A Framework for Developing Mobile Location Based Applications

Andrej Krevl, Tone Vidmar, Matjaz Pancur and Mojca Ciglaric
University of Ljubljana, Faculty of Computer and Information Science
Trzaska 25
SI-1000 Ljubljana
Slovenia

{ andrej.krevl, tone.vidmar, matjaz.pancur, mojca.ciglaric } @fri.uni-lj.si

Saso Tomazic and Ales Zavec
University of Ljubljana, Faculty of Electrical Engineering
Trzaska 25
SI-1000 Ljubljana
Slovenia

{ saso.tomazic, ales.zavec } @fe.uni-lj.si

Stanko Ciglaric
Slovenian Institute of Quality and Metrology
Trpinceva ul. 39
SI-1000 Ljubljana
Slovenia

stane.ciglaric@siq.si

ABSTRACT

Location based services and applications are buzzwords nowadays, yet they have been around for quite some time in a variety of applications. However, due to high costs of the positioning equipment the number of such applications is increasing too slowly. Even though "location based" is a hot topic, there are no publicly available application development frameworks that would enable rapid location based application development on COTS hardware. The paper presents different options for determining location of mobile devices such as mobile phones and handhelds. It describes positioning possibilities using WiFi networks, GSM networks, Bluetooth beacons and the GPS system. Furthermore, it proposes an extensible and open framework for rapid mobile location based application development. The paper specifies the components comprising the framework, data structures used for spatial data interchange and Web Services used for communication among components. It also describes a location aware application prototype built on top of the proposed framework. This application displays spatial data according to the mobile device location and it provides means for entering our own location-based data. The paper demonstrates that building applications on top of the proposed framework is feasible, discusses the benefits and drawbacks of our approach and proposes integration with different sensors for example for tracking soldier's vital signs such as heartbeat or limbs movement.

Krevl, A.; Vidmar, T.; Pancur, M.; Ciglaric, M.; Tomazic, S.; Zavec, A.; Ciglaric, S. (2006) A Framework for Developing Mobile Location Based Applications. In *Dynamic Communications Management* (pp. 1-1 – 1-10). Meeting Proceedings RTO-MP-IST-062, Paper 1. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int/abstracts.asp.

1.0 INTRODUCTION

Today, location based services are very popular, especially when they are location aware. Location based applications are more common then the media hype suggests. We can find them in car and marine navigation systems, in military applications and in tracking software used by large logistics companies like FedEx. Most often, such applications are proprietary software, which means that we cannot use them or integrate them into our own applications without paying hefty license fees. Furthermore, these applications mostly rely on existing (preconfigured) spatial databases that are kept locally on devices where location aware applications are run. While local storage greatly increases application performance, it is inappropriate for applications that need up-to-date spatial data like road construction information or troops' location information. Another problem with spatial databases is that their cost can rise up to thousands of dollars if coverage of a larger area is required. Since maps and applications are developed by the same company it is hard to switch to another product and avoid vendor lock-in. Another drawback of existing applications is their dependence on the Global Positioning System (GPS) to learn their location. Although GPS is the most common system for positioning, the common devices like mobile phones do not have an appropriate receiver. Lastly, the fore mentioned systems rarely include the possibility of inserting new spatial data to the database.

The framework proposed in this paper tries to overcome some of these problems by using open standards, commonly used technologies and COTS hardware. This makes the framework extensible and flexible, enabling it for use in a variety of location aware applications. Due to modular design and standard interfaces it is possible to connect mobile devices with different sensors on one hand and to communicate the acquired data to our servers on the other hand.

