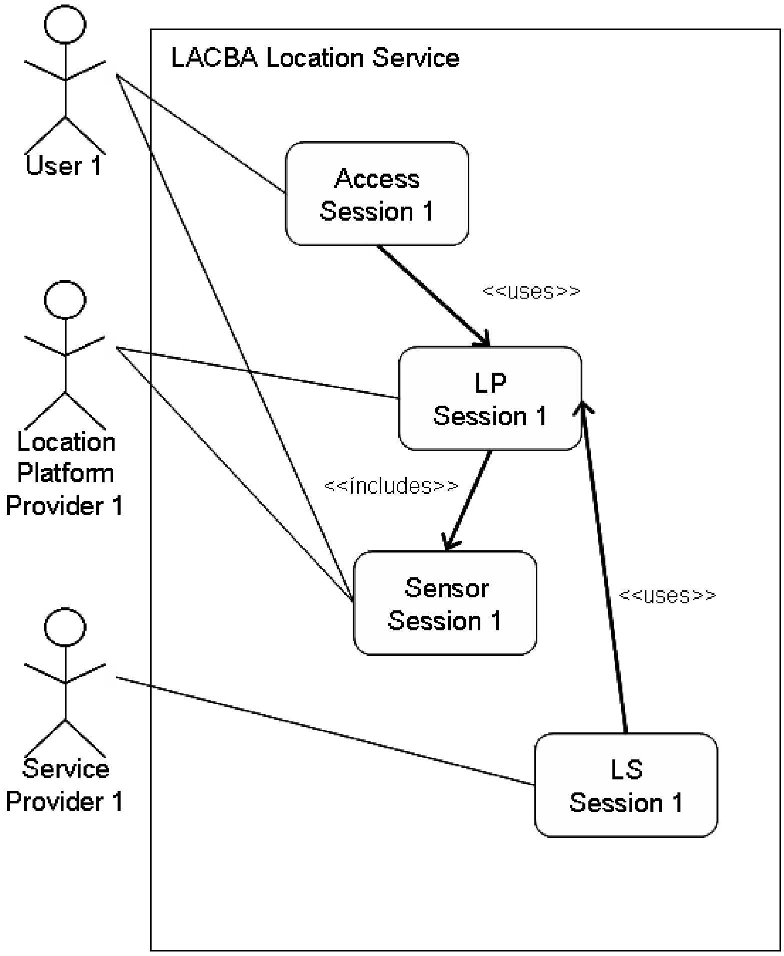


Figure B 2 UML Use Case Description of Location Service Mobility at t_1

At t_{1-2}, User 1 has an additional Sensor and LP Session with Location Platform Provider 2 since User 1 approaches the service area boundary of Location Platform Provider 1 (see Figure B.3). Hence LS Session 1 is signalled the availability of both LP Session 1 and 2 respectively.

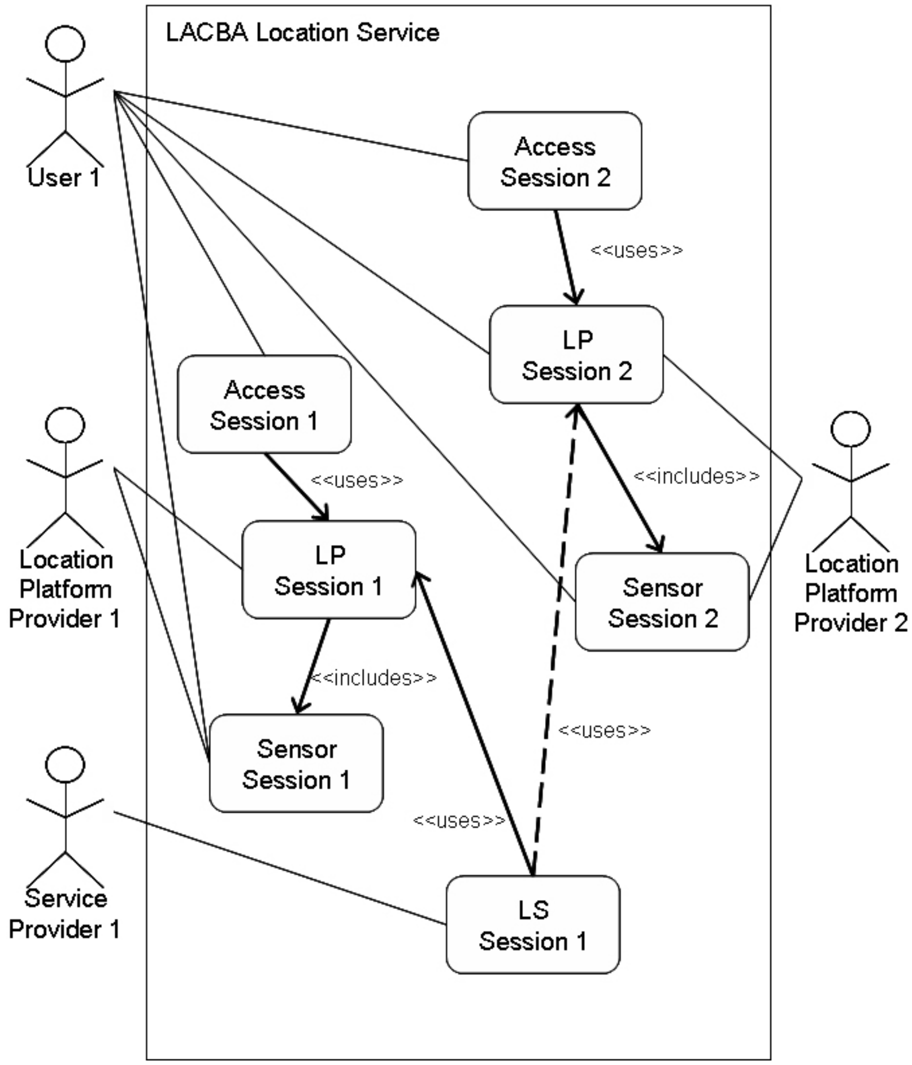


Figure B 3 UML Use Case Description of Location Service Mobility at t_{1-2}

Finally, User 1 leaves the service area of Location Platform Provider 1 so that the LP Session 2 of Location Platform Provider 2 is the only one that remains still active for the user, hence being used by LS Session 1 from now on i.e. the handoff to LP Session 2 carried out (Figure B.4)

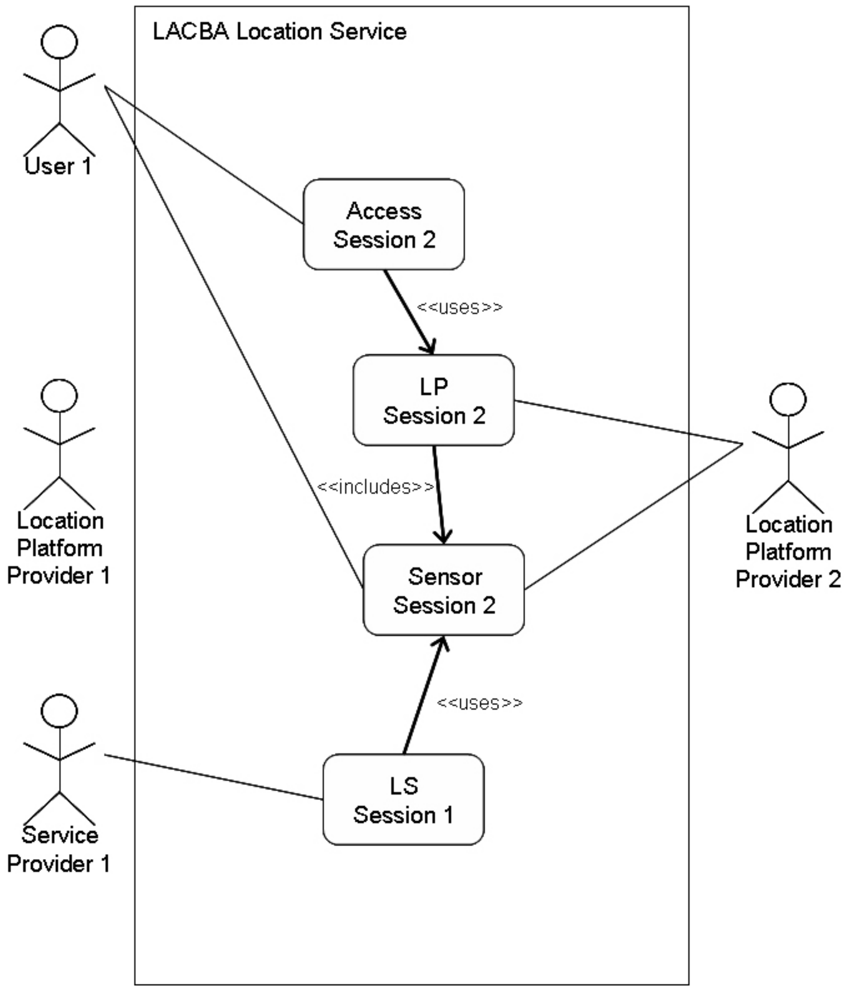


Figure B 4 UML Use Case Description of Location Service Mobility at t_2

Appendix C

C1 Summary of Simulation Modelling Factors

Modeling factor	Short symbol	Unit	Derivation	Factor type
Environment area	A	m²	specified at start	Environment
Average cell radius	R	m	specified at start	Environment/technology
Idealized cell area	M	m²	$M = 4R^2$	Environment/technology
Locations	l	integer	computed during simulation	Environment/technology
Network evolution factor (cell resets per hour)	E	integer	specified at start	Environment/technology
Initial user number	I	integer > 0	specified at start	Environment/community
Maximum addressable target group size	T	integer	specified at start	Environment/community
Current community size	N	integer	computed during simulation	Environment/community
Average usage per week	U	integer	specified at start	User/global
Location service usage probability (per hour)	p_{ls}	percentage	$p_{ls} = \frac{U}{168} x\, 100\%$	User/global
Requests	r	integer	computed during simulation	User/individual
Contributions	c	integer	computed during simulation	User/individual
Satisfactions	s	integer	computed during simulation	User/individual
Frustrations	f	integer	computed during simulation	User/individual
Mobility	m	m/h	specified at start	User/global
Mobility probability per hour	p_m	percentage	$p_m = \frac{m}{R} x\, 100\%$	User/global
Redundant location service request probability	p_{rls}	percentage	specified at start	User/global
Probability of recruitment	p_r	percentage	specified at start	User/global
Recruitment threshold	r_t	integer	$r_t \leq (s - f)$	User/global
Discouragement threshold	d_t	integer	$d_t \leq (f - s)$	User/global

Table C 1 Summary of LACBA simulation modeling factors

C2 StarLogo source code

The code for the individual user behaviour (StarLogo turtle procedures) according to the LACBA model (Figure 5.3) is as follows:

```
// The user mobility
to roam
 rt random 60
 lt random 60
 if mobility > random cellradius [step]
end

// The user contribution procedure including cell indicator colour
// coding for visual representation in the simulator canvas on how
// many locations have been contributed by users for a particular
// cell
```

```
to contribute
  set contributions contributions + 1
  set locations locations + 1
  case locations
  [
    1 [ set value  value + 5]
    2 [ set value  value + 3]
    3 [ set value  value + 2]
    4 [ set value  value + 1]
  ]
end

// This procedure is computed for each user on every simulation step:
// 1) Whether the user uses the location service in his current cell
// 2) If so, is the location service successfully carried out or
//    hasn't a location been mapped for that cell in the data pool
//    yet?
// 3) Is a redundant location service request carried out despite
//    successful location service provisioning?
// 4) The thresholds are computed determining whether the user dies //
      or recruits a new user
to request-service
 if usage + (satisfaction * 0.1) > random 168
 [
      set requests requests + 1
      ifelse locations > 0
      [
            ifelse locationrequestprob > random 100
            [
                  contribute
                  set frustrations frustrations + 1
            ]
            [
                  contribute
                  set satisfaction satisfaction + 1
            ]
      ]
      [
            contribute
            set frustrations frustrations + 1
      ]
      if (frustrations - satisfaction) > frustthreshold [ die ]
      if (satisfaction - frustrations) > satisthreshold
      [
            set recruitingprobmod (recruitingprob *
                              (1 - (count-turtles / maxtargetgroup)))
            if recruitingprobmod > random 100
            [
                  set recruitedusers recruitedusers + 1
                  hatch
                  [
                        set satisfaction 0
                        set frustrations 0
                        set requests 0
                        set contributions 0
                  ]
            ]
      ]
 ]

end
```

The code for the simulation observation and data capturing (i.e. Observer Procedures):

```
// The setup of the global simulation parameters in StarLogo
patches-own [locations]
turtles-own [satisfaction requests frustrations contributions]
globals [clock value recruitedusers maxtargetgroup recruitingprobmod]

to setup
 ca
 crt users
 set clock 0
 set value 0
 set recruitedusers 0
 set maxtargetgroup 760
 ask-turtles [scale-color red satisfaction 0 20 setxcor random 100
setycor random 100]
 ask-patches [setpc black]
end

// The main method carrying out the procedures performed by each user
// according to the LACBA model (Figure 5.3)
to go
 set clock clock + 1
 ask-turtles [ roam request-service ]
 degenerate-data
 reflect-changes
end

to gountil
  loop [
  if (clock / 168) >= until [ stop ]
  go
  ]
end

// The code executing the network evolution on the available cells
to degenerate-data
 if (clock mod change) = 0
 [
      ask-patch-at random screen-width random screen-height [ set
locations -1 ]
 ]
end

// Updating the user frustration, satisfaction and location cell
// mapping visualizations in the currently running simulation
to reflect-changes
 ask-turtles [scale-color red satisfaction 0 20]
 ask-patches [if locations > 0 [scale-pc green locations 1 10]]
 ask-patches [if locations < 0 [scale-pc blue 1 0 2]]
end
```

C3 Derivation of the Maximum Addressable Target Group Size for the Simulation

Looking at the global community user parameters the maximum addressable target group size has been determined the following: Considering a particular usage environment, e.g. all the people living in Munich, roughly 80% of the people should have a mobile phone and 20% of those in turn use mobile data services regularly. Out of those people every tenth person might use a navigation service in some form or another on a regular basis. Using the distribution of user types in the technology adoption lifecycle [Moo99] roughly 16% of these users have an affinity for new innovative (possibly still error prone) technologies and are potential lead users on using them (i.e. visionaries and early adopters). Hence, when considering the potential target group size in Munich for our community-based location service example, this can be estimated to 3328 users (on a total population of roughly 1,3 million), i.e. roughly 0,256%.

