Location-based Service with Spatial Data Analysis within IP Multimedia Subsystem

Jari Kellokoski, Oleksandr Ivannikov, Timo Hämäläinen
Department of Mathematical Information Technology
University of Jyväskylä
FI-40014 Jyväskylä, Finland
Email: jari.k.kellokoski, oleksandr.v.ivannikov, and timo.t.hamalainen @jyu.fi

Kari Luostarinen
Metso Paper Inc.
P.O. Box 587
FI-40101 Jyväskylä, Finland
Email: kari.luostarinen@metso.com

***Abstract*—Nowadays applications and services, which utilize information about client's geographical position, are becoming more and more popular. Such kind of applications are usually called Location-based services (LBS). Location-Based Service technology has great potential to add value to existing or new wireless mobile data services and for example for LBS developers, mobile advertising and spatial data providers. Virtually all of the big mobile device and services-oriented companies are acquiring start-up companies dedicated to mobile advertising based on location data. The IP Multimedia Subsystem (IMS), on the other hand, is meant to be global and access-independent and to have all Internet protocol (IP) based connectivity and service control architecture, as defined originally by 3GPP in their Release 5. Since that time, it also has evolved to cover the Fixed Mobile Convergence (FMC) and provides reliable charging, security and quality of service. This research investigates the methods of analyzing spatial data (movement pattern extraction) and shows how "smart" location services can be build on top of the IP Multimedia Subsystem. It shows that open Internet standards and open source solutions can be utilized to build an LBS service on top of IMS. As the standards are open, it is easy for anyone to see what's behind the service and satisfy themselves that no proprietary solution has been used. The proposed software architecture opens possibility to add other models for spatial data analysis.**

Keywords*—*Location-based services; IP Multimedia Subsystem; spatial data analysis.

I. INTRODUCTION

The IP Multimedia Subsystem was a significant step forward to integration of different multimedia services into a single infrastructure which allows the user to access those services via a common interface and one (Session Initiation Protocol - SIP [3]) account. Unfortunately, location services were not included into the IMS core, and the current solutions do not allow the use of geographical location within IMS services. LBS differ quite a lot in their functionality and in their approach for location data processing. Some of them are comparatively "simple" and inform mainly on the current position of object. Other services, which are more "intelligent" deal with a set of location data that usually represents the history of movement for a certain period and object. By analyzing such kind of data, it is possible to find some rules which describe how an object behaves in space-time dimensions. The aim of present research is twofold. First, to develop location-based service that collects movement history data and analyzes it in order to predict the future location of an object. Second, to connect that service to IMS and show how both of these technologies can work together. By combining these, the common software architecture is created that can be easily expanded by adding additional modules to the location-based service. The prediction of the future location is then linked with timestamps. As a result of this, the location-based service can deliver services beforehand. This brings great value in case of advertising as they would not have to wait until the user is in the vicinity of the service. Instead the ads could be sent based on the history prediction. Sections II and III are giving overview of the location-based services and the IMS, while Section IV presents the proposed architecture and implementation.

II. RELATED WORK ON LOCATION BASED SERVICE

A Location-based service infrastucture can be accessed with a mobile device utilizing geographical position data. The key to LBS is to know the location of the user, so that an appropriate service can be delivered. However, that location might not necessarily have to be related to the current position but could also be some future location of the user. The most common examples of a location-based services are: turn-by-turn navigation, locating people, requesting the nearest service point. Nowadays location-based services can be used in combination with messaging, for example in mobile advertising.

A. Positioning technologies

Today, there are many ways to get position information for the mobile device. There are many terrestrial systems based on radio signals including Angle of Arrival, Time of Arrival, Time Difference of Arrival and satellite-based systems such as Global Positioning System which is the best-known and the only fully operational satellite positioning system [25]. Assisted-GPS [6] is used in the implementation part of this research.

