UbiLoc: A System for Locating Mobile Devices using Mobile Devices

Felix ALCALA[1], Jöran BEEL[2], Arne FRENKEL[3], Bela GIPP[4], Hagen HÖPFNER[5] and Johannes LÜLF[6]

Otto-von-Guericke-University of Magdeburg, Department of Computer Science
Institute of Technical and Business Information Systems
P.O. Box 4120, D-39016 Magdeburg, Germany

[1]mail@felixalcala.de, [2]joeran@beel.org, [3]arne@frenkel.org, [4]bela@gipp.net,
[5]hoepfner@iti.cs.uni-magdeburg.de, [6]mail@johannesluelf.com

***Abstract* - Considering the current trends of the development of mobile devices (e.g. mobile phones or embedded hardware) we note that these devices become smaller but provide more functions. Today, it is not unusual to combine a GPS controller with laptops or personal digital assistants. Even mobile phones with integrated GPS chips are available. Therefore, we can use the exact coordinates of mobile devices in order to realize novel applications. In this paper we present a system that is able to localize mobile devices from other mobile devices as well as from a standard desktop computer.**

1 Introduction and motivation

Location based services (LBS) are one of the most interesting application areas in the sector of mobile information systems. The central characteristic of the hardware used in this context is its compactness and capability. Even small mobile devices become more and more powerful. Therefore, LBS are no longer restricted to mobile computers like laptops or personal digital assistants (PDAs), but can also be implemented with mobile phones.

The combination of wireless networking and the possibility to locate user devices enables IT providers to offer services that are customized to the client's current location. Generally, the localization can be done in two different ways [1]. Firstly, one can use information provided by the mobile network the device is connected to. Since wireless network technologies (GSM [2], GPRS [3], HSCSD [4]) are based on a cellular infrastructure, the position of a user can be roughly estimated by using information of the nearest base station. A more precise method computes information of several nearby stations (so-called multi cell triangulation). Secondly, a device can be located using the global positioning system (GPS) [5], which is far more precise. Both methods have advantages and disadvantages that are not deepened further in this paper.

The project being presented in this paper was designed as a basis to enhance another system called "GSM-Schutzengel" [6]. This award-winning system automatically notifies the emergency medical services in case of a car crash. The notice includes the user's current position, which ensures the fastest possible help. The original approach uses multi cell triangulation to locate the devices. To achieve a more precise localization, it was assayed whether and how GPS is suited to assist this and other location-based information systems.

Unfortunately, a GPS capable mobile device has not been available to us. As we wanted to evaluate our theoretical ideas in practice, we decided to implement an alternative GPS-based system in which a laptop connected to an external GPS controller simulates the mobile device to be located.

2 Overall architecture

The main goal of the project was the localization of mobile devices through mobile devices. The device to be localized is GPS capable and has access to the internet. Its position is permanently being