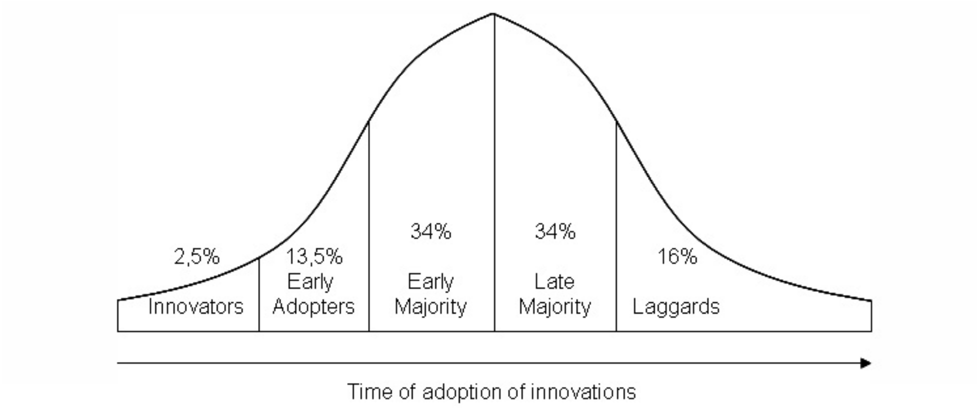


Figure C 1 The technology adoption lifecycle according to Moore

For the simulation, we are looking only at the major inner-city districts of Munich as shown in Table C.1. Here we rely on the current population statistics of the respective districts [MUC06] which sum up to 298750 registered residents. Applying the previously derived 0,256% of the potential target group size for the whole of Munich, results to roughly 760 potential community members for the corresponding districts. An exact estimate is hard to determine since there always exists a possibility that the location service could potentially start a hype thus cause the estimated theoretical maximum to increase rapidly. But this is subject to complex market dynamics and marketing processes beyond the scope of this work. Furthermore, we could have included a percentage of people using the service but not living in the respective usage environment (e.g. people traveling through the districts on their way to work, tourists visiting the city, etc.). However, we still assume that the majority of the people living in the respective environments perform the sensor data collection and contributions to the data pool. Hence this value should be taken as a reasonable estimate.

District of Munich	Number of residents	District Area (km²)
Altstadt – Lehel	18631	3,1639
Ludwigvorstadt – Isarvorstadt	43945	4,3872
Maxvorstadt	42201	4,2917
Schwabing – West	56033	4,367
Au – Haidhausen	52202	4,2178
Sendling	35966	3,9396
Sendling - Westpark	49472	7,8156
Total	**298750**	**32,1828**

Table C 2 Residents count for the target inner-city districts of Munich [MUC06]

C4 Summary of Evolution Rate vs. Initial User Base on Community Development

The table shows a detailed summary of the simulation results visualized in Figure 5.8 of chapter 5.

Initial user count (I)	Occurrence of a cell failure	Average user satisfaction	Average user frustration	% of cells covered after 10 weeks
60	never	15,87	3,6	100
	once a week	13,8	4,88	93,4
	every 2 days	13,26	5,82	90,1
	every day	13,1	6,2	87,3
	twice a day	12,85	13,4	54,1
	every hour	0	0	0
75	never	19,1	2,6	100
	once a week	18,93	2,73	99,85
	every 2 days	18,4	2,8	99,73
	every day	16,78	2,82	99,65
	twice a day	15,53	3,13	99,03
	every hour	0	0	0
100	never	36,7	3,56	100
	once a week	34,58	4,35	100
	every 2 days	33,58	4,37	99,78
	every day	31,5	4,43	99,78
	twice a day	26,46	4,6	99,5
	every hour	0	0	0
125	never	48,35	5,63	100
	once a week	46,83	5,75	99,8
	every 2 days	46,33	5,88	99,8
	every day	46,18	5,98	99,7
	twice a day	45,3	6,03	99,6
	every hour	16,05	7,9	86,68
150	never	57,43	6,75	100
	once a week	56,77	7	100
	every 2 days	56,25	7,1	99,88
	every day	55,78	7,23	99,88
	twice a day	54,13	7,65	99,8
	every hour	34,3	7,98	95,75
200	never	69,1	8	100
	once a week	69,1	8,2	100
	every 2 days	69	8,3	100
	every day	68,8	8,4	100
	twice a day	66,6	8,7	99,7
	every hour	53,3	10,3	96,3

Table C 3 Summary of evolution rate and initial user base effects on the community development

Appendix D

D1 Histograms of Measurement Point Distribution per Access Points

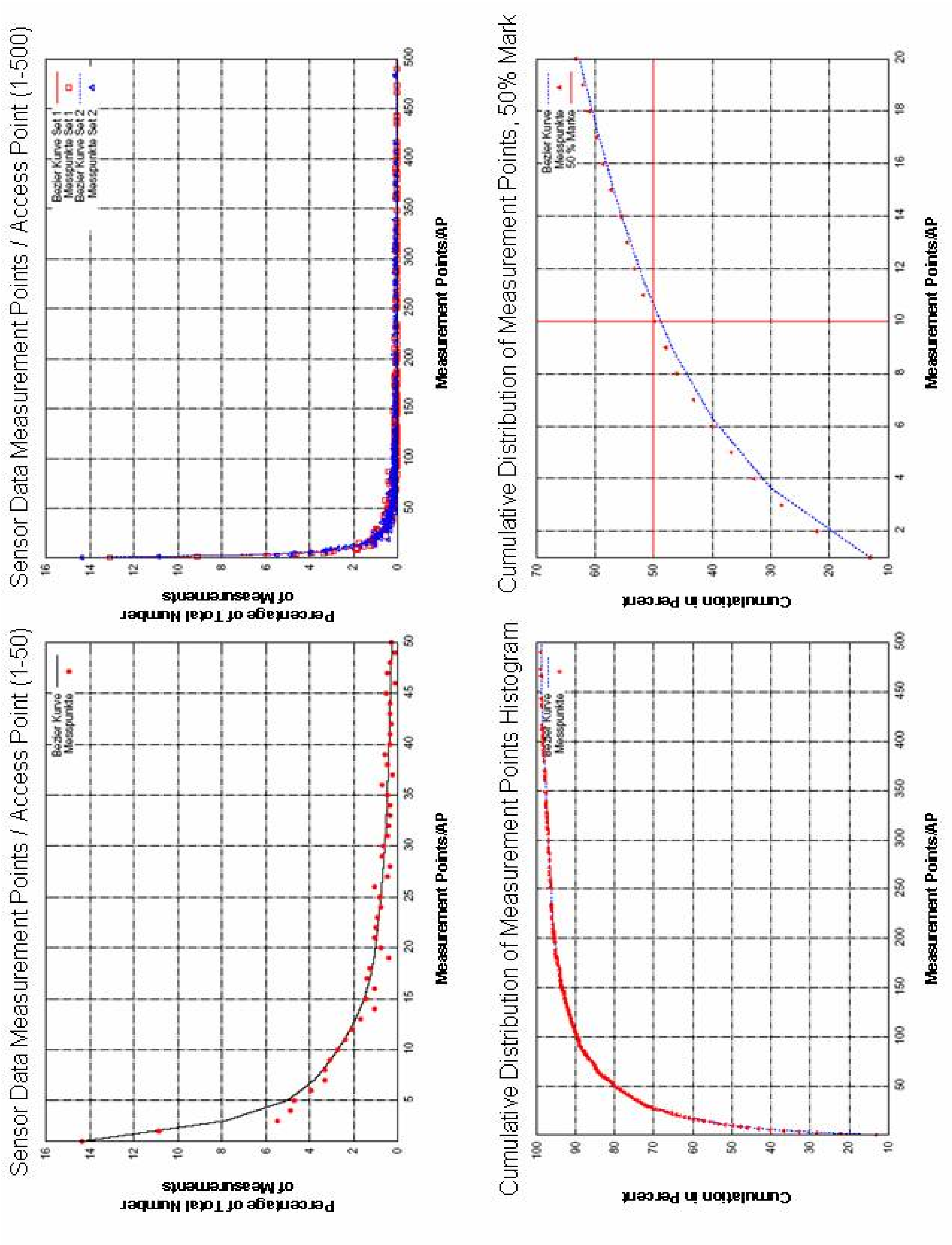


Figure D 1 Sensor data measurement distribution per access point

D2 Source Code of Sub-Cell Model Initialization Methods

Source code for **arithmetic mean** initialization:

```perl
#!/usr/bin/perl
#Initialization using Arithmetical Mean

use strict;
use DBI();

#MySQL Settings
my $mysql_host = "localhost";
my $mysql_db = "database";
my $mysql_login = "user";
my $mysql_passw = "password";
my $mysql_table = "rawdata_table";

#Connect with Database
our $dbh = DBI->connect( "DBI:mysql:database=$mysql_db;host=$mysql_host",
                  $mysql_login, $mysql_passw, { 'RaiseError'=> 1 } );

my $query="INSERT into init_mean ";
$query.="SELECT BSSID, ";
$query.="sum(lat)/count(*) as LAT, ";
$query.="sum(lon)/count(*) as LON ";
$query.="FROM $mysql_table WHERE 1 ";
$query.="group by BSSID";

$dbh->do($query);
$dbh->disconnect();
```

Source code for **point-of-max RSS** initialization:

```perl
#!/usr/bin/perl
#Initialization using Median

use strict;
use DBI();
use Math::NumberCruncher;

#MySQL Settings
my $mysql_host = "localhost";
my $mysql_db = "database";
my $mysql_login = "user";
my $mysql_passw = "password";
my $mysql_table = "rawdata_table";

#Connect with Database
our $dbh = DBI->connect( "DBI:mysql:database=$mysql_db;host=$mysql_host",
                  $mysql_login, $mysql_passw, { 'RaiseError'=> 1 } );

#Define SQL Query
my $query="SELECT BSSID, ";
$query.="GROUP_CONCAT(LAT ORDER BY LAT ASC SEPARATOR ',') as LATLIST, ";
$query.="GROUP_CONCAT(LON ORDER BY LON ASC SEPARATOR ',') as LONLIST ";
$query.="FROM $mysql_table WHERE 1 group by BSSID";

my $sth = $dbh->prepare($query);
$sth->execute();
while ( my $ref = $sth->fetchrow_hashref() ) {
    my @latarray=split(/,/, $ref->{'LATLIST'});
    my @lonarray=split(/,/, $ref->{'LONLIST'});
    #Calculate Medians
    my $latmedian = Math::NumberCruncher::Median(\@latarray);
    my $lonmedian = Math::NumberCruncher::Median(\@lonarray);
```

```
    #Write Initalization Table
    my $iquery="insert into init_median (‘BSSID‘,‘LAT‘,‘LON‘) ";
    $iquery.="VALUES (’$ref->{’BSSID’}’,’$latmedian’,’$lonmedian’)";
    $dbh->do($iquery);
}
$sth->finish();
$dbh->disconnect();
```

Source code for **median** initialization:

```
#!/usr/bin/perl
#Initialization using Point of Maximal Signal Strength

use strict;
use DBI();

#MySQL Settings
my $mysql_host = "localhost";
my $mysql_db = "database";
my $mysql_login = "user";
my $mysql_passw = "password";
my $mysql_table = "rawdata_table";

#Connect with Database
our $dbh = DBI->connect( "DBI:mysql:database=$mysql_db;host=$mysql_host",
                  $mysql_login, $mysql_passw, { ’RaiseError’=> 1 } );

my $query="insert into init_maxsign ";
$query.="SELECT t1.BSSID, LAT, LON, SIGNAL ";
$query.="FROM ";
$query.="(SELECT BSSID, MAX(SIGNAL) as MAXSIG ";
$query.="FROM ";
$query.="$mysql_table WHERE 1 ";
$query.="group by BSSID) as t1, ";
$query.="$mysql_table as t2 ";
$query.="WHERE t1.BSSID=t2.BSSID ";
$query.="AND t1.MAXSIG=t2.SIGNAL ";
$query.="group by BSSID ";

$dbh->do($query);
$dbh->disconnect();
```

D3 Source Code of Implemented Sub-Cell Model Positioning Methods

Source code for **Point of Max. RSS - Median Filter** combination:

```
#!/usr/bin/perl
#Initialization: Point of Maximal Signal Strenght
#Positioning: Median

use strict;
use DBI();
use Geo::Ellipsoid;
use Math::NumberCruncher;

#Using WGS84 Coordinates
my $geo = Geo::Ellipsoid->new(
    ellipsoid => ’WGS84’,
    units => ’degrees’
);

#MySQL Settings
my $mysql_host = "localhost";
```

```
my $mysql_db = "database";
my $mysql_login = "user";
my $mysql_passw = "password";
my $mysql_table = "rawdata_table";

#Define Timewindow min and max
my $min_win="0";
my $max_win="10";

#Connect with Database
our $dbh = DBI->connect( "DBI:mysql:database=$mysql_db;host=$mysql_host",
                 $mysql_login, $mysql_passw, { 'RaiseError'=> 1 } );

#Determine Timestamp range
my $TS_MIN;
my $TS_MAX;
my $query="SELECT MIN(TIMESTAMP) AS TSMIN FROM $mysql_table WHERE 1";
my $sth = $dbh->prepare($query);
$sth->execute();
while ( my $ref = $sth->fetchrow_hashref() ) {
    $TS_MIN=$ref->{'TSMIN'};
}
$sth->finish();
$query="SELECT MAX(TIMESTAMP) AS TSMAX FROM $mysql_table WHERE 1";
$sth = $dbh->prepare($query);
$sth->execute();
while ( my $ref = $sth->fetchrow_hashref() ) {
    $TS_MAX=$ref->{'TSMAX'};
}
$sth->finish();

my $win=$min_win;
#Perform Positioning with window as Parameter
while ($win <= $max_win)
{
    #Set Timestamp to 1st TS
    my $TS=$TS_MIN;
    #LOOP for all Timestamps
    while($TS <=$TS_MAX) {
          #Get GPS Positions at TS
          $query="SELECT LAT,LON FROM $mysql_table WHERE TIMESTAMP=$TS ";
          $query.="LIMIT 0,1;";
          $sth = $dbh->prepare($query);
          $sth->execute();
          my $gps_lat;
          my $gps_lon;
          while ( my $ref = $sth->fetchrow_hashref() ) {
                $gps_lat=$ref->{'LAT'};
                 $gps_lon=$ref->{'LON'};
          }
          $sth->finish();

          #Estimate Positon: Median
          my $query="SELECT COUNT(t1.BSSID) as Used_BSSIDs, ";
          $query.="GROUP_CONCAT(LAT ORDER BY LAT ASC SEPARATOR ',') as
                   LATLIST, ";
          $query.="GROUP_CONCAT(LON ORDER BY LON ASC SEPARATOR ',') as
                   LONLIST ";
          $query.="FROM init_maxsign as t1, ";
          $query.="(SELECT DISTINCT(BSSID) FROM `$mysql_table` WHERE ";
          $query.="TIMESTAMP<=$TS and TIMESTAMP >=$TS-$win) as t2 ";
          $query.="WHERE t1.BSSID=t2.BSSID GROUP BY NULL";
          my $sth = $dbh->prepare($query);
          $sth->execute();

          while ( my $ref = $sth->fetchrow_hashref() ) {
                my @latarray=split(/,/, $ref->{'LATLIST'});
                my @lonarray=split(/,/, $ref->{'LONLIST'});
                my $latmedian = Math::NumberCruncher::Median(\@latarray);
```

```
                my $lonmedian = Math::NumberCruncher::Median(\@lonarray);
                my $laterr=$gps_lat-$latmedian;
                my $lonerr=$gps_lon-$lonmedian;
                my $pyt=sqrt($laterr*$laterr+$lonerr*$lonerr);

                #Calculate Positioning Error in meters
                my @origin = ( $gps_lat, $gps_lon);
                @dest = ( $ref->{’EST_LAT’}, $ref->{’EST_LON’} );
                my $dist = $geo->range( @origin, @dest );
                $dist = sprintf("%.3f", $dist);

                #Write Results in Database
                my $iquery="INSERT INTO pos_maxsign_median ";
                $iquery.="( ";
                $iquery.="‘TIMESTAMP‘,‘window‘,‘GPS_LAT‘, ";
                $iquery.="‘GPS_LON‘,‘Used_BSSIDs‘,‘EST_LAT‘, ";
                $iquery.="‘EST_LON‘,‘LAT_Error‘,‘LON_Error‘, ";
                $iquery.="‘2D_error‘,‘Distance‘ ";
                $iquery.=") ";
                $iquery.="VALUES (";
                $iquery.="’$TS’,’$win’,’$gps_lat’,’$gps_lon’, ";
                $iquery.="’$ref->{’Used_BSSIDs’}’,’$ref->{’EST_LAT’}’, ";
                $iquery.="’$ref->{’EST_LON’}’,’$laterr’,’$lonerr’, ";
                $iquery.="’$pyt’,’$dist’ ";
                $iquery.=");";
                $dbh->do($iquery);
          }
          #Increment Timestamp
          $TS++;
    }
    #Increment Windosize
    $win++;
}
$dbh->disconnect();
```