B. Building blocks for LBS

A Location-based service infrastructure consists of several components: mobile device (User-side device that is able to use location-based services), mobile network, service and

 ISBN: 978-1-61208-148-9

content provider (Owner of a geographical data storage; offers actual location-based services), positioning system. Detailed investigation of mobile devices and mobile network components is not within the scope of the current research. Mapping data that content providers use is discussed further in this chapter. LBS applications typically use information from several content databases: the road network (digital maps); business and landmark information, often referred to as Yellow Pages, or Point-of-Interest (POI) information; and dynamic data such as traffic and weather reports. One additional data type used in this research is history data on the user's movement over time.

C. Maps data

Building LBS applications starts with the collection of road data. Map database vendors collect and convert raw geographic content into digital formats. The map data are captured in many ways, ranging from satellite imagery to scanned maps to manually digitized paper maps. Some vendors physically drive along each road segment in a GPS-equipped car, recording every change of direction and photographing road signs be able to provide information on specific road conditions such as turns and height/weight restrictions. Each vendor's data are different, which accounts for some of the discrepancy in the maps and routes generated. One of the easiest ways of getting maps data and using it to utilize the application programming interface (API) of some known available maps applications such as Google Maps [9] or Microsoft Bing Maps [12]. This research employs the Google Maps API.

D. Point-of-Interest Information

One of the most popular LBS applications is Yellow Pages, or concierge services. Mobile concierge-type services help users locate businesses near a specified location. These services help answer questions such as: "Where is the airport?" or "Where is the nearest gas station?" Concierge applications use business and landmark information that has been compiled into POI databases. Integrating a map database with a POI database creates a detailed, digital representation of a road network and business services available along it. These POI databases contain the kind of detailed information typically found in a phone directory and add value to the map database's geographic content [23].

E. Movement Patterns

By definition, moving objects are entities whose positions or geometric attributes change over time. However, in many cases the dimension of an object is not as important as its position. Hence, moving objects are considered as moving points, whose trajectories (paths through space and time) can be visualized and analyzed. In most cases, moving object data sets are quite large. Therefore, it is necessary to develop efficient data mining algorithms in order to extract useful information. This information contains knowledge about object behavior, which is known as movement pattern. Generally, movement patterns include any recognizable spatial and temporal regularity or any interesting relationship in a set of movement data [24]. Every movement, by nature, has several spatial, temporal and spatio-temporal parameters. In case of spatial movement, the parameters are position, distance, and direction. In case of temporary movement, the additional parameters are duration and travel time. And in case of spatio-temporal movement, the additional parameters are speed and acceleration. The importance of each characteristic depends on domain and the problem we need to solve. In other words, in most cases there is no need to record and analyze all information about object movement. In addition to movement parameters mentioned above, we should consider a set of other environment variables, which influence the choice of pattern extraction algorithm. One of the most significant factors is the number of objects involved with movement (individual or group movement). However, it is important to understand the difference between groups which consist of independent objects and groups in which objects are linked to each other. In the first case, pattern extraction task can be divided into subtasks, and extraction an algorithm can be applied to each object separately. In the second case, we deal with the behavior of the entire group. This type of movement is usually much more difficult to analyze, and advanced data mining techniques should be applied [10]. Another important factor to consider before collecting information about object movement is path type. Paths of movement objects may take different forms. Most objects travel more or less continuously, generating a continuous path (a pedestrian, a car moving on a road). Such a continuous path is typically discretized into regular steps prior to computing the movement parameters. This research employs the individual and continuous movement.

III. IP MULTIMEDIA SUBSYSTEM

In order to support universal IP connectivity the IMS should be able to use multiple transport technologies to guaranteed connectivity. So regardless of the underlying access network or the user's terminal features, IMS-related services should be usable. The IMS can be seen as a glue between different services [7]. The OpenIMS core [14] has offered an open source based possibility to try out the IMS deployment. Although the OpenIMS is not a commercial solution its performance can be considered to be suitable for smaller players in the telecommunications business [5], [4]. Other additional reasons for the IMS deployment are:

- IMS offers, for 3GPP/ETSI-TISPAN, a standardized and developed IP system environment containing standard interfaces to external systems and networks
- IMS contains, as a default Authentication, the Authorization and Accounting (AAA) system for user identification and service authorization.
- IMS enables SIP/Voice over IP (VoIP) capability by default when using public or private WLAN networks anywhere in the world, and usually comes with a single flat rate fee.