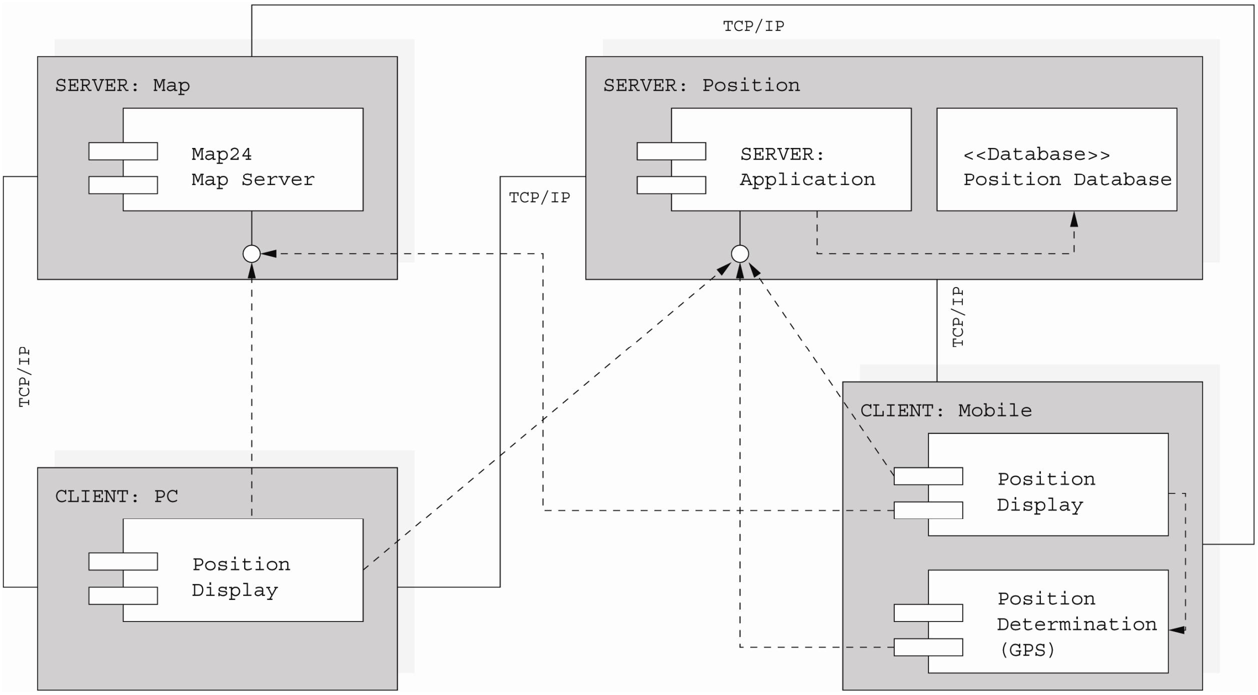

Figure 1: System

determined and updated in a central data base. To query positions from the database, we implemented a desktop PC application and an application for Java-enabled mobile phones. Both applications can show the position of the localized object on a map. In order to prevent misuse of position data, the system contains mechanisms to grant rights to selected users: It can be specified which mobile devices a user is allowed to locate. In the following we will present the overall architecture of the system.

2.1 Involved components

The system (figure 1) consists of four nodes, two servers and two clients, where the clients may appear in more than one instance. The functionality of the clients is divided into the components *position display* (locating user) and *position determination* (located user). The latter is connected to a GPS controller and transmits the current position of the device to the server. In order to reduce the amount of data to be sent, the user can freely set transmission intervals and define a minimum change of position.

All transmitted positions are inserted into a database by the *position server*. Besides this, the server saves advanced information like insertion date and time or the velocity of the user. The component *position display* can now – assuming that the necessary rights have been granted – provide information by querying the server. The map material that is needed to display the location is downloaded from the online mapping service Map24 [7], which delivers a static picture of any map section within Europe (through an appropriately encoded URL). The sequence diagram in figure 2 illustrates the data flow during a position request. All nodes are implemented in Java.

2.2. Hardware Requirements

Our mobile software is implemented in Java 2 Micro Edition (J2ME), so the mobile devices have to support Java's Connected Limited Device Configuration Specification (CLDC) [8]. Furthermore, they must be able to establish an internet connection. The device that will be located additionally has to fulfil the following requirements:

- *A permanent connection to the internet:* To ensure a certain degree of significance, the position of the users needs frequent updates. Therefore, a permanent internet connection is suggestive, for example through GPRS. Being a packed based service, the costs of a GPRS connection are usually