Source code for **Point of Max. RSS - Mean Filter** combination:

```
#!/usr/bin/perl
#Initialization: Point of Maximal Signal Strenght
#Positioning: Arithmethical Mean

use strict;
use DBI();
use Geo::Ellipsoid;

#Using WGS84 Coordinates
my $geo = Geo::Ellipsoid->new(
    ellipsoid => ’WGS84’,
    units => ’degrees’
);

#MySQL Settings
my $mysql_host = "localhost";
my $mysql_db = "database";
my $mysql_login = "user";
my $mysql_passw = "password";
my $mysql_table = "rawdata_table";

#Define Timewindow min and max
my $min_win="0";
my $max_win="10";

#Connect with Database
our $dbh = DBI->connect( "DBI:mysql:database=$mysql_db;host=$mysql_host",
                 $mysql_login, $mysql_passw, { ’RaiseError’=> 1 } );

```

```
#Determine Timestamp range
my $TS_MIN;
my $TS_MAX;
my $query="SELECT MIN(TIMESTAMP) AS TSMIN FROM $mysql_table WHERE 1";
my $sth = $dbh->prepare($query);
$sth->execute();
while ( my $ref = $sth->fetchrow_hashref() ) {
    $TS_MIN=$ref->{’TSMIN’};
}
$sth->finish();
$query="SELECT MAX(TIMESTAMP) AS TSMAX FROM $mysql_table WHERE 1";
$sth = $dbh->prepare($query);
$sth->execute();
while ( my $ref = $sth->fetchrow_hashref() ) {
    $TS_MAX=$ref->{’TSMAX’};
}
$sth->finish();

my $win=$min_win;
#Perform Positioning with window as Parameter
while ($win <= $max_win)
{
    #Set Timestamp to 1st TS
    my $TS=$TS_MIN;
    #LOOP for all Timestamps
    while($TS <=$TS_MAX) {
        #Get GPS Positions at TS
        $query="SELECT LAT,LON FROM $mysql_table WHERE TIMESTAMP=$TS ";
        $query.="LIMIT 0,1;";
        $sth = $dbh->prepare($query);
        $sth->execute();
        my $gps_lat;
        my $gps_lon;
        while ( my $ref = $sth->fetchrow_hashref() ) {
            $gps_lat=$ref->{’LAT’};
            $gps_lon=$ref->{’LON’};
        }
        $sth->finish();

        #Estimate Positon: Arithmetical Mean
        my $query="SELECT COUNT(t1.BSSID) as Used_BSSIDs, ";
        $query,="AVG(LAT) as EST_LAT, ";
        $query.="AVG(LON) as EST_LON ";
        $query.="FROM init_maxsign as t1, ";
        $query.="(SELECT DISTINCT(BSSID) ";
        $query.="FROM $mysql_table WHERE ";
        $query.="TIMESTAMP<=$TS and ";
        $query.="TIMESTAMP >=$TS-$win) as t2 ";
        $query.="WHERE t1.BSSID=t2.BSSID ";
        $query.="GROUP BY NULL";
        my $sth = $dbh->prepare($query);
        $sth->execute();

        while ( my $ref = $sth->fetchrow_hashref() ) {
            my $laterr=$gps_lat-$ref->{’EST_LAT’};
            my $lonerr=$gps_lon-$ref->{’EST_LON’};
            my $pyt=sqrt($laterr*$laterr+$lonerr*$lonerr);

            #Calculate Positioning Error in meters
            my @origin = ( $gps_lat, $gps_lon);
            @dest = ( $ref->{’EST_LAT’}, $ref->{’EST_LON’} );
            my $dist = $geo->range( @origin, @dest );
            $dist = sprintf("%.3f", $dist);

            #Write Results in Database
            my $iquery="INSERT INTO pos_maxsign_mean ";
            $iquery.="( ";
            $iquery.="‘TIMESTAMP‘,‘window‘,‘GPS_LAT‘, ";
            $iquery.="‘GPS_LON‘,‘Used_BSSIDs‘,‘EST_LAT‘, ";
```

```
                $iquery.="’EST_LON‘,‘LAT_Error‘,‘LON_Error‘, ";
                $iquery.="’2D_error‘,‘Distance‘ ";
                $iquery.=") ";
                $iquery.="VALUES (";
                $iquery.="’$TS’,’$win’,’$gps_lat’,’$gps_lon’, ";
                $iquery.="’$ref->{’Used_BSSIDs’}’,’$ref->{’EST_LAT’}’, ";
                $iquery.="’$ref->{’EST_LON’}’,’$laterr’,’$lonerr’, ";
                $iquery.="’$pyt’,’$dist’ ";
                $iquery.=");";
                $dbh->do($iquery);
        }
        #Increment Timestamp
        $TS++;
    }
    #Increment Windosize
    $win++;
}
$dbh->disconnect();
```

Source code for **Mean - Mean Filter** combination:

```
#!/usr/bin/perl
#Initialization: Arithmethical Mean
#Positioning: Arithmethical Mean

use strict;
use DBI();
use Geo::Ellipsoid;

#Using WGS84 Coordinates
my $geo = Geo::Ellipsoid->new(
    ellipsoid => ’WGS84’,
    units => ’degrees’
);

#MySQL Settings
my $mysql_host = "localhost";
my $mysql_db = "database";
my $mysql_login = "user";
my $mysql_passw = "password";
my $mysql_table = "rawdata_table";

#Define Timewindow min and max
my $min_win="0";
my $max_win="10";

#Connect with Database
our $dbh = DBI->connect( "DBI:mysql:database=$mysql_db;host=$mysql_host",
                  $mysql_login, $mysql_passw, { ’RaiseError’=> 1 } );

#Determine Timestamp range
my $TS_MIN;
my $TS_MAX;
my $query="SELECT MIN(TIMESTAMP) AS TSMIN FROM $mysql_table WHERE 1";
my $sth = $dbh->prepare($query);
$sth->execute();
while ( my $ref = $sth->fetchrow_hashref() ) {
    $TS_MIN=$ref->{’TSMIN’};
}
$sth->finish();
$query="SELECT MAX(TIMESTAMP) AS TSMAX FROM $mysql_table WHERE 1";
$sth = $dbh->prepare($query);
$sth->execute();
while ( my $ref = $sth->fetchrow_hashref() ) {
    $TS_MAX=$ref->{’TSMAX’};
}
```

```
$sth->finish();

my $win=$min_win;
#Perform Positioning with window as Parameter
while ($win <= $max_win)
{
    #Set Timestamp to 1st TS
    my $TS=$TS_MIN;
    #LOOP for all Timestamps
    while($TS <=$TS_MAX) {
            #Get GPS Positions at TS
            $query="SELECT LAT,LON FROM $mysql_table WHERE TIMESTAMP=$TS ";
            $query.="LIMIT 0,1;";
            $sth = $dbh->prepare($query);
            $sth->execute();
            my $gps_lat;
            my $gps_lon;
            while ( my $ref = $sth->fetchrow_hashref() ) {
                    $gps_lat=$ref->{’LAT’};
                    $gps_lon=$ref->{’LON’};
            }
            $sth->finish();

            #Estimate Positon: Arithmetical Mean
            my $query="SELECT COUNT(t1.BSSID) as Used_BSSIDs, ";
            $query,="AVG(LAT) as EST_LAT, ";
            $query.="AVG(LON) as EST_LON ";
            $query.="(SELECT DISTINCT(BSSID) ";
            $query.="FROM $mysql_table WHERE ";
            $query.="TIMESTAMP<=$TS and ";
            $query.="TIMESTAMP >=$TS-$win) as t2 ";
            $query.="WHERE t1.BSSID=t2.BSSID ";
            $query.="GROUP BY NULL";
            my $sth = $dbh->prepare($query);
            $sth->execute();

            while ( my $ref = $sth->fetchrow_hashref() ) {
                    my $laterr=$gps_lat-$ref->{’EST_LAT’};
                    my $lonerr=$gps_lon-$ref->{’EST_LON’};
                    my $pyt=sqrt($laterr*$laterr+$lonerr*$lonerr);

                    #Calculate Positioning Error in meters
                    my @origin = ( $gps_lat, $gps_lon);
                    @dest = ( $ref->{’EST_LAT’}, $ref->{’EST_LON’} );
                    my $dist = $geo->range( @origin, @dest );
                    $dist = sprintf("%.3f", $dist);

                    #Write Results in Database
                    my $iquery="INSERT INTO pos_mean_mean ";
                    $iquery.="( ";
                    $iquery.="‘TIMESTAMP‘,‘window‘,‘GPS_LAT‘, ";
                    $iquery.="‘GPS_LON‘,‘Used_BSSIDs‘,‘EST_LAT‘, ";
                    $iquery.="‘EST_LON‘,‘LAT_Error‘,‘LON_Error‘, ";
                    $iquery.="‘2D_error‘,‘Distance‘ ";
                    $iquery.=") ";
                    $iquery.="VALUES (";
                    $iquery.="’$TS’,’$win’,’$gps_lat’,’$gps_lon’, ";
                    $iquery.="’$ref->{’Used_BSSIDs’}’,’$ref->{’EST_LAT’}’, ";
                    $iquery.="’$ref->{’EST_LON’}’,’$laterr’,’$lonerr’, ";
                    $iquery.="’$pyt’,’$dist’ ";
                    $iquery.=");";
                    $dbh->do($iquery);
            }
            #Increment Timestamp
            $TS++;
    }
    #Increment Windosize
    $win++;
}
```

```
$dbh->disconnect();
```

Source code for **Mean – Median Filter** combination:

```
#!/usr/bin/perl
#Initialization: Arithmethical Mean
#Positioning: Median

use strict;
use DBI();
use Geo::Ellipsoid;
use Math::NumberCruncher;

#Using WGS84 Coordinates
my $geo = Geo::Ellipsoid->new(
    ellipsoid => 'WGS84',
    units => 'degrees'
);

#MySQL Settings
my $mysql_host = "localhost";
my $mysql_db = "database";
my $mysql_login = "user";
my $mysql_passw = "password";
my $mysql_table = "rawdata_table";

#Define Timewindow min and max
my $min_win="0";
my $max_win="10";

#Connect with Database
our $dbh = DBI->connect( "DBI:mysql:database=$mysql_db;host=$mysql_host",
                  $mysql_login, $mysql_passw, { 'RaiseError'=> 1 } );

#Determine Timestamp range
my $TS_MIN;
my $TS_MAX;
my $query="SELECT MIN(TIMESTAMP) AS TSMIN FROM $mysql_table WHERE 1";
my $sth = $dbh->prepare($query);
$sth->execute();
while ( my $ref = $sth->fetchrow_hashref() ) {
    $TS_MIN=$ref->{'TSMIN'};
}
$sth->finish();
$query="SELECT MAX(TIMESTAMP) AS TSMAX FROM $mysql_table WHERE 1";
$sth = $dbh->prepare($query);
$sth->execute();
while ( my $ref = $sth->fetchrow_hashref() ) {
    $TS_MAX=$ref->{'TSMAX'};
}
$sth->finish();

my $win=$min_win;
#Perform Positioning with window as Parameter
while ($win <= $max_win)
{
    #Set Timestamp to 1st TS
    my $TS=$TS_MIN;
    #LOOP for all Timestamps
    while($TS <=$TS_MAX) {
           #Get GPS Positions at TS
           $query="SELECT LAT,LON FROM $mysql_table WHERE TIMESTAMP=$TS ";
           $query.="LIMIT 0,1;";
           $sth = $dbh->prepare($query);
           $sth->execute();
           my $gps_lat;
```

```
        my $gps_lon;
        while ( my $ref = $sth->fetchrow_hashref() ) {
            $gps_lat=$ref->{’LAT’};
            $gps_lon=$ref->{’LON’};
        }
        $sth->finish();


        #Estimate Position: Median
        my $query="SELECT COUNT(t1.BSSID) as Used_BSSIDs, ";
        $query.="GROUP_CONCAT(LAT ORDER BY LAT ASC SEPARATOR ’,’) as
            LATLIST, ";
        $query.="GROUP_CONCAT(LON ORDER BY LON ASC SEPARATOR ’,’) as
            LONLIST ";
        $query.="FROM init_mean as t1, ";
        $query.="(SELECT DISTINCT(BSSID) FROM ‘$mysql_table‘ WHERE ";
        $query.="TIMESTAMP<=$TS and TIMESTAMP >=$TS-$win) as t2 ";
        $query.="WHERE t1.BSSID=t2.BSSID GROUP BY NULL";
        my $sth = $dbh->prepare($query);
        $sth->execute();

        while ( my $ref = $sth->fetchrow_hashref() ) {
            my @latarray=split(/,/, $ref->{’LATLIST’});
            my @lonarray=split(/,/, $ref->{’LONLIST’});
            my $latmedian = Math::NumberCruncher::Median(\@latarray);
            my $lonmedian = Math::NumberCruncher::Median(\@lonarray);
            my $laterr=$gps_lat-$latmedian;
            my $lonerr=$gps_lon-$lonmedian;
            my $pyt=sqrt($laterr*$laterr+$lonerr*$lonerr);

            #Calculate Positioning Error in meters
            my @origin = ( $gps_lat, $gps_lon);
            @dest = ( $ref->{’EST_LAT’}, $ref->{’EST_LON’} );
            my $dist = $geo->range( @origin, @dest );
            $dist = sprintf("%.3f", $dist);

            #Write Results in Database
            my $iquery="INSERT INTO pos_mean_median ";
            $iquery.="( ";
            $iquery.="‘TIMESTAMP‘,‘window‘,‘GPS_LAT‘, ";
            $iquery.="‘GPS_LON‘,‘Used_BSSIDs‘,‘EST_LAT‘, ";
            $iquery.="‘EST_LON‘,‘LAT_Error‘,‘LON_Error‘, ";
            $iquery.="‘2D_error‘,‘Distance‘ ";
            $iquery.=") ";
            $iquery.="VALUES (";
            $iquery.="’$TS’,’$win’,’$gps_lat’,’$gps_lon’, ";
            $iquery.="’$ref->{’Used_BSSIDs’}’,’$ref->{’EST_LAT’}’, ";
            $iquery.="’$ref->{’EST_LON’}’,’$laterr’,’$lonerr’, ";
            $iquery.="’$pyt’,’$dist’ ";
            $iquery.=");";
            $dbh->do($iquery);
        }
        #Increment Timestamp
        $TS++;
    }
    #Increment Windosize
    $win++;
}
$dbh->disconnect();
```