2.0 RELATED WORK

Much of the work on location based applications has been done by Intel Research Seattle. Their engineers developed the software called Place Lab [1] which focuses on using radio beacons other than GPS for location discovery. Place Lab utilizes IEEE 802.11 (WiFi) access points, GSM network base stations and Bluetooth devices to determine location. Decoupling location based applications from GPS receivers is an excellent idea, but it relies on pre-defined spatial databases, holding radio beacon locations (GPS coordinates of GSM network base stations, WiFi access points and stationary Bluetooth devices). Such spatial data can be provided by organizations that use WiFi networks on their campus or it can be acquired by war-driving. War-driving is essentially a process of collecting radio beacon locations by driving around cities equipped with different wireless receivers and a GPS device. Every time a radio beacon is discovered, the current location, signal strength and beacon identification number (e.g. WiFi SSID) are logged. These logs are then filtered and inserted into beacon location databases.

German mobile network operator T-mobile was among the first companies to offer a navigation service to its mobile network users [2]. T-mobile provides software and services for road navigation. When a user wants to get from its current location A to the destination B, she enters address B into the application running on her mobile device. The mobile application contacts the appropriate service provided by T-mobile which calculates the optimal route between points A and B. The mobile application then displays driving directions to point B in a fashion similar to one found in car navigation systems. The mobile application requires a Bluetooth GPS receiver to be connected to the mobile device to determine its current position.

Google has recently entered on the location aware application with its Google Earth [9] and Google Maps [8] applications. These applications can access a vast collection of satellite imagery overlaid with map information and can be customized by adding user defined layers (map overlays that contain user defined objects at user defined geo coordinates). There is also a fully customizable Pro version of Google Earth that allows commercial use.

Place Lab's ability to determine location based on radio beacons other than GPS, T-mobile's thin client and server side processing approach and Google's ability to add user defined spatial data to the maps are all great ideas but they each lack exactly the advantages of the other two. So a modern location aware application should comprise of all the ideas spawned by the fore mentioned applications.

3.0 POSITIONING

People determine their location by objects that are in their sight. These can be hills, distinctive buildings in a city or stars on the night sky. Despite the advances made in computer vision research, computers remain unreliable at detecting random objects, so they have to rely on different means for determining their location.

The most commonly used positioning system is GPS. It comprises of 24 satellites orbiting around the Earth and enables us to determine our location anywhere on our planet with an accuracy of roughly 10 meters. To determine their location, GPS receivers need to obtain signals from at least four different satellites. Four satellites define four spheres defined by the difference between send time (from satellite) and reception time (GPS receiver). The intersection of these spheres presents the current location of the GPS receivers.

Mobile devices, especially mobile phones are gaining on their popularity and are very commonly used in our everyday life. Most of these mobile phones are connected to GSM networks. GSM networks are cellular networks by design and every cell in a cellular network has its own base station with a unique base station identifier. Since our mobile phones always know which base station they are connected to, we can use this information to determine the location of our mobile phone. However, because areas covered by a single base station vary in both shape and size it is hard to determine mobile device location accurately. While the distances between base stations in urban regions are between 200 and 500 meters, they can grow to a couple of kilometres in rural regions. Accuracy can be increased by considering time advance and hand over time information, but this can only be obtained from the mobile network operators who usually charge for such services.

Some of the operators are already offering web services that can be used by location aware applications. Unfortunately, there is no common API (Application Programming Interface) for accessing location information on operator's servers, which renders such services useless for application that spread over different operators' networks.

With WiFi networks gaining on their popularity, especially in SoHo environments, they can provide a good means of determining our location, particularly in residential areas and in the vicinity of larger organizations. Similarly to GSM networks, WiFi clients connect to WiFi access points (base stations), which are uniquely identified by their SSID. To determine optimal transfer rates between the client and the base station, client needs to measure the strength and quality of access point's signal. These two parameters enable us to determine our distance from the access point and thus our location. If we receive signals from several access points, we can use triangulation to determine our location with the accuracy of around 15 meters.