The core elements of the IMS architecture are called: Call Session Control Functions (CSCF). There are three main CSCFs:

 ISBN: 978-1-61208-148-9

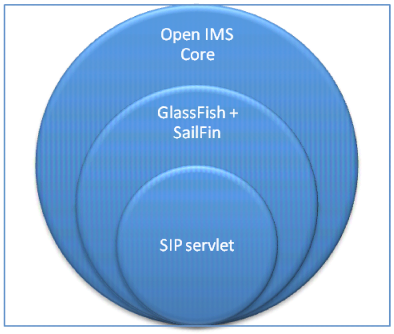


Fig. 1. The LBS integration

Proxy CSCF (P-CSCF), Serving CSCF (S-CSCF), Interrogating CSCF (I-CSCF). Each of these has its own special tasks [22]. Proxy-CSCF is the first contact point for users within IMS. This means that all SIP signaling traffic from the users terminal will be sent to the P-CSCF. S-CSCF is responsible for handling registration processes, making routing decisions, maintaining session states, and storing service profiles. I-CSCF is a contact point within the service operator's network for all connections destined to a subscriber of that network operator. The Home Subscriber Server (HSS) can be considered as the main data storage for all subscriber and service-related data of the IMS. The most important data that the HSS holds is user identities, registration information, access parameters and service triggering information [1]. For the research the Open IMS Core implementation was deployed. This implementation used only the basic IMS Core elements, consisting Proxy-CSCF, Serving-CSCF Interrogating-CSCF and HSS, for the test platform. These elements are made by the Open IMS Core project by Fraunhofer Fokus [14]. The CSCFs used were from Open IMS core project SVN version 732 and HSS was Java based FHoSS (Fokus Home Subscriber Server) running with MySQL server version 5.0.51a. Java used was that of Sun Microsystems Java Runtime Environment 1.6.0 update 12. The hardware configuration had the following characteristics; XCSCF servers and HSS located in the same virtualization based Linux PC framework; Server Dell PowerEdge 2950 with 2 Intel Xeon E5420 processors and 32 GB ECC-DDR2 memory; 4 750 GB SAS disks using RAID 10 configuration. Virtualization was done with Citrix's XenServer 5.0.0. A virtual machine was configured with 2 virtual processors, each using a 4096 MB main memory, 15GB of disk memory and one 1Gb Ethernet port.

IV. Proposed LBS architecture for the LBS Application Server

IMS architecture supports the extension of core functionality with the help of application servers (AS). Thus, integrating LBS into IMS means implementing and configuring the application server, which will function as part of IMS. In the current research, the GlassFish application server and the SailFin SIP servlet container were chosen as the deployment platform for