Source code for **Median – Mean Filter** combination:

```
#!/usr/bin/perl
#Initialization: Median
#Positioning: Arithmethical Mean

```

```
use strict;
use DBI();
use Geo::Ellipsoid;

#Using WGS84 Coordinates
my $geo = Geo::Ellipsoid->new(
    ellipsoid => 'WGS84',
    units => 'degrees'
);

#MySQL Settings
my $mysql_host = "localhost";
my $mysql_db = "database";
my $mysql_login = "user";
my $mysql_passw = "password";
my $mysql_table = "rawdata_table";

#Define Timewindow min and max
my $min_win="0";
my $max_win="10";

#Connect with Database
our $dbh = DBI->connect( "DBI:mysql:database=$mysql_db;host=$mysql_host",
                    $mysql_login, $mysql_passw, { 'RaiseError'=> 1 } );

#Determine Timestamp range
my $TS_MIN;
my $TS_MAX;
my $query="SELECT MIN(TIMESTAMP) AS TSMIN FROM $mysql_table WHERE 1";
my $sth = $dbh->prepare($query);
$sth->execute();
while ( my $ref = $sth->fetchrow_hashref() ) {
    $TS_MIN=$ref->{'TSMIN'};
}
$sth->finish();
$query="SELECT MAX(TIMESTAMP) AS TSMAX FROM $mysql_table WHERE 1";
$sth = $dbh->prepare($query);
$sth->execute();
while ( my $ref = $sth->fetchrow_hashref() ) {
    $TS_MAX=$ref->{'TSMAX'};
}
$sth->finish();

my $win=$min_win;
#Perform Positioning with window as Parameter
while ($win <= $max_win)
{
    #Set Timestamp to 1st TS
    my $TS=$TS_MIN;
    #LOOP for all Timestamps
    while($TS <=$TS_MAX) {
            #Get GPS Positions at TS
            $query="SELECT LAT,LON FROM $mysql_table WHERE TIMESTAMP=$TS ";
            $query.="LIMIT 0,1;";
            $sth = $dbh->prepare($query);
            $sth->execute();
            my $gps_lat;
            my $gps_lon;
            while ( my $ref = $sth->fetchrow_hashref() ) {
                    $gps_lat=$ref->{'LAT'};
                    $gps_lon=$ref->{'LON'};
            }
            $sth->finish();

            #Estimate Positon: Arithmetical Mean
            my $query="SELECT COUNT(t1.BSSID) as Used_BSSIDs, ";
            $query,="AVG(LAT) as EST_LAT, ";
            $query.="AVG(LON) as EST_LON ";
            $query.="FROM init_median as t1, ";
```

```
        $query.="(SELECT DISTINCT(BSSID) ";
        $query.="FROM $mysql_table WHERE ";
        $query.="TIMESTAMP<=$TS and ";
        $query.="TIMESTAMP >=$TS-$win) as t2 ";
        $query.="WHERE t1.BSSID=t2.BSSID ";
        $query.="GROUP BY NULL";
        my $sth = $dbh->prepare($query);
        $sth->execute();

        while ( my $ref = $sth->fetchrow_hashref() ) {
            my $laterr=$gps_lat-$ref->{'EST_LAT'};
            my $lonerr=$gps_lon-$ref->{'EST_LON'};
            my $pyt=sqrt($laterr*$laterr+$lonerr*$lonerr);

            #Calculate Positioning Error in meters
            my @origin = ( $gps_lat, $gps_lon);
            @dest = ( $ref->{'EST_LAT'}, $ref->{'EST_LON'} );
            my $dist = $geo->range( @origin, @dest );
            $dist = sprintf("%.3f", $dist);

            #Write Results in Database
            my $iquery="INSERT INTO pos_median_mean ";
            $iquery.="( ";
            $iquery.="`TIMESTAMP`,`window`,`GPS_LAT`, ";
            $iquery.="`GPS_LON`,`Used_BSSIDs`,`EST_LAT`, ";
            $iquery.="`EST_LON`,`LAT_Error`,`LON_Error`, ";
            $iquery.="`2D_error`,`Distance` ";
            $iquery.=") ";
            $iquery.="VALUES (";
            $iquery.="'$TS','$win','$gps_lat','$gps_lon', ";
            $iquery.="'$ref->{'Used_BSSIDs'}','$ref->{'EST_LAT'}', ";
            $iquery.="'$ref->{'EST_LON'}','$laterr','$lonerr', ";
            $iquery.="'$pyt','$dist' ";
            $iquery.=");";
            $dbh->do($iquery);
        }
        #Increment Timestamp
        $TS++;
    }
    #Increment Windosize
    $win++;
}
$dbh->disconnect();
```

Source code for **Median – Median Filter** combination:

```
#!/usr/bin/perl
#Initialization: Median
#Positioning: Median

use strict;
use DBI();
use Geo::Ellipsoid;

#Using WGS84 Coordinates
my $geo = Geo::Ellipsoid->new(
    ellipsoid => 'WGS84',
    units => 'degrees'
);

#MySQL Settings
my $mysql_host = "localhost";
my $mysql_db = "database";
my $mysql_login = "user";
my $mysql_passw = "password";
my $mysql_table = "rawdata_table";
```

```perl

#Define Timewindow min and max
my $min_win="0";
my $max_win="10";

#Connect with Database
our $dbh = DBI->connect( "DBI:mysql:database=$mysql_db;host=$mysql_host",
                  $mysql_login, $mysql_passw, { 'RaiseError'=> 1 } );

#Determine Timestamp range
my $TS_MIN;
my $TS_MAX;
my $query="SELECT MIN(TIMESTAMP) AS TSMIN FROM $mysql_table WHERE 1";
my $sth = $dbh->prepare($query);
$sth->execute();
while ( my $ref = $sth->fetchrow_hashref() ) {
    $TS_MIN=$ref->{'TSMIN'};
}
$sth->finish();
$query="SELECT MAX(TIMESTAMP) AS TSMAX FROM $mysql_table WHERE 1";
$sth = $dbh->prepare($query);
$sth->execute();
while ( my $ref = $sth->fetchrow_hashref() ) {
    $TS_MAX=$ref->{'TSMAX'};
}
$sth->finish();

my $win=$min_win;
#Perform Positioning with window as Parameter
while ($win <= $max_win)
{
    #Set Timestamp to 1st TS
    my $TS=$TS_MIN;
    #LOOP for all Timestamps
    while($TS <=$TS_MAX) {
           #Get GPS Positions at TS
           $query="SELECT LAT,LON FROM $mysql_table WHERE TIMESTAMP=$TS ";
           $query.="LIMIT 0,1;";
           $sth = $dbh->prepare($query);
           $sth->execute();
           my $gps_lat;
           my $gps_lon;
           while ( my $ref = $sth->fetchrow_hashref() ) {
                  $gps_lat=$ref->{'LAT'};
                  $gps_lon=$ref->{'LON'};
           }
           $sth->finish();

           #Estimate Positon: Median
           my $query="SELECT COUNT(t1.BSSID) as Used_BSSIDs, ";
           $query.="GROUP_CONCAT(LAT ORDER BY LAT ASC SEPARATOR ',') as
                    LATLIST, ";
           $query.="GROUP_CONCAT(LON ORDER BY LON ASC SEPARATOR ',') as
                    LONLIST ";
           $query.="FROM init_median as t1, ";
           $query.="(SELECT DISTINCT(BSSID) FROM `$mysql_table` WHERE ";
           $query.="TIMESTAMP<=$TS and TIMESTAMP >=$TS-$win) as t2 ";
           $query.="WHERE t1.BSSID=t2.BSSID GROUP BY NULL";
           my $sth = $dbh->prepare($query);
           $sth->execute();

           while ( my $ref = $sth->fetchrow_hashref() ) {
                  my @latarray=split(/,/, $ref->{'LATLIST'});
                  my @lonarray=split(/,/, $ref->{'LONLIST'});
                  my $latmedian = Math::NumberCruncher::Median(\@latarray);
                  my $lonmedian = Math::NumberCruncher::Median(\@lonarray);
                  my $laterr=$gps_lat-$latmedian;
                  my $lonerr=$gps_lon-$lonmedian;
                  my $pyt=sqrt($laterr*$laterr+$lonerr*$lonerr);
```

```

                    #Calculate Positioning Error in meters
                    my @origin = ( $gps_lat, $gps_lon);
                    @dest = ( $ref->{'EST_LAT'}, $ref->{'EST_LON'} );
                    my $dist = $geo->range( @origin, @dest );
                    $dist = sprintf("%.3f", $dist);

                    #Write Results in Database
                    my $iquery="INSERT INTO pos_median_median ";
                    $iquery.="( ";
                    $iquery.="‘TIMESTAMP‘,‘window‘,‘GPS_LAT‘, ";
                    $iquery.="‘GPS_LON‘,‘Used_BSSIDs‘,‘EST_LAT‘, ";
                    $iquery.="‘EST_LON‘,‘LAT_Error‘,‘LON_Error‘, ";
                    $iquery.="‘2D_error‘,‘Distance‘ ";
                    $iquery.=") ";
                    $iquery.="VALUES (";
                    $iquery.="'$TS','$win','$gps_lat','$gps_lon', ";
                    $iquery.="'$ref->{'Used_BSSIDs'}','$ref->{'EST_LAT'}', ";
                    $iquery.="'$ref->{'EST_LON'}','$laterr','$lonerr', ";
                    $iquery.="'$pyt','$dist' ";
                    $iquery.=");";
                    $dbh->do($iquery);
            }
            #Increment Timestamp
            $TS++;
    }
    #Increment Windosize
    $win++;
}
$dbh->disconnect();
```

D4 Dataset and Sub-Cell Model Graphical Visualization

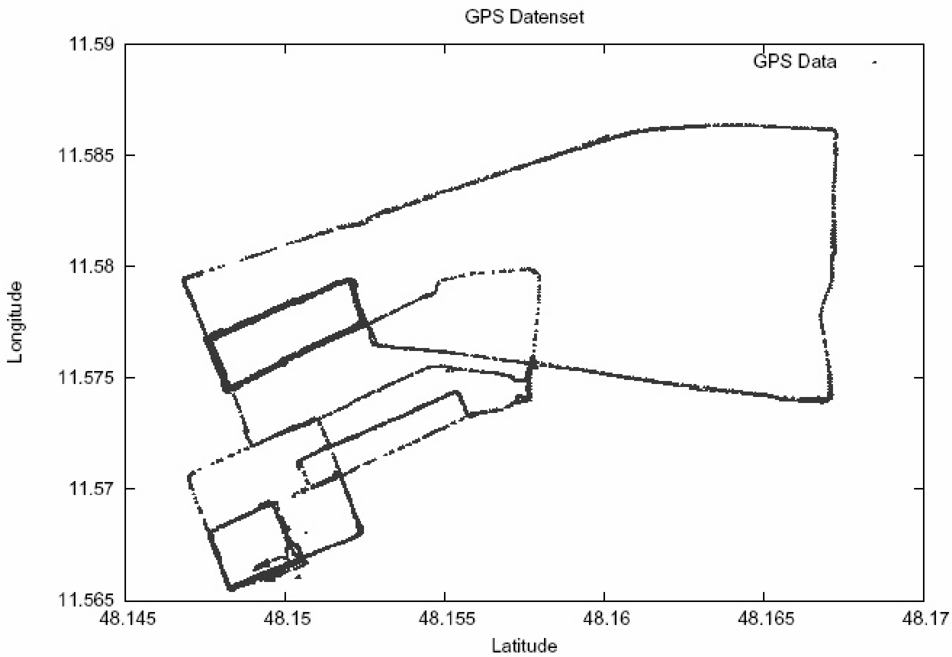


Figure D 2 GPS dataset of the test route within the city of Munich

Figure D 3 Initialization examples showing Mean, Median and Point-of-Max. RSS calculations for a cluster prototype at particular GPS reference point instance

Figure D 4 Positioning using several cluster prototypes at a particular instant using GPS for comparison (Median – Mean filter combination)

Appendix E

E1 The Java Demonstrator for the DHT-based Peer-to-Peer System LES Discovery Mechanism

The DHT- based P2P-based location service discovery mechanism using Chord and XPath search structure was implemented in a Java demonstrator application, simulating the hierarchical Chord ring structure (see Figure E.1). Here, the original three layer ring hierarchy (Figure 7.3) was extended by another room ring layer for RFID-based localization and identification of infrastructure devices and room sings as explained in the previous section.