Another wireless technology that is quickly gaining on its popularity is Bluetooth. There are many devices that use Bluetooth to communicate, and these radio sources can also help us determine our location. Since Bluetooth was designed for Personal Area Networks (PAN) and is typically short ranged it allows for very precise location discovery. However, Bluetooth sources require some additional filtering before we add them to our spatial databases, because most Bluetooth enabled devices are mobile phones and PDAs which do no have a stationary location.

Positioning based on WiFi, GSM and/or Bluetooth has one important advantage over GPS. Besides determining our location outside, in open areas, it can also help us with determining our location inside buildings. On the other hand GPS performs excellently even in rural areas where other radio beacons are very scarce.

4.0 FRAMEWORK ARCHITECTURE

The framework comprises of three elements (Figure 1): location aware client (our mobile device), the application server (broker) and the database server that holds spatial data. This approach is used to maximize inter-operability between different components, meaning that any component can be replaced by a new one as long as their APIs are compatible e.g. the spatial database running MS SQL could be easily replaced by Oracle or an open source PostgreSQL Database. On the other hand, three-layer architecture enables us to store and process the spatial data on the server side to overcome limitations of mobile devices.

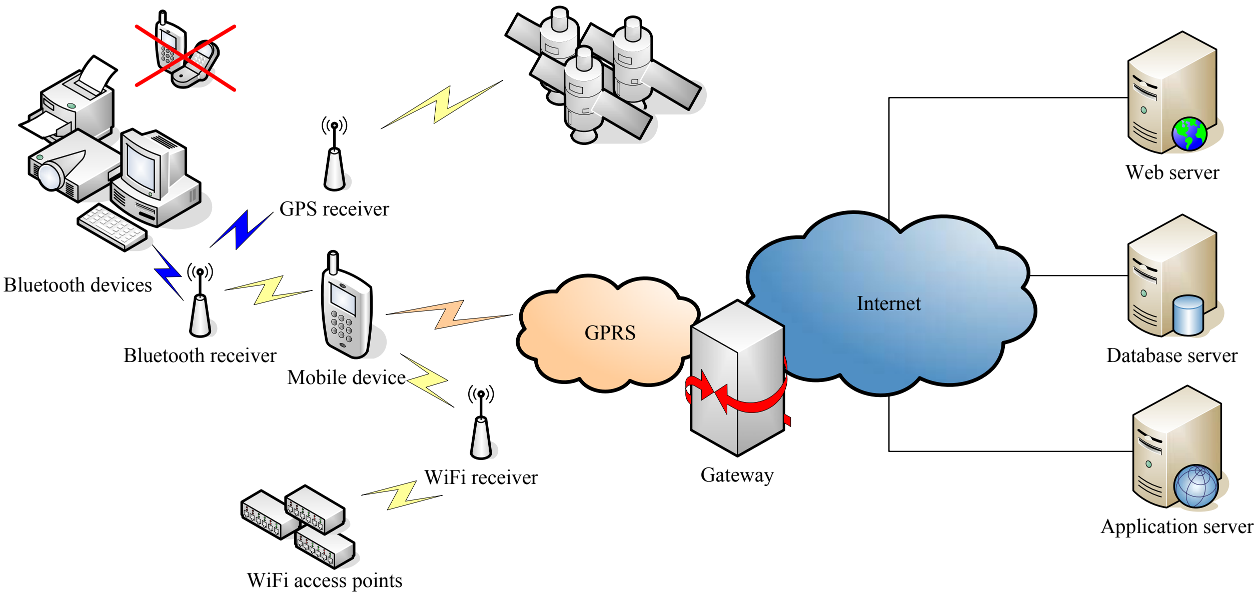

Figure 1: Framework architecture.

Spatial data is stored in a relational database on the database server. There might be performance disadvantages to this (specialized geographical databases are tuned to geographical data and hence provide much greater speeds), but interoperability and the variety of available database management systems currently outweigh developing a special geographical database system. Spatial database contains both the locations of radio beacons with their associated geographical positions and user defined data such as maps, object location information, path information etc. Because of its simple design, the relational database can be easily extended to hold any kind of location data, meta-data or even sensor data.