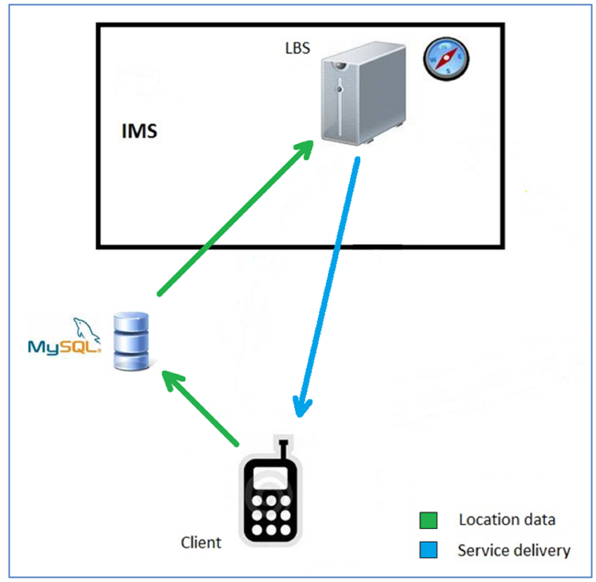


Fig. 2. The data flows

a location-based service (application server) [18], [11], [21]. To test the LBS functionality in a real IMS environment, the Open IMS Core implementation was deployed as described in chapter III. The LBS itself is implemented as a SIP servlet [2] and written in Java. The implementation environment had following characteristics Sailfin 1.0 which is based on GlassFish v2.1 (Java EE 5) and SIP Sevlet API 1.1 (JSR 289). This developed application server was based on Intel Core 2 Duo E8400 CPU with a 4 GB DDR2 memory and a 1Gb Ethernet port. The operating system was Debian GNU/Linux 5.0. The integration approach described above is shown in Fig. 1. Another important part of the architecture is the client. In the context of a developed system, the client is a mobile device with an installed application, which "understands" the SIP protocol and periodically sends the user's location data. However, for testing purposes these requirements could be simplified. During the development phase of the SIP Test Agent plug-in for NetBeans IDE, was used to send SIP messages to a location-based service [20]. As shown in Fig. 2, the interconnection between the Client and IMS consists of 3 components (Client, LBS and MySQL database [19] and of 2 data flows (collecting location data and providing the service). The MySQL database includes 2 major tables: the table of users ("users") and the table of services ("services"). The purpose of the "services" table is to provide information about POIs during the service delivery phase. The "Users" table is used for saving the user's data, such as: location information (latitude, longitude), the name of the user, presence status ("online" or "offline") and some other fields. The important feature of the database implementation is that the the "users" table "knows" only the current position of the client at a particular moment. The application Server (Location-based

 ISBN: 978-1-61208-148-9

service) scans the table once every minute to obtain location data and merges all the coordinates into one route. LBS then uses this information to create a user profile and to choose services with the best match with the user's everyday routes. When enough statistical data have been collected, the application server switches to the service mode, in which mode it sends advertising messages to the client. Position data saved on the application server is in the Extensible Markup Language (XML) [8] format and tagged for latitude, longitude, date and time.

A. Implementation of the LBS Application Server History Analysis Module

The main idea of the application server functionality is to collect location / time information about client, and based on this data, try to predict the user's movements in the nearest future. In anticipation of these movements, advertising message will be sent to the user with a POI description. The client's routes are saved in the XML format and include a set of "coordinates" tags such as latitude, longitude, date and time. In addition, the user should also provide information about home location and the location of work office, so that the user's movement pattern can be extracted correctly. Fig. 3 gives a general view of the application server functionality. In the figure "History" means the user's routes that were described above (a set of XML files). The "Spatial data analysis" part will be covered later in this chapter. The POIs database is implemented with the help of the MySQL server, thus application server will connect to it to obtain needed information. Finally, "Potential services" is an intersection between the POIs database and the user's movement pattern. Preparation for the extraction of a movement pattern consists of several steps. First, a spatial area between the home and work locations should be divided into regions (Fig. 4). When choosing the size of that region, we should try to find a compromise [17]. The region should be small enough to be accurate, but should not be too small to avoid generation of unnecessary data and to provide better pattern extraction. In the current research, region width equals a 0.003 longitude coordinate difference (about 200 meters) and region height equals a 0.002 latitude coordinate difference (about 220 meters). Second, we should save the regions in a format suitable for data processing. Thus we save them in a matrix in which each region is represented by 4 borders so that there are left, right, top and bottom borders. Third, we parse the history data of the user's route and represent this information in a similar (tabular) way, including latitude, longitude, date and time. Fourth, we should find the correspondence between the latitude/longitude coordinates in the movement history table and region number (region number equals row index in the regions table) so that there would be region, date and time. This can be achieved by checking a simple condition: "latitude (top border AND latitude) bottom border AND longitude (right border AND longitude) left border". We then divide all the history data by days, so a sequence of regions can be composed for each day (Fig. 5). Finally, we create a table of time stamps. It