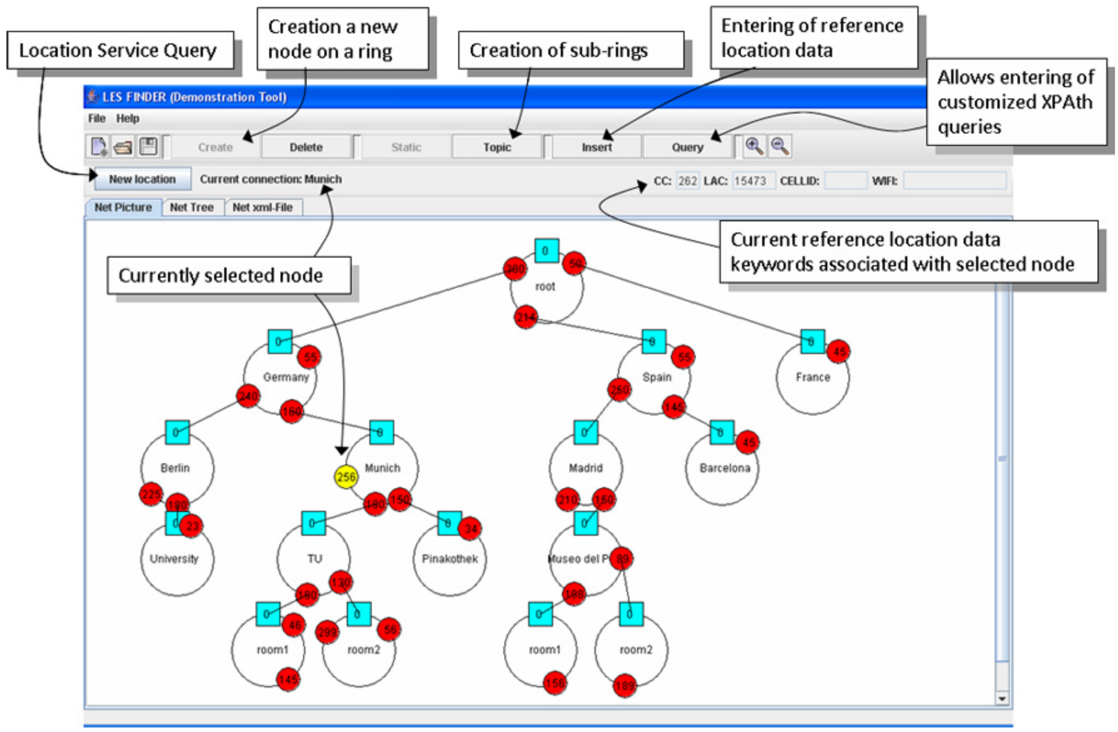


Figure E 1 The DHT-based P2P System Demonstrator application

The demonstrator tool allows for the creation of additional Chord rings, location platform nodes, and reference location data information at arbitrary positions in the hierarchical ring structure via respective GUI dialogue boxes. Individual hash codes are computed automatically upon entering the corresponding keywords. A hash code range restriction has been set between 1 and 360 in the demonstrator due to visualization reasons i.e. drawing nodes onto 360 degree positions on a circle.

Upon selecting an arbitrary node in the demonstrator, a location service query can be performed. The location service query GUI dialogue box is shown in Figure E.2 and allows the entering of reference location data keywords supporting GSM (i.e. country code, LAC and Cell ID), WLAN (i.e. MAC address), RFID (i.e. an arbitrary string value) and GPS information (i.e. longitude and latitude values). Again, the appropriate hash code values are computed and shown next to the data entry fields. On executing the query search, the demonstrator will show each hop of the search as it moves through the ring structure, highlighting each queried node until the target node has been found.

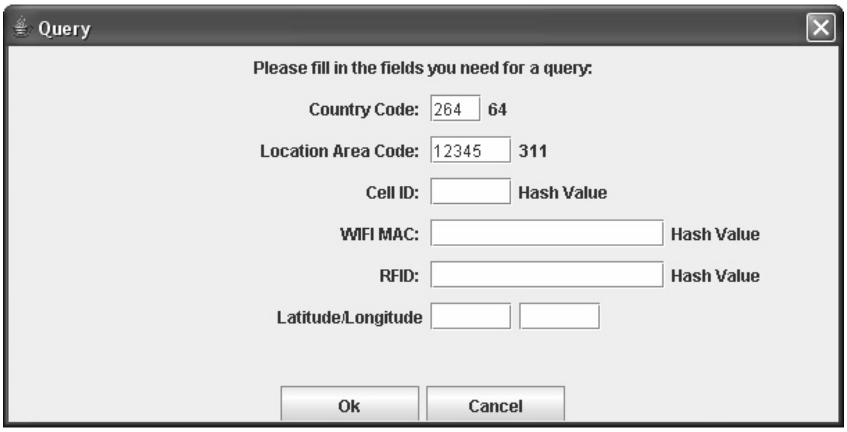


Figure E 2 Dialogue box to enter sensor data keywords for the location service query

When using GPS coordinates for a location service query, a transformation function is used similar to the UTM format [BVV02]. The configuration tool box of the Java demonstrator for this purpose is shown in Figure E.3. A desired bit key length for the GPS reference ID or a resolution radius can be specified instead for determining the appropriate key length. The longer the key length the higher the possible resolution of the GPS coordinate e.g. a bit key length of 28 can map a GPS coordinate pair to a unique 250m x 250m geographic area.

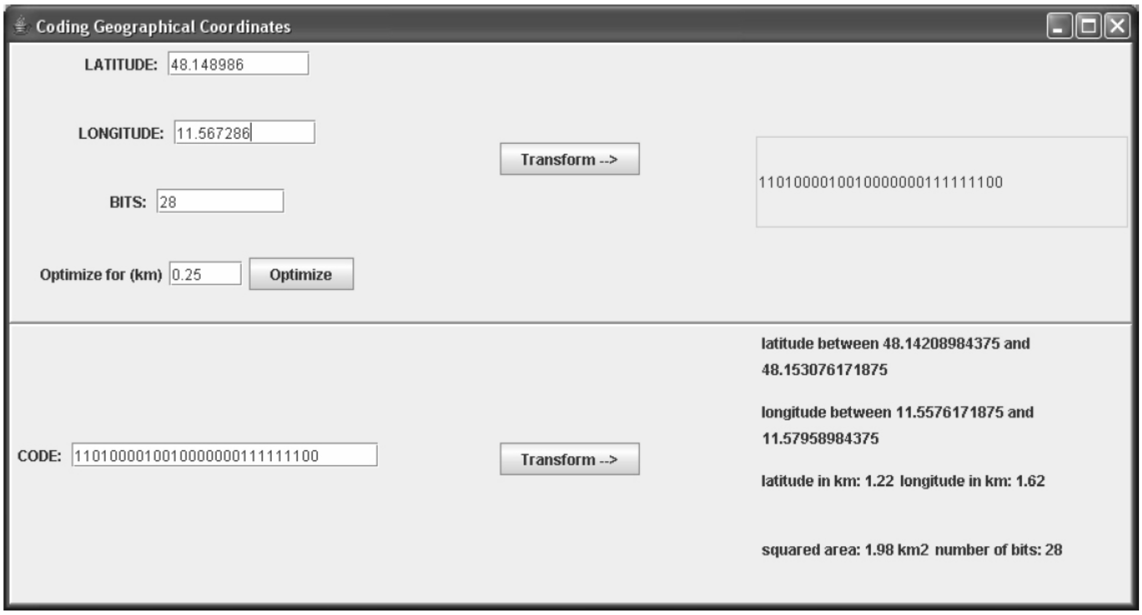


Figure E 3 Dialogue tool box for transforming GPS coordinates to a unique n bit code and in reverse

More information on the Demonstrator tool, example search queries and especially the reference location data ring insertion using our special GPS coding can be found here [Ipp06][Pue06].

Appendix F

F1 The LCS prototype implementation

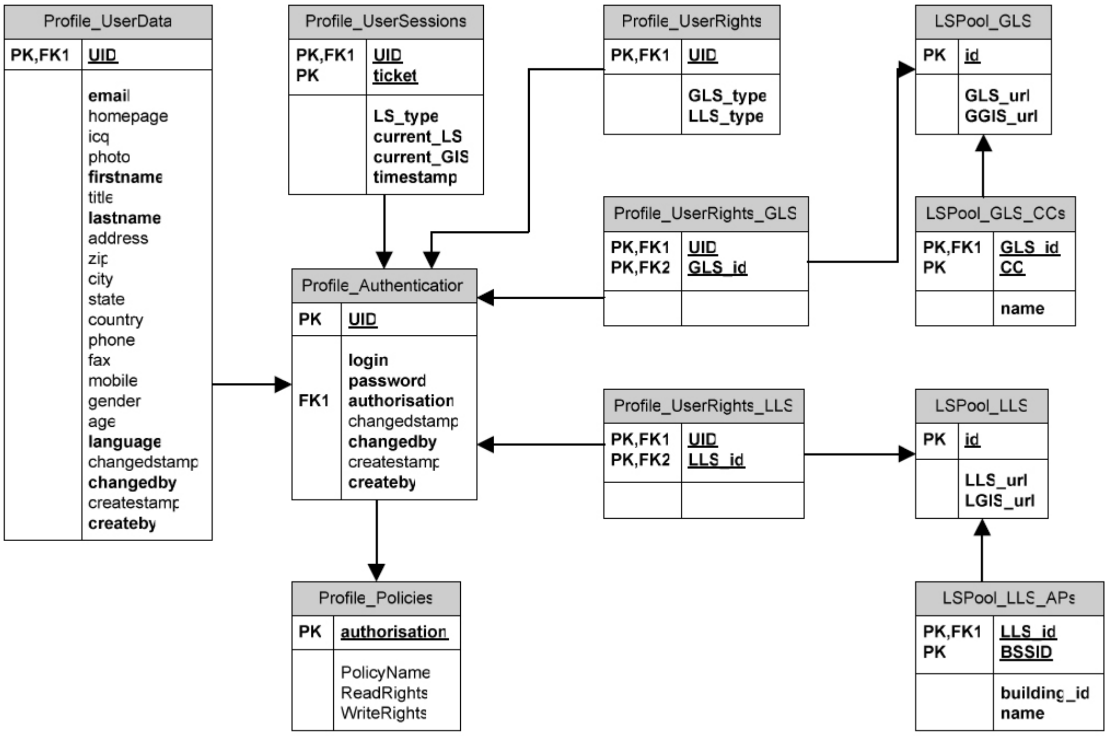


Figure F 1 The database scheme of the LCS prototype

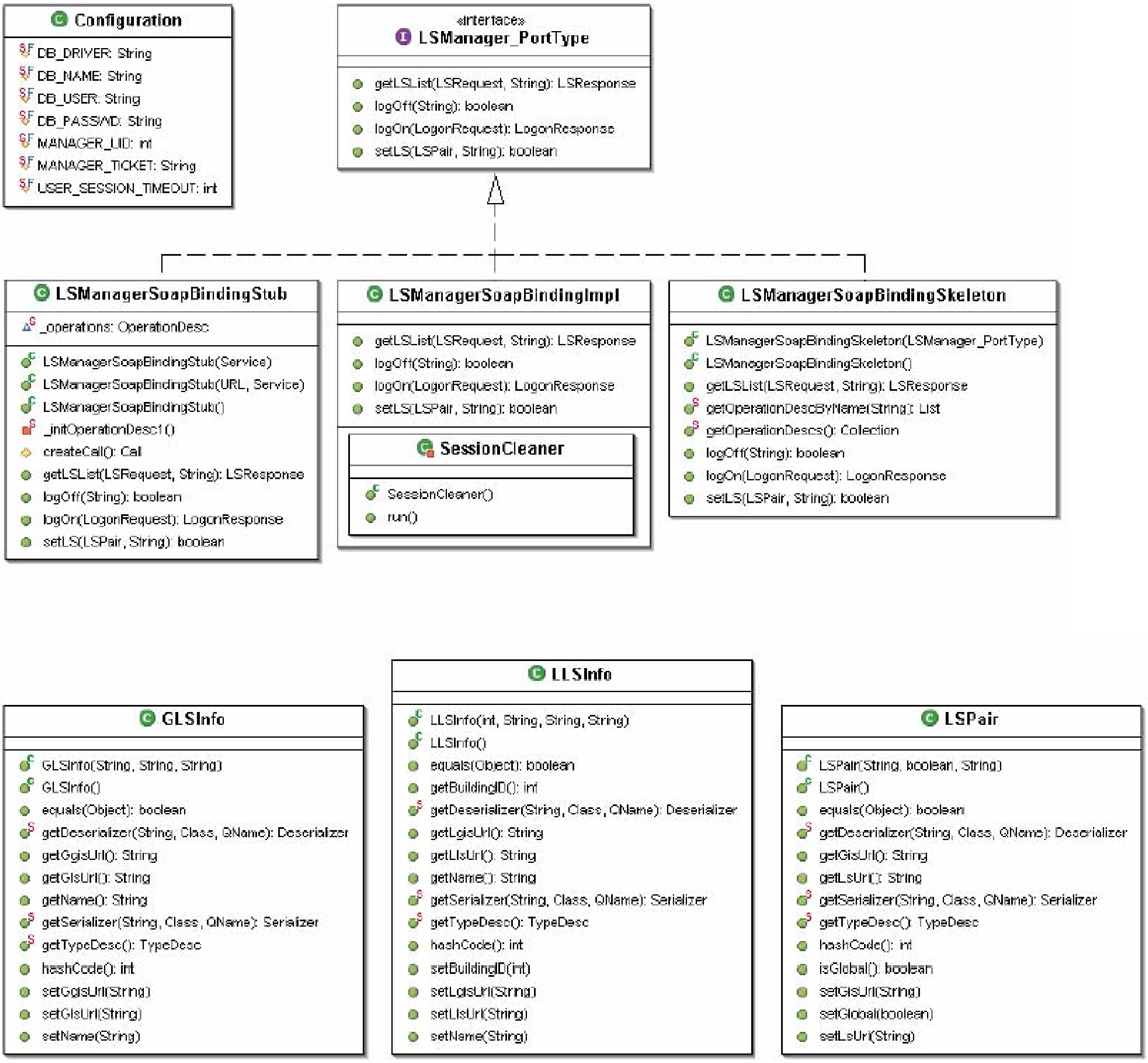


Figure F 2 Class diagram of the LCS prototype (Part 1)