The application server provides web services for mobile clients and communicates with the database server. Client communication is based on standard SOAP messages while native communication is used when querying the database server. This is against our goals to use only open standards, but it doesn't represent a huge constraint since we are using web services to wrap the proprietary communication. Native database communication can be easily replaced by ODBC or JDBC (dependant on the server platform) still using a native database communication protocol is recommended for best performance.

The mobile device obtains either its geographical coordinates or a list of radio beacons (base stations, access points or Bluetooth devices) in its vicinity from one of its wireless receivers (GPS receiver, WiFi receiver, Bluetooth receiver). If our mobile device obtains a list of radio beacons it first encapsulates these data in a SOAP envelope and sends it to the appropriate web service on the application server. The web service executes a query against the spatial database and returns the approximate location to the mobile client. Now the mobile application can query the web services for interesting information based on its current location. Again the query is encapsulated in a SOAP envelope and sent to the web service, which queries the spatial database server for the required data and returns it to the client. The mobile device displays received location information over a map if one is available from the web service. Location information can contain interesting places, objects, paths etc. in the area around our current location.

Figure 2 shows how our framework interconnects with other components of the system. The framework abstracts methods for access to I/O devices (most commonly different sensors) via serial or Bluetooth interfaces. It also provides methods for obtaining our current location and for obtaining location information. Methods for accessing web services are provided and also used internally for obtaining location information from the application servers. The framework is utilized by a mobile application which does not have to deal with the implementation details of obtaining location information, sensor communication and web services access.

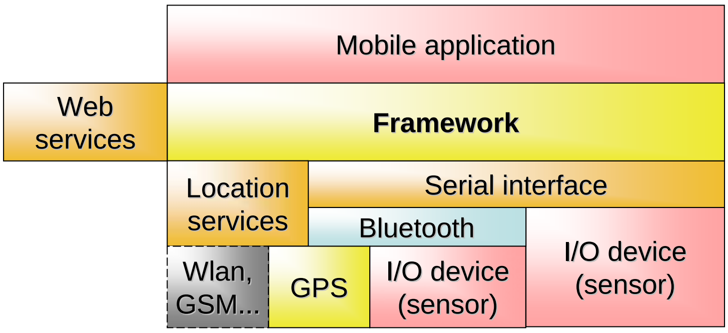

Figure 2: Framework components.

4.1. Usage Scenarios

Figure 3 shows the generic usage scenarios supported by the proposed framework. Location data represent information about interesting places and object in the proximity of our current location. Location data is read from the spatial database on the database server. The basic task of the mobile application is to display location data based on our current location. If a map of the currently displayed area is available, location data are drawn over the map for easier orientation. This functionality can be found in almost all location aware applications.

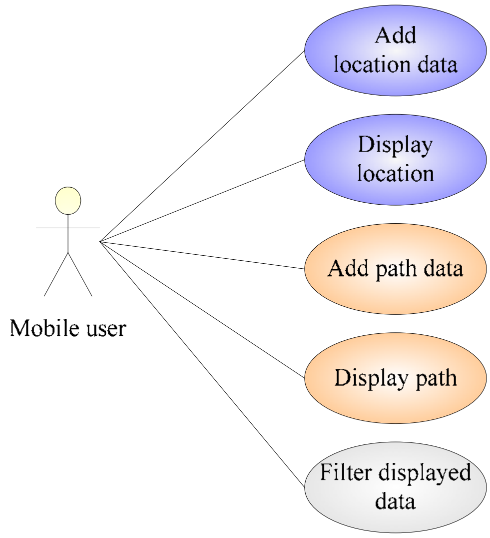

Figure 3: Framework use cases.

A path is defined a set of locations identified by the same meta-data. A path can represent a street, a bicycle tour, a walk around the city centre, directions how to get from point A to point B etc. In addition to displaying location data, the mobile application also needs to be able to show paths that reside in the vicinity of our current location.