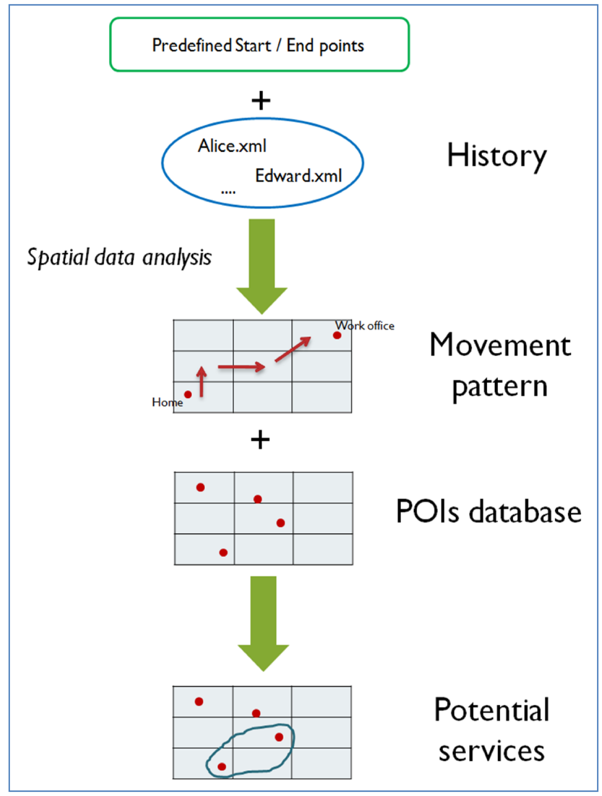


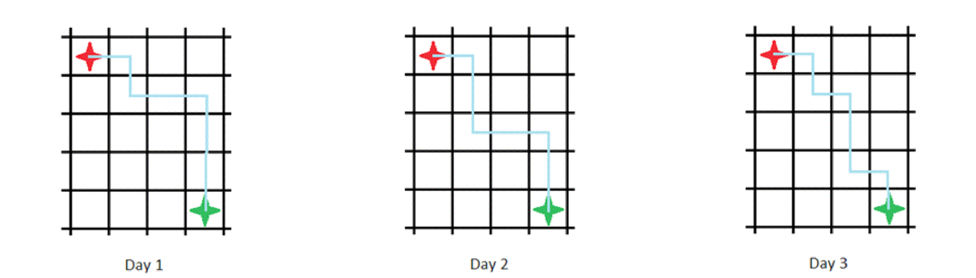
Fig. 3. Application server functionality in general level

has the same dimensions as the table of region sequence, but consists of corresponding time values. Now everything is ready for the extraction of movement pattern. Movement pattern means relations between the present location of a moving object, its past location(s) and time. There are several techniques and algorithms to find out the rules according to which the object moves. These algorithms are dependent on the application domain and information needed in a particular situation [13]. In our current research we were not so interested about relations between location/time values inside a particular sequence. What we needed to know is "What are the common (the most frequent) locations (and corresponding time stamps) of an object during a trip from home to work (from work to home)?". The movement pattern extraction algorithm goes as follows: input is the region sequence table, output is the movement pattern. And for:

1) Movement pattern = day 1 sequence
2) For each region in sequence if (region in movement pattern NOT EQUAL region in next day sequence) region in movement pattern = UNKNOWN
3) Repeat step 2 for next day sequence

After applying this algorithm to the example shown in Fig. 5, we will get the next pattern: 1, 2, 6, UNKNOWN, UNKNOWN, UNKNOWN, 16, 20. To complement this information with time values, we calculate the average time using

 ISBN: 978-1-61208-148-9



Day	Region	Region	Region	Region	Region	Region	Region	Region
1	1	2	6	7	8	12	16	20
2	1	2	6	10	11	12	16	20
3	1	2	6	7	11	15	16	20
...	...	...	...	...	...	...	...	...