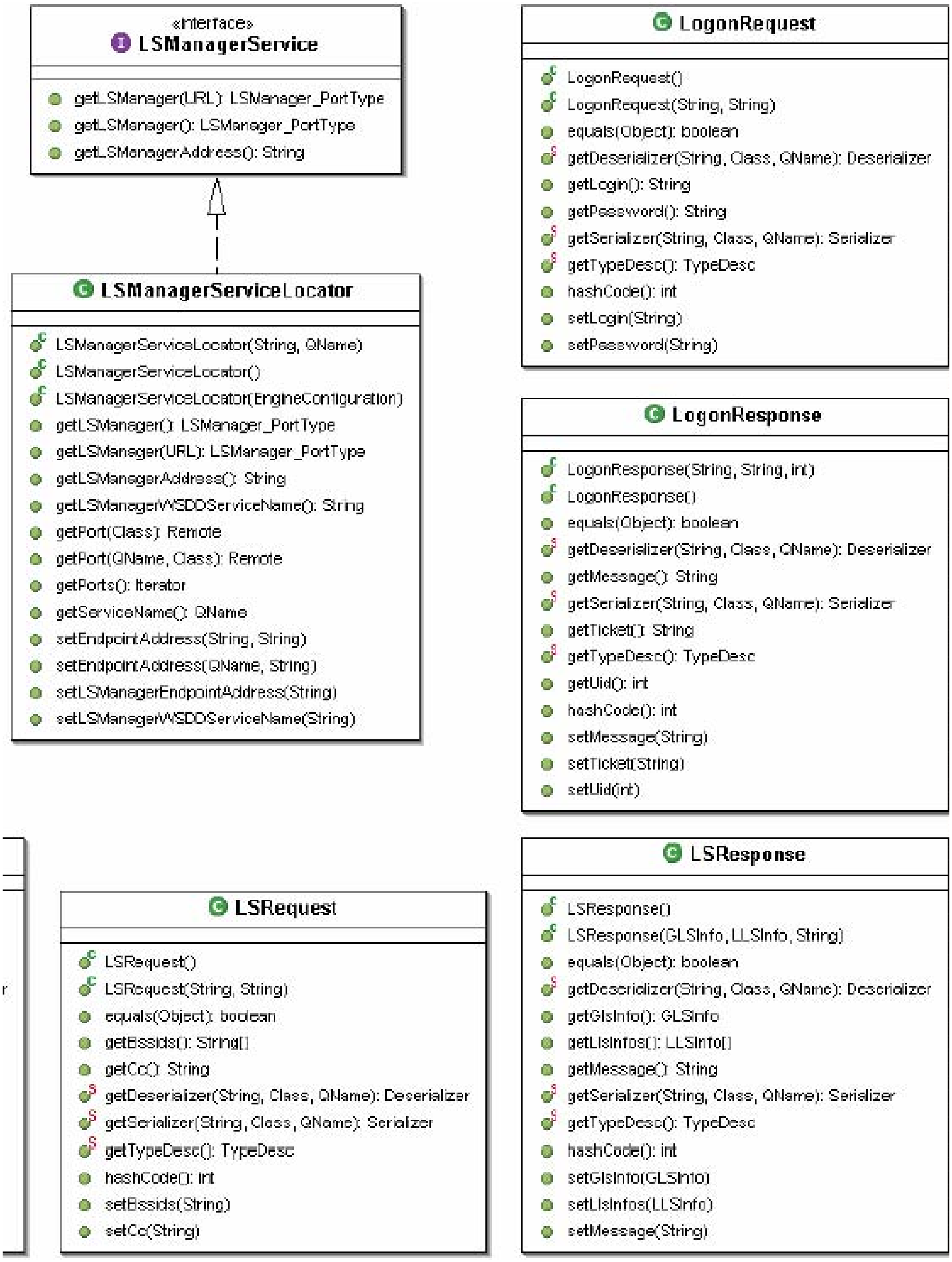


Figure F 3 Class diagram of the LCS prototype (Part 2)

F2 The GLES prototype implementation

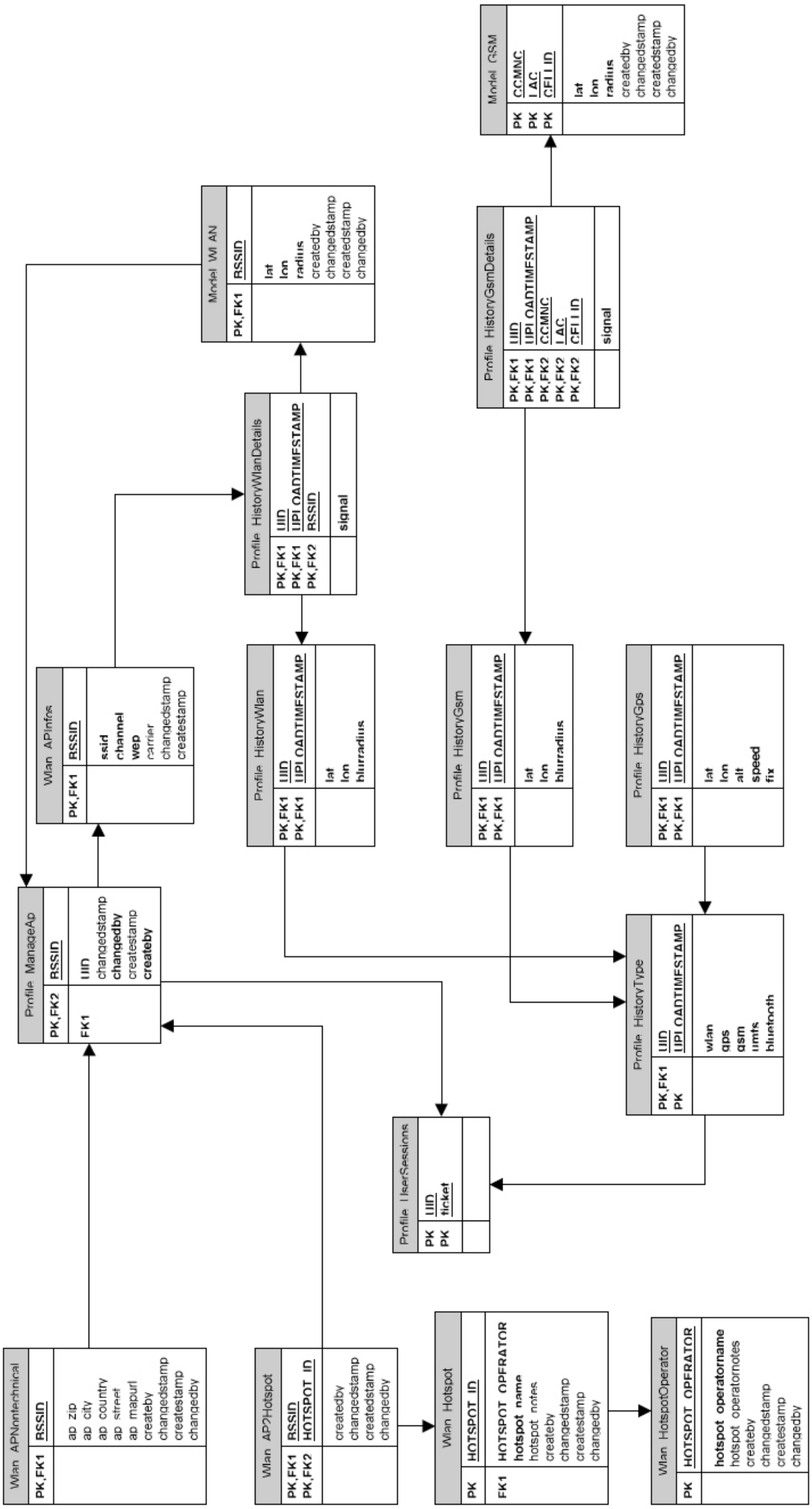


Figure F 4 The database scheme of the GLES prototype

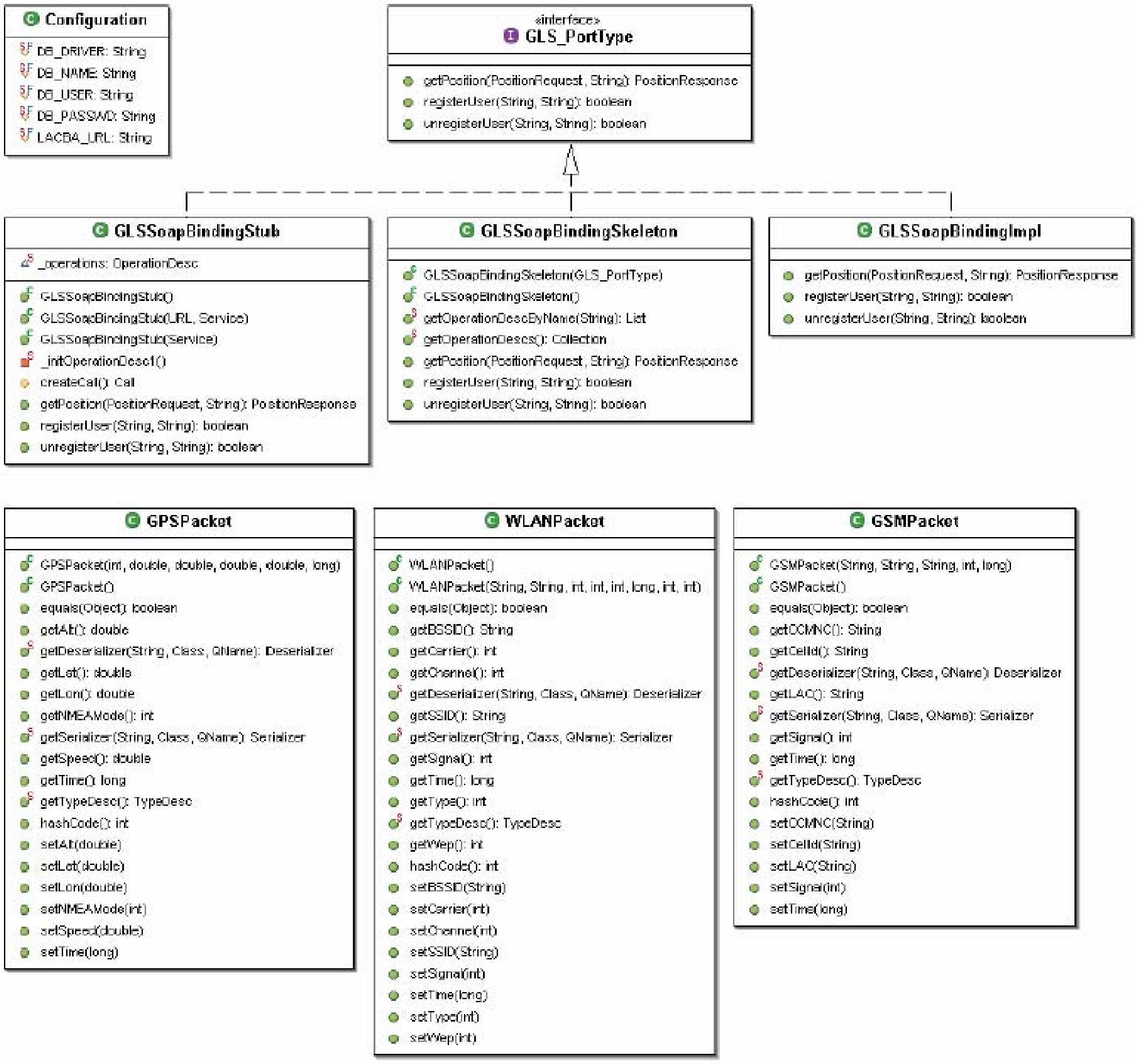


Figure F 5 Class diagram of the GLES prototype (Part 1)

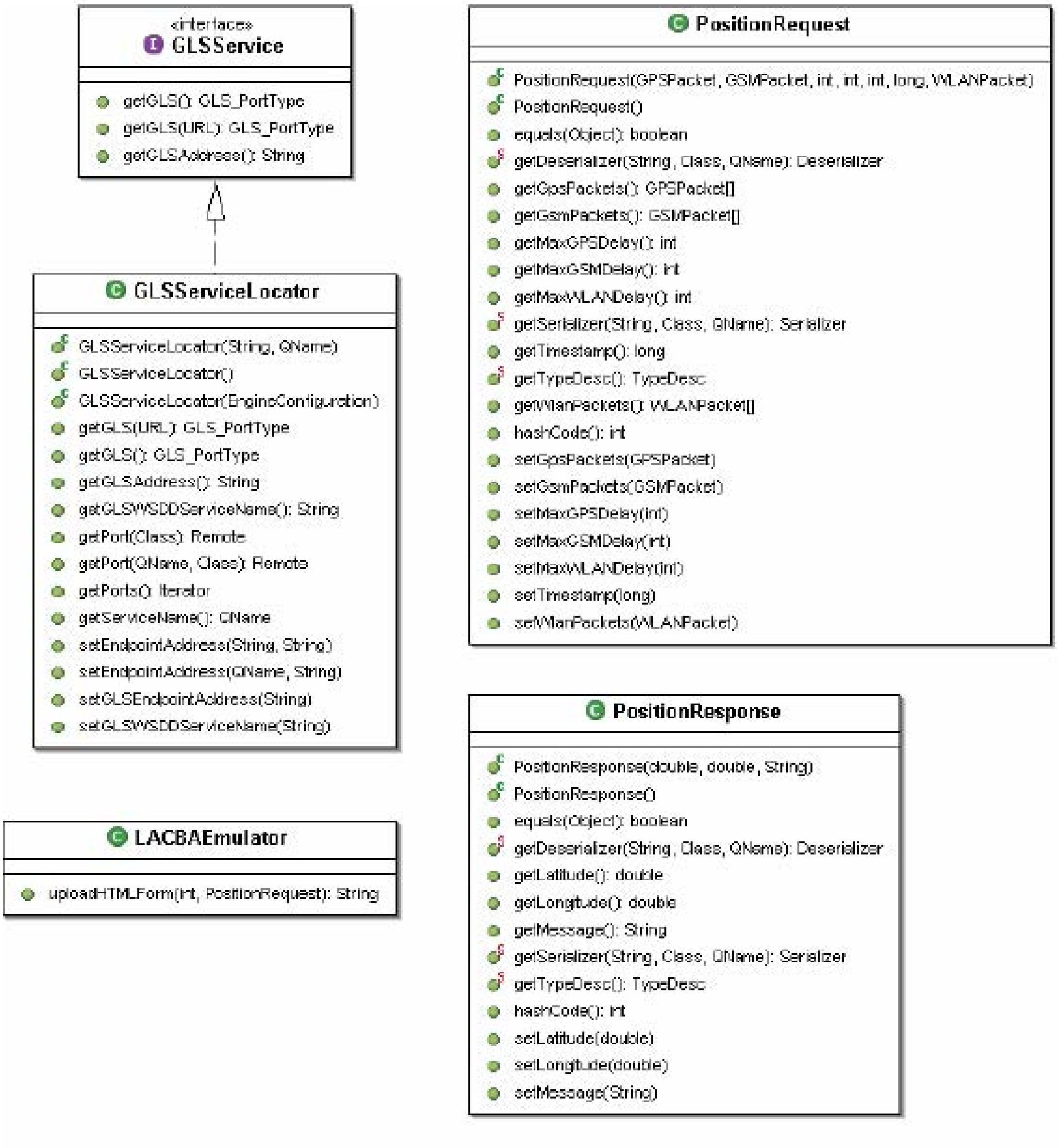


Figure F 6 Class diagram of the GLES prototype (Part 2)