In contrast to commonly available location aware applications, our framework adds the option of inserting new (user defined) location data into the spatial database. Open source software proved to be successful, so why not apply the same principle to spatial databases. Say you are just enjoying your meal at a nice restaurant. You could insert the location of this restaurant into the spatial database along with your comments and let others know about the place. Inserting a new location requires only the entry of meta-data associated with the current location.

Paths can be inserted into the spatial database in a similar manner. First the meta-data is entered. Then, the current location is sampled in regular time intervals and stored in the spatial database as a part of the path. Sampling stops on user demand.

Since spatial databases are bound to grow, displaying all locations on the small screen of a mobile device at once would be more confusing than helpful, so data filtering can be applied to location data and path data to show only locations of a certain type (e.g. restaurants) or a certain type of paths (e.g. bicycle tours).

4.2. Proof of Concept

We have developed a prototype navigation application that implements all of the functionality described with use cases (Figure 3), but only uses GPS to determine the current location. Nevertheless, Figure 4 shows that the application is fairly complex, considering it has to run on limited devices. However, our tests have shown that devices like Nokia Series 40 mobile phones have no problem coping with our application.

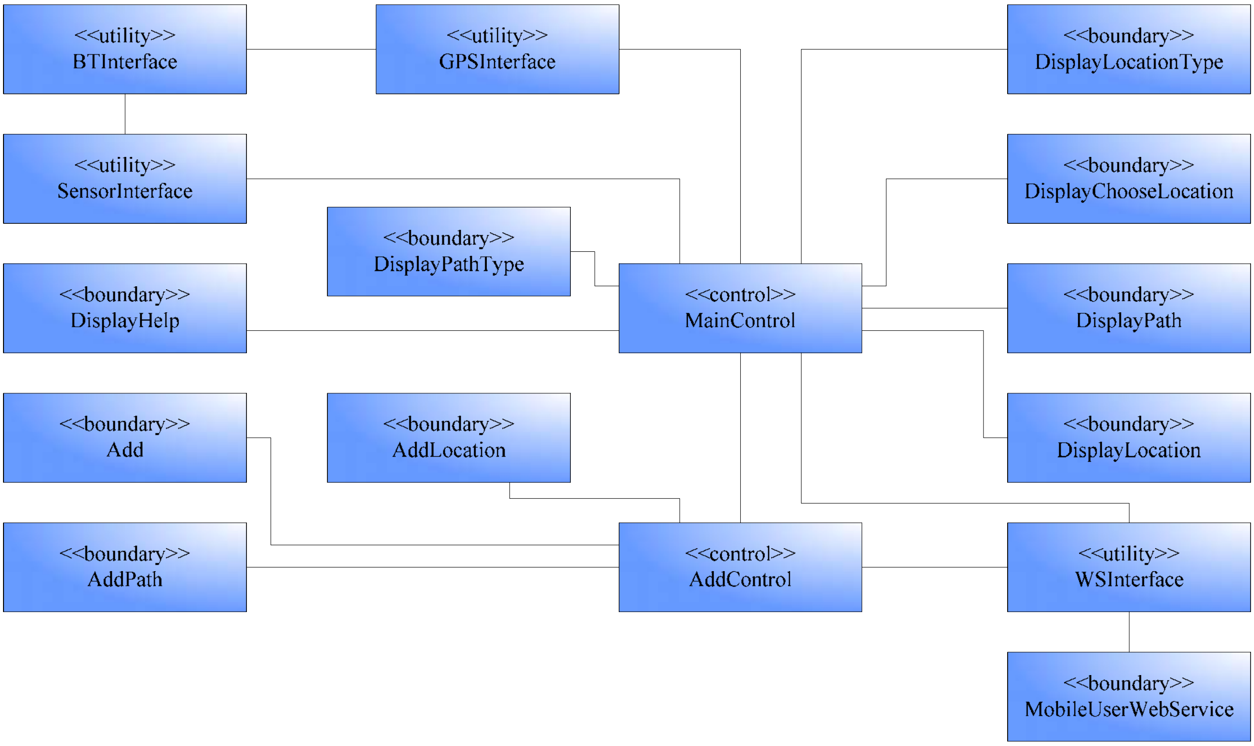

Figure 4: Prototype architecture.