Fig. 5. Example of region sequence per day

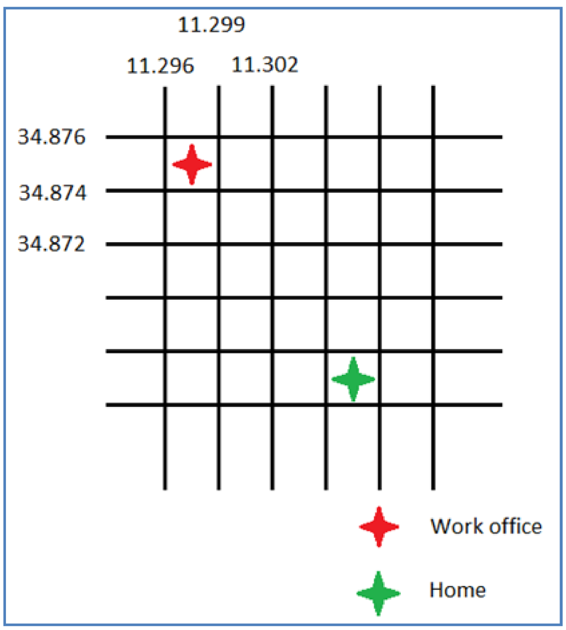


Fig. 4. Spatial regions

the time stamps table. For example, the average time for region 1 could be: (16.45 + 16.50 + 16.51) / 3 = 16.48. After calculating all the average values we will get the next time sequence:16.48, 16.49, 16.50, NOT NEEDED, NOT NEEDED, NOT NEEDED, 16.55, 16.57.

B. Overall architecture of the LBS system with Client, Server and Application Server

In order to get real history data to the application server, a supporting client and servers were implemented. This way real movement data could be collected and the analysis would be based on real-life data. There were also other reasons for having a proper client and server for LBS. The server implementation resembles the application server described previously. It runs on top of Open IMS and was deployed into the Glassfish and Sailfin combination. The relevant features of the server implementation are:

- Provides administration tasks for the whole LBS implementation. It has a web interface for administrative tasks such as manage users and POIs
- Provides a map view presentation of the existing users and POIs
- Can add tasks for a specific user group and can monitor the progress of the task, which is location-dependent and requires actions from the user

The client is mobile device based, and in this case it is implemented with Qt [16] and runs on top of the Maemo 5 [15] platform in Nokia N900 device. The benefits of the N900 device are: built-in IMS support through the SIP protocol and a capable web browser environment. Thus it is easy to connect to IMS services and Google maps data. The main features of the client implementation are:

- User is able to see him/herself, friends and POIs on the map
- User is able to interact with the tasks that are given: one can view, accept and set them as done
- User can communicate via IMS calls and messages with other users

The overall architecture is shown in Fig. 6.

C. Architecture and Implementation Considerations

To get better understanding of the proposed architecture and implementation done, the following aspects were discussed; performance, history data analysis module and need for any further modules. The performance of the Open IMS Core is, as previously mentioned, studied in [5], [4]. Based on these studies Open IMS Core performance was adequate very well for small enterprise usage. This research utilized the same

 ISBN: 978-1-61208-148-9

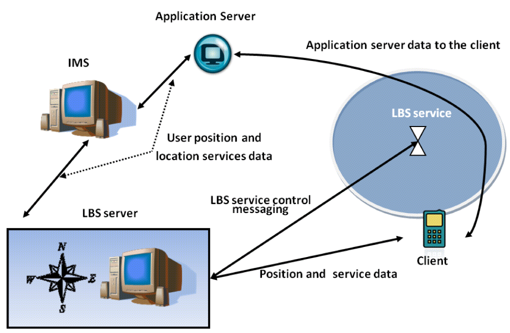


Fig. 6. Overall LBS architecture

hardware configuration as in the previous performance measurements. Therefore, overall hardware and software configuration was considered to be sufficient for a robust environment. In the history analysis module the regional split (see Fig. 4) was chosen to be similar then the average POI radius in LBS service. The service POI area was defined as a radius but for the history pattern analysis the region was divided to rectangle areas. For pedestrian movement analysis this arrangement worked fine. However, it may need to be reconsidered when applying algorithm for other forms of traffic such as car traffic. Performance of the algorithm was not any issue with tens of real users. However the scalability of the current implementation will be part of the future work.The presented history analysis module provided needed information that the extraction algorithm itself works. In addition, other models could be added to further enrich the feature set of the LBS application server. For example algorithm that detects the end points of user's routes could be useful in the same context as the presented model. The same applies for the detection of the user's movement type and the detection or if the user is moving somewhere else then usually (traveling). All these further details could be build as a separate modules of the LBS application server.