F3 The LLES and LGIS Prototype Implementation

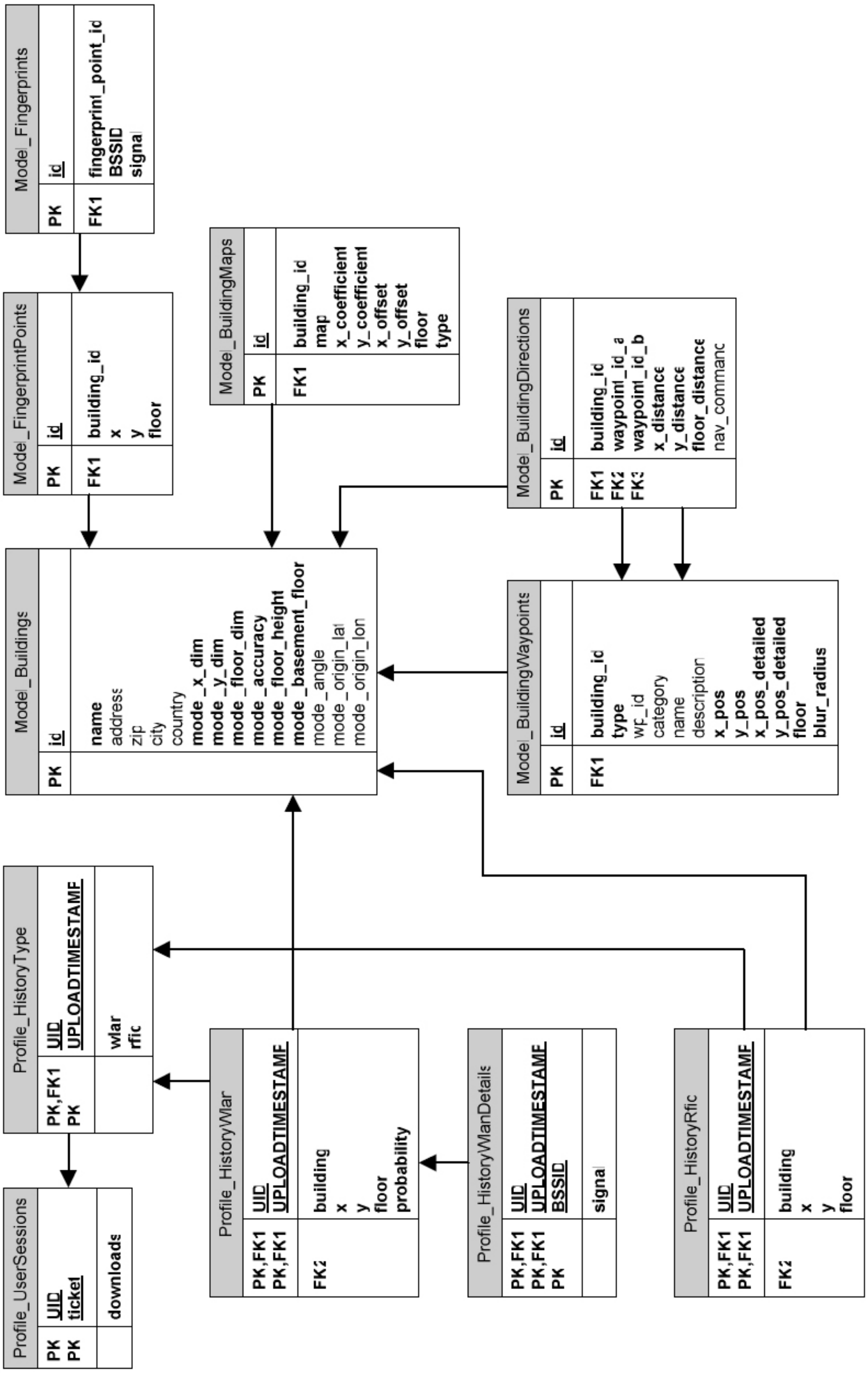


Figure F 7 The database scheme of the LLES and LGIS prototype

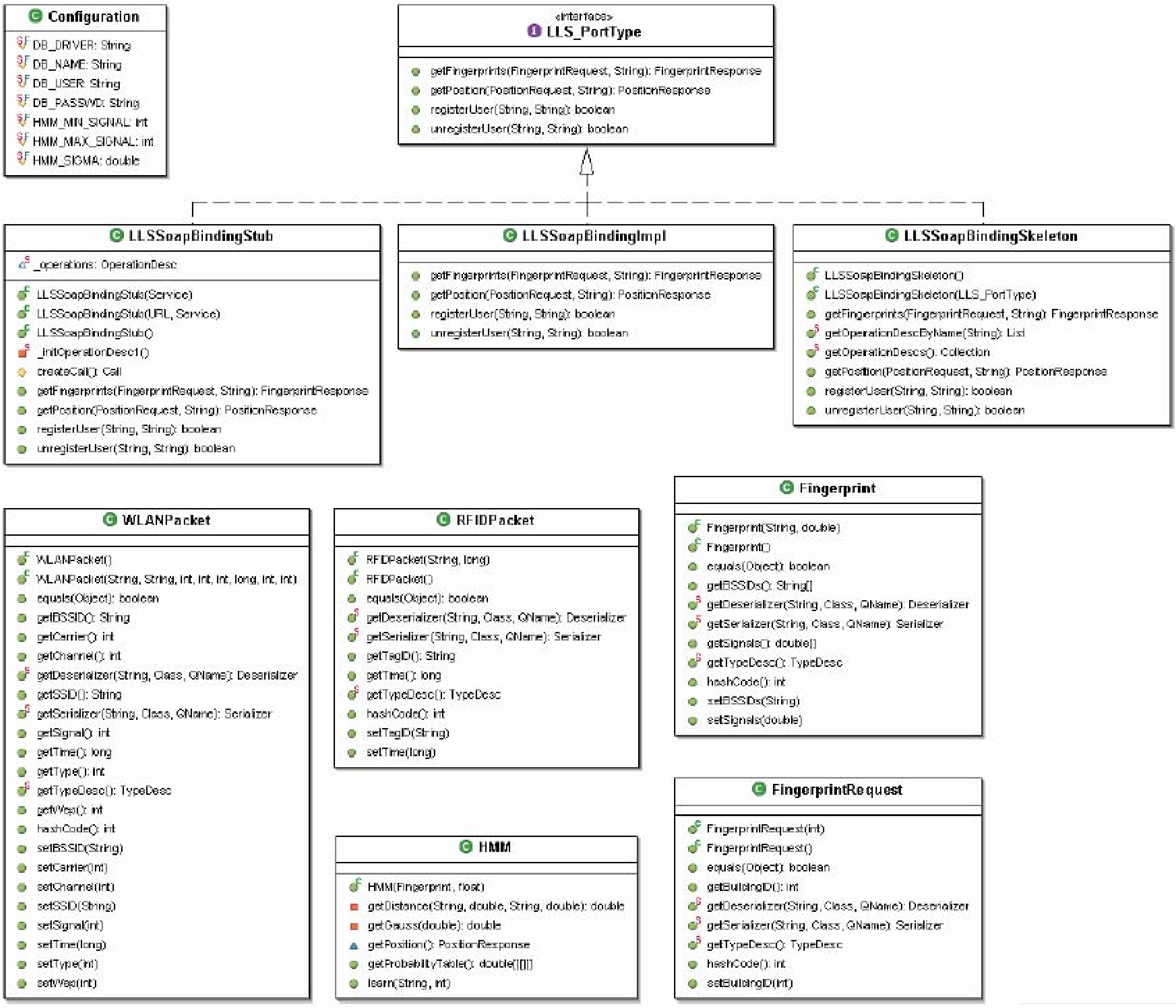


Figure F 8 Class Diagram of the LLES prototype (Part 1)

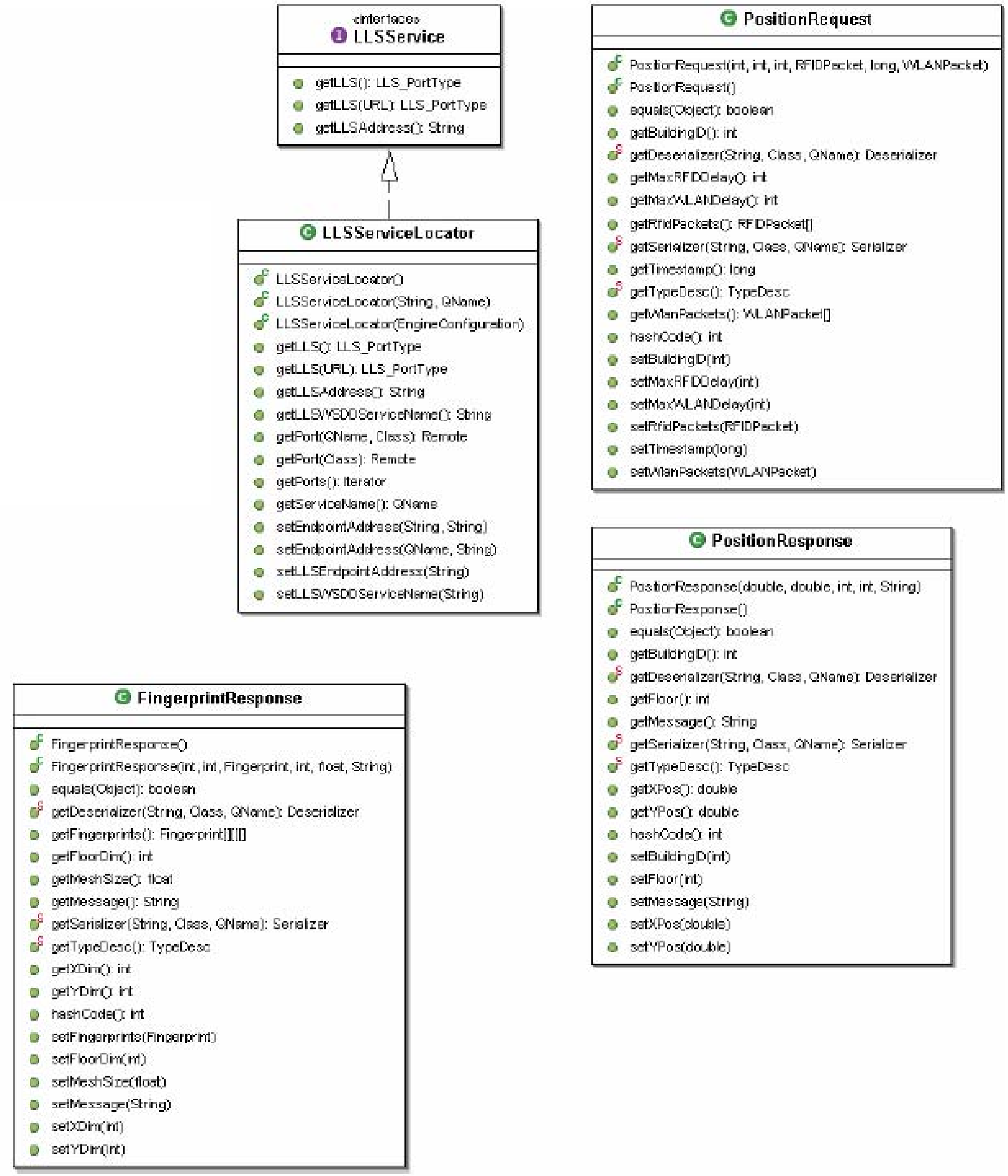


Figure F 9 Class Diagram of the LLES prototype (Part 2)

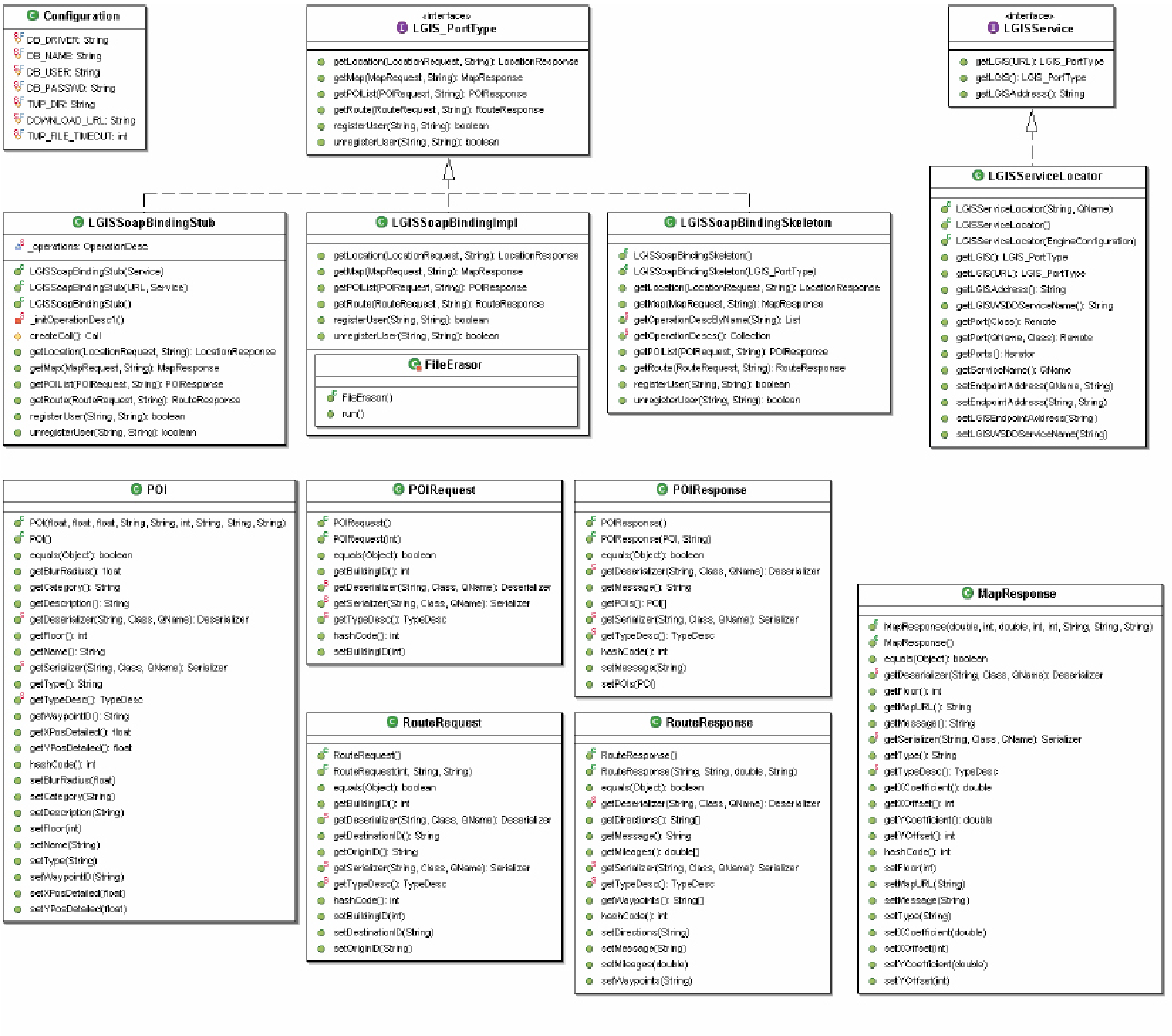


Figure F 10 Class diagram of the LGIS prototype (Part 1)