The center of the application is the MainControl class that spawns different Display classes for user interaction. Besides switching between displays it also communicates with the framework: it holds location information obtained from GPSinterface utility class and obtains location information through the web service interface (WSInterface) class.

The AddControl class is in charge of adding location and paths. If we want to add a new location to the spatial database, we need to enter appropriate meta-data (name, description and location type). The AddControl class will obtain current location information from the MainControl class, and then WSInterface will be contacted to send the newly entered location information to the database. Detailed sequence of interactions is shown on Figure 5.

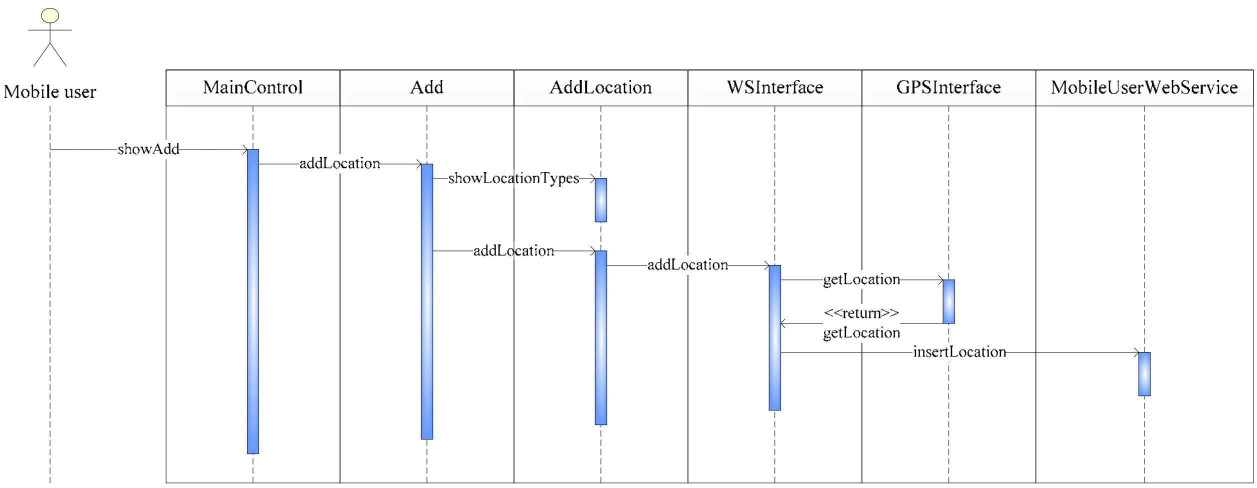

Figure 5: Adding a new location.

Adding a new path is similar to adding a location up to the point of entering meta-data. Afterwards the AddControl class starts sampling current location data from the MainControl class in regular time intervals. Location data is sent to the spatial database via WSInterface class where it is stored among other locations that belong to the same path. Sampling is stopped by the AddControl class on user demand.

Utility class SensorInterface enables us to read from sensors that use standard RS232 interface for communication. Currently it only supports serial over Bluetooth connections, but unfortunately our heart beat sensor is still in the development phase, so we could not test this part of the framework in our prototype. Since we are using modular design and open standards, the relational database can be easily extended with a table that would hold sensor specific data. The same goes for adding sensor related methods to web services and the mobile application.

Figure 6 shows the mail screen of the prototype mobile application. The bigger red dot is displaying our current location and moves over the map as our location changes. The smaller yellow and pink dots represent previously entered location data.