V. CONCLUSION AND FUTURE WORK

We have presented a location-based service architecture and implementation. The architecture was build on top of an Open IMS implementation and other open protocols and standards like SIP and XML. The spatial data analysis based on the user's movement history was possible and it provided, for example, a way to find services that are presumably located on the route that the user moves along between work and home. Timestamps are added so that it is possible to estimate the time when the user is taking this route next time. All this information is potentially very valuable for advertisement purposes, for example. Our future research continues with fine tuning of the movement pattern algorithm and adding other algorithms to the developed location-based service module. The aim is that the application would automatically find the correct end locations and detect the corresponding transport method user is using so that they would not need to be marked manually.

REFERENCES

[1] 3GPP. IP Multimedia Subsystem (IMS); Stage 2. TS 23.228, 3rd Generation Partnership Project (3GPP), September 2008.

[2] C. Chrighton, D. T. Long, and D. C. Page. Jain slee vs sip servlet which is the best choice for an ims application server? *Telecommunication Networks and Applications Conference*, pages 448–453, November 2008.

[3] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), June 2002. Updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630.

[4] J. Kellokoski, E. Tukia, E. Wallenius, T. Hämäläinen, and J. Naarmala. Call and messaging performance comparison between ims and sip networks. *Internet Multimedia Services Architecture And Applications (IMSAA), 2010*, pages 1–5, December 2010.

[5] J. Kellokoski, E. Tukia, E. Wallenius, T. Hämäläinen, and J. Naarmala. Registration performance comparison between ip multimedia subsystem and session initiation protocol networks. *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010*, pages 20–25, October 2010.

[6] T. Goze, O. Bayrak, M. Barut, and M.O. Sunay. Secure user-plane location (supl) architecture for assisted gps (a-gps). *Advanced Satellite Mobile Systems, 2008. ASMS 2008. 4th*, page 229, September 2008.

[7] V. Leung, T. Wong, and P. TalebiFard. Breaking the silos - access and service convergence over the mobile internet. *MSWiM'08: Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 286–293, October 2008.

[8] World Wide Web Consortium. Extensible markup language (xml) 1.0 (fifth edition). http://www.w3.org/TR/xml/, November 2008.

[9] Google. Google maps api family. http://code.google.com/apis/maps/index.html, May 2011.

[10] R. Haining. Spatial data analysis: Theory and practice. *International Journal of Geographical Information Science*, pages 300–301, April 2004.

[11] H. Khlifi and J.-C. Gregoire. Ims application servers: Roles, requirements, and implementation technologies. *Internet Computing, IEEE, vol 12*, pages 40–51, May 2008.

[12] Microsoft. Bing maps. http://www.microsoft.com/maps/, April 2011.

[13] H. Miller and H. Jiawei. *Geographic Data Mining and Knowledge Discovery*. CRC Press, Cleveland, US, 2009.

[14] Fraunhofer FOKUS NGNI. Openimscore.org. http://www.openimscore.org/, May 2011.

[15] Nokia. Meamo software platform. http://maemo.nokia.com/, May 2011.

[16] Nokia. Qt - cross platform application and ui framework. http://qt.nokia.com/, May 2011.

[17] S. Openshaw. *The Modifiable Areal Unit Problem*. Geo Books, Norwich, UK, 1984.

[18] Oracle. Glassfish, open source application server. https://glassfish.dev.java.net, May 2011.

[19] Oracle. Mysql. http://mysql.com, May 2011.

[20] Oracle. Netbeans. http://netbeans.org, May 2011.

[21] Oracle. Sailfin, sip servlet. https://sailfin.dev.java.net, May 2011.

[22] M. Poikoselkä and G. Mayer. *The IMS IP Multimedia Concepts and Services*. Wiley, Chichester, UK, 2009.

[23] J. Schiller and A. Voisard. *Location-Based Services*. Morgan Kaufmann Publishers, California, US, 2004.

[24] S. Dodge R. Weibel and A.-K. Lautenschütz. Towards a taxonomy of movement patterns. *Information Visualization*, pages 1–13, August 2008.

[25] J. Wilson and S. Fotheringham. *The Handbook of Geographic Information Science*. Wiley, Chichester, UK, 2008.

 ISBN: 978-1-61208-148-9