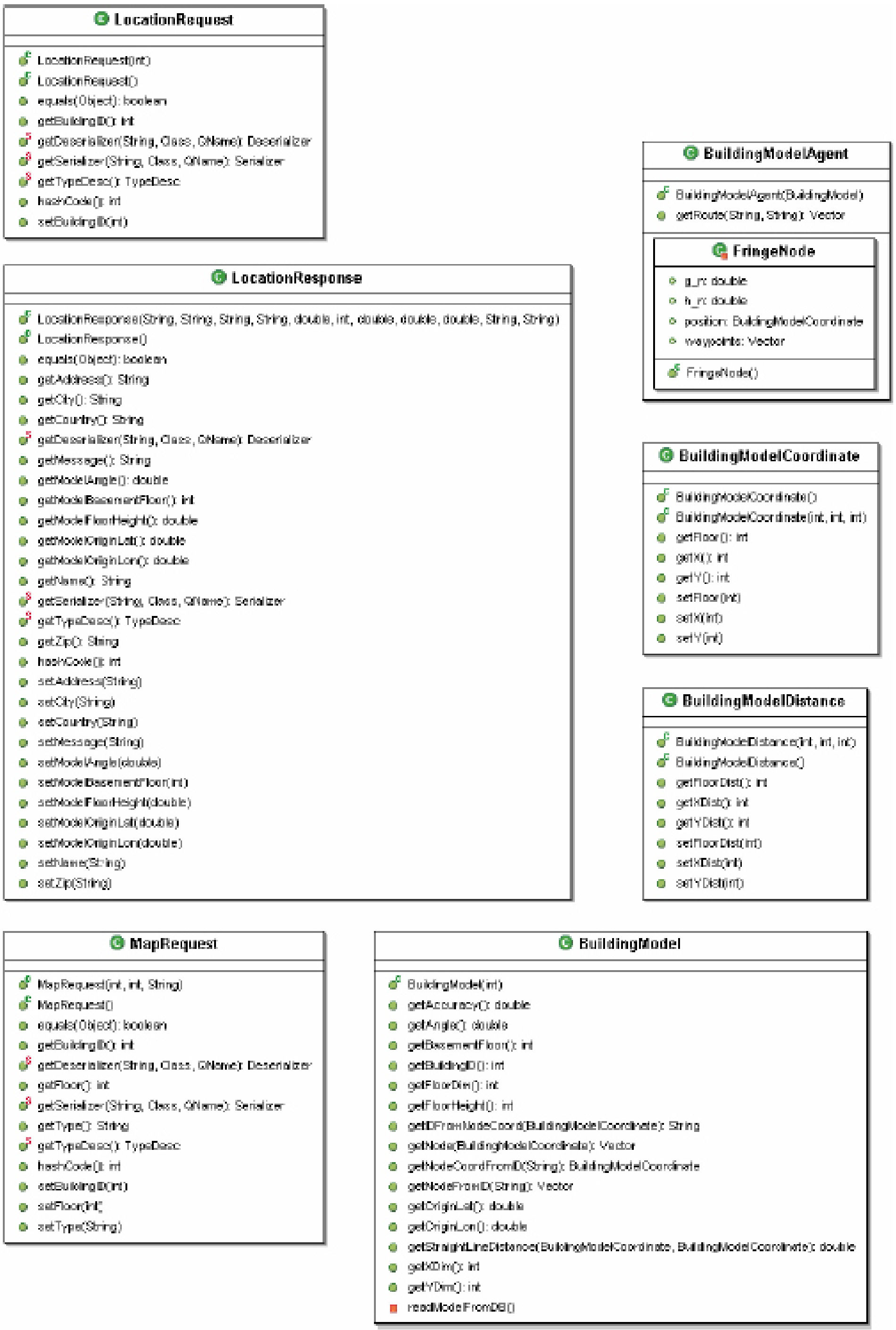


Figure F 11 Class diagram of the LGIS prototype (Part 2)

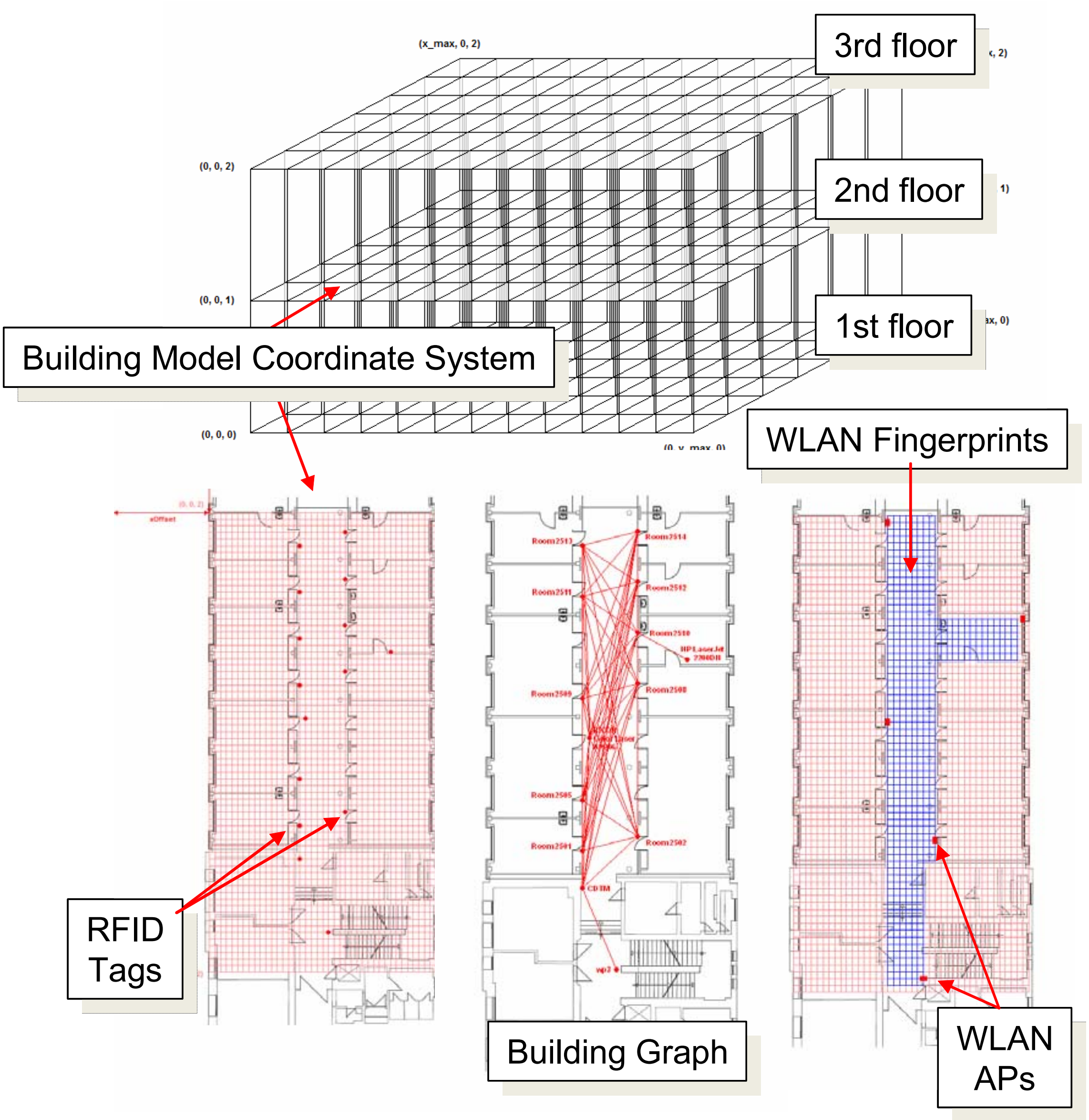


Figure F 12 A brief illustration of the LGIS building model

Figure F.8 briefly illustrates the building model used in the LGIS. A 3D topology represents the building structure of the building the CDTM is situated at TU Munich. On the 2D coordinate plane a grid of mesh size 0.5 meters is used for each building floor. The third coordinate fulfilling the 3D topology is the floor number ranging from zero to two (i.e. ground up to second floor). RFID tag and WLAN fingerprint markers recorded during initialization are mapped to the respective coordinates in the building model representing the location where the measurement was taken. Both marker types are used as waypoints and POI area definitions on the 3D grid on performing indoor navigation or providing additional location-based content (e.g. door signs and movable equipment such as computers or printers, etc.). The initialized physical space providing localization and location content spans the entire hallway and one room of the CDTM (on the 2nd floor) as well as the entire staircase from the ground up to the 2nd floor. The building model coordinate system's origin (0,0) maps to the global WGS84 coordinate system, hence allowing for a direct location information translation.

F4 The LACBA Client Prototype Implementation

The ideal development platform providing the necessary communication and sensor interface access and flexibility was chosen to be a standard commercially available notebook PC (Dell Inspiron 8100, Pentium III at 1.1GHz and 512MB RAM) running Knoppix Linux 3.7 operating system. A picture of the working LACBA client prototype with attached network/positioning devices can be seen in Figure F.9. An UMTS mobile phone (Nokia 6630) is used for ubiquitous data communication to the LACBA LCS and LP components. The GSM mobile phone (Nokia 6230) is used to gather GSM Cell ID information. Due to the USB serial port emulation under Linux it has not been feasible to simultaneously run an Internet data connection and gather Cell-ID information over the same serial connection i.e. mobile phone, hence requiring two devices. Furthermore, an external Netgear USB 802.11b device with a Prism2 chipset for WLAN data packet scanning, a NaviLock USB GPS mouse, and a PCMCIA Tagnology RFID reader are used. It can be expected that future multimode mobile devices will have these most common technologies integrated and be accessible to applications in a sufficient manner.

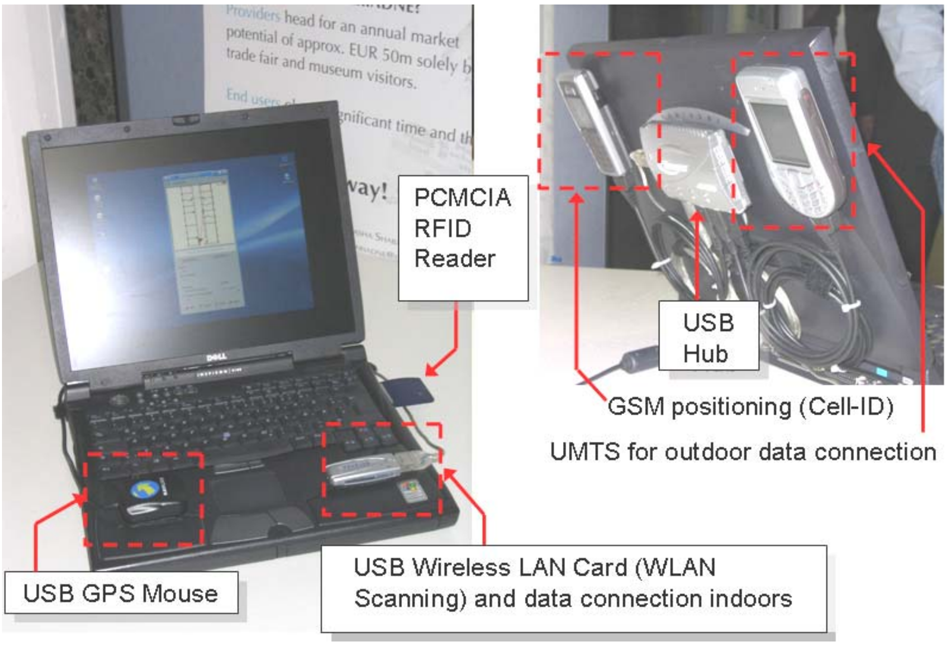


Figure F 13 The LACBA client prototype

The Ariadne navigation application was implemented running locally on the LACBA client prototype platform. Screenshots of the navigation GUI can be seen in Figure F.10. On the leftmost side, it shows the navigation using the GGIS and GLES LACBA server components, using GPS and GSM (as backup) for localization. The user triggers the handover manually via the GUI dialogue box (middle). The indoor-based navigation is shown on the rightmost side, using the LGIS and LLES server components and WLAN for localization. GSM is still available for backup but cannot be used using the LGIS, hence a switch to the GGIS necessary in the backup case. Last but not least, RFID is used as location triggers in the event of a RFID tag detection by the LACBA client.

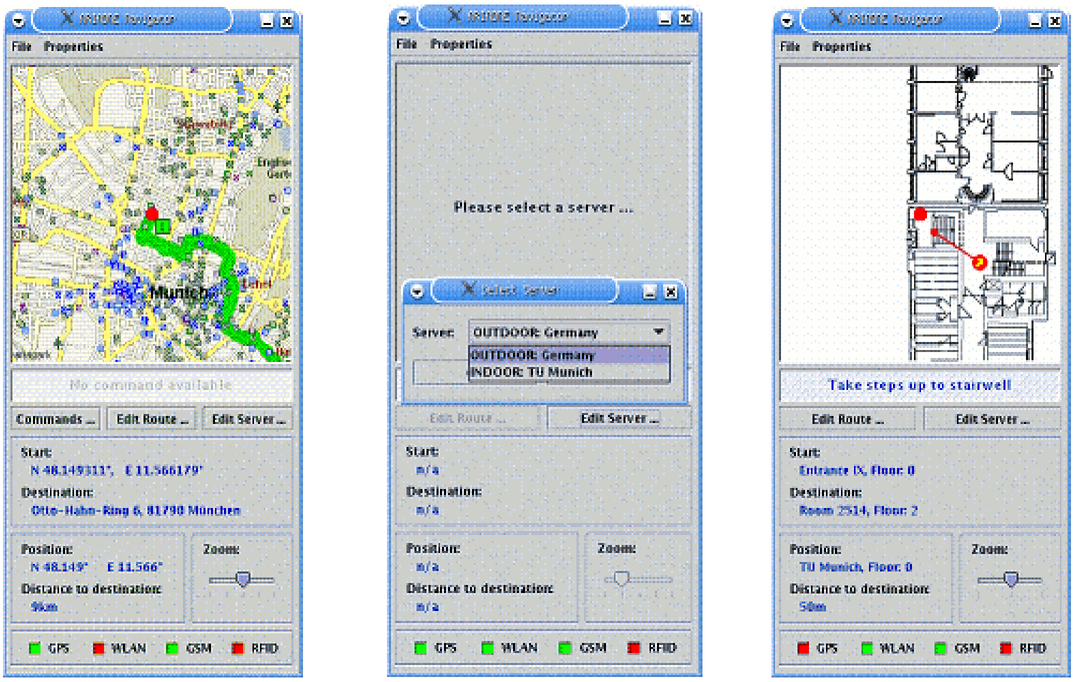


Figure F 14 The Ariadne GUI performing a handover from outdoors to indoors (left to right)