Since this is a prototype the developer still needs some insight into the framework to communicate with the web services and the sensors, however this is going to be simplified in the further versions of the framework.

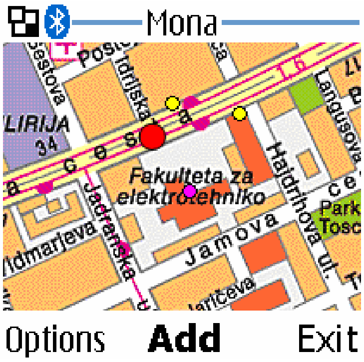

Figure 6: Prototype mobile navigation application

5.0 DISCUSSION

This paper has shown that building location aware distributed applications is feasible by using open standards and technologies. Yes, there is a lot of existing location aware applications, but there are no open, easily extensible, frameworks, that can be used to effortlessly develop location aware applications. By using web services we can integrate any client platform like J2ME or .Net compact framework with any spatial database, be it Microsoft SQL server, Oracle 10g or a third party specialized geographic database. Even web services can be deployed on any of the application servers available today.

Of course, no prototype includes every feature from the ground up. That is why we would like to include Place Lab's ability to determine user's location with the help of radio beacons in the wild. Using WiFi access points, GSM base stations and Bluetooth devices for positioning can bring location aware applications to many new users, who found GPS equipment too expensive in the past. On the other hand, our software can also be easily upgraded to use the up-coming Galileo project for positioning services.

Another interesting addition to the framework is the ability to connect different sensors to our mobile devices. A heartbeat sensor would make a great doctor's accessory, since she could be automatically warned of heartbeat irregularities of her high risk patients. Sensor integration and location awareness could spawn numerous applications that will utilize ubiquitous computing to enhance and simplify our everyday life.

There are still some minor issues with transferring large datasets (raster maps) over relatively slow GPRS networks. This can be solved by using much faster UMTS enabled mobile networks, or alternatively by replacing raster (bitmap) maps with their vectorized counterparts. We would also like to test the feasibility of creating our own vectorized maps from path entries in the spatial database.

6.0 REFERENCES

[1] A. LaMarca, Y. Chawathe, S. Consolvo, et al, Place Lab: Device Positioning Using Radio Beacons in the Wild, Intel Research Seattle, 2003

[2] T-Mobile, http://www.t-mobile.de/, December 2005

[3] J. Schiller, A. Voisard, Location-Based Services, Morgan Kaufmann Publishers, 2004

[4] KREVL, Andrej, CIGLARIČ, Mojca. A framework for developing distributed location based applications. V: IPDPS 2006 : proceedings [of the] 20th International Parallel and Distributed Processing Symposium, April 25-29, 2006, Rhodes Island, Greece. Piscataway: IEEE, 2006

[5] TOMAŽIČ, Sašo, VIDMAR, Tone, CIGLARIČ, Mojca, PANČUR, Matjaž, KREVL, Andrej, ZAVEC, Aleš, CIGLARIČ, Stane, KOS, Anton. *Informacijsko komunikacijski sistem NCO-NCW-NNEC : projekt "Viking"*. Ljubljana: Univerza v Ljubljani, 2005

[6] TOMAŽIČ, Sašo, VIDMAR, Tone, CIGLARIČ, Mojca, PANČUR, Matjaž, KREVL, Andrej, ZAVEC, Aleš, KOS, Anton. Lokacijske storitve v mobilnih sistemih. Ljubljana: [Univerza v Ljubljani], 2005

[7] B.N.Schilit, A. LaMarca, D. McDonald, J. Tabert, E. Cadag, G. Borrielo, W.G. Griswold, J. Hong, Bootstraping the Location-enhanced World Wide Web, Intel Research Seattle, 2005

[8] Google Maps, http://maps.google.com/, September 2006

[9] Google Earth, http://earth.google.com/, September 2006