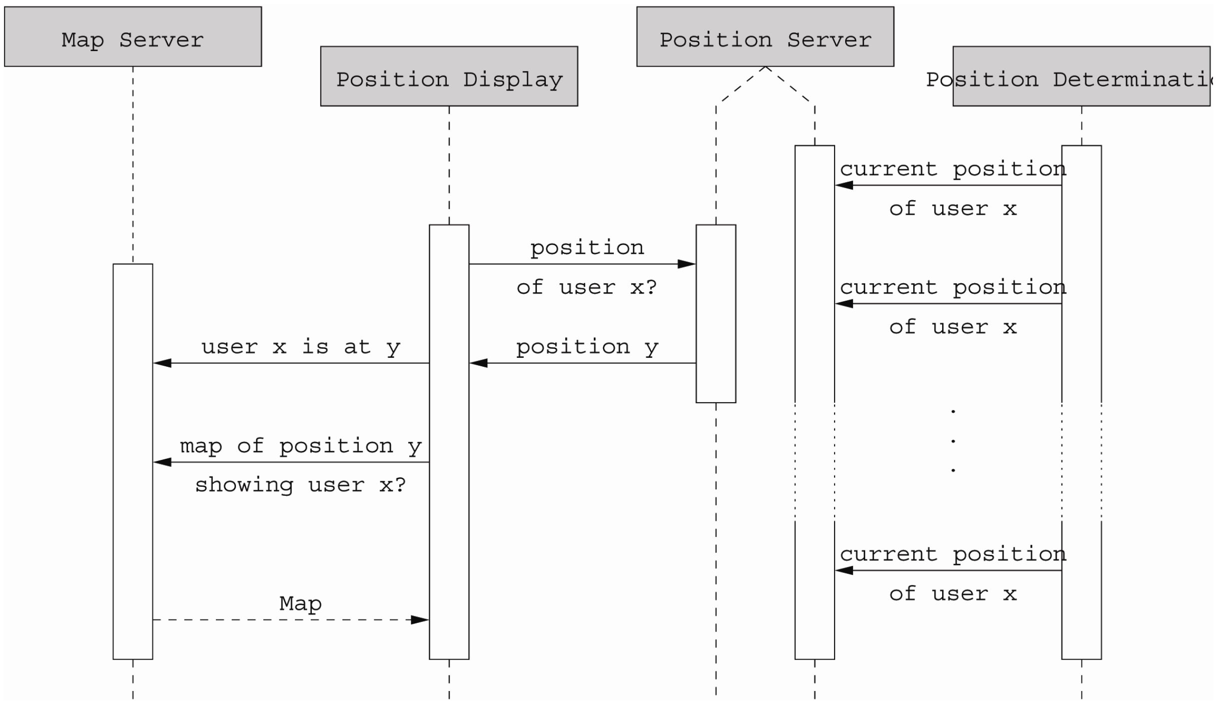

Figure 2: Sequence Diagram for Requesting a Position

determined by the network traffic and not by the duration of the connection. Since a position information packet is a string of less than 400 characters, the resulting traffic is low.

- *GPS capability:* The device to be located must have access to an integrated or external GPS receiver.
- *Java support:* A Java program determines the position and transfers this information to the server. The device must be multi-tasking capable, so that it is not blocked by the execution of this program.

3 Functionality provided by the server

The devices that request locations and the devices that provide locations are connected to a central server. The server administrates position data and user information as well as localization rights in a relational database and ensures the consistency, persistency and correctness of this data. For example, location information has to be converted since the user sends such information in a format that differs from the format being compliant with the maps. Due to the limitations of J2ME, this conversion cannot reasonably be done on the mobile device. Besides this, J2ME has no support for sending instances of objects over a network. Therefore, we had to create a string based protocol to replace Java's powerful *serialization* feature. Our protocol has the following syntax:

[action]×[condition]×[user]×[dataID 1]×[dataRecord 1]×···×[data ID *i*]×[dataRecord *i*]

× is a (predefined) separator. For example, the string

`getGdp×latest×ownUserData×UsernameToLocalize`

returns the last position of a user. The own user data must be supplied so that the server can check if the user is authorized to localize the selected device. The insertion of new position information is done as follows:

`insertGdp××ownUserData×longitude×12.65×latitude×48.15×...`

Here, a `condition` is not necessary (represented by two consecutive separators).

In our prototype, network communication is not encrypted. Once the system is used as a productive system, encryption of personal data and position information is mandatory. First tests show that depending on the Java version, problems can arise on mobile devices due to J2ME limitations. On the server side we do not expect difficulties.

As mentioned before, users are not allowed to locate every other user of the service. This problem is solved in a pragmatic way: during each communication process, the username and password are transmitted. If authentication fails, the server cancels the current operation and sends an error message to the requesting user.

4. Functionality provided by the client applications

As mentioned in the beginning, a user's location can be retrieved from a personal computer as well as from a mobile device. In the following, both client applications are presented.

4.1 Mobile phone client

The prototype uses mobile phones to represent mobile devices. In principle, the application can easily be exported to other mobile devices such as PDAs or Smartphones, as long as the devices fulfil the requirements discussed in section 2.2.

Programming Language: The most important requirement was compatibility. The software had to be executable on all popular mobile phones and PDAs without major modifications. Therefore, proprietary languages such as In-Fusio's ExEn[1] did not fit our needs. Today, most mobile devices that come into the market are equipped with a "Java Virtual Machine", which challenged us to implement our software in Java. The Java specification is divided into several variants that cover different hardware capabilities. Besides the prevalent regular edition (J2SE/J2EE), Sun offers the Java 2 Micro Edition (J2ME) which is optimized for usage on mobile devices. This standard is further split up into several subgroups. Mobile phones use the Mobile Information Device Profile (MIDP), which is based on the Kilobyte Virtual Machine (KVM). It only needs 32 to 512 kb main memory. Amongst other features, MIDP provides network support (HTTP 1.1), the possibility to permanently store information, and a graphical user interface that automatically adapts to the device's characteristics.
Unfortunately, the low hardware requirements of MIDP bring about severe technical restrictions. For example, most devices using MIDP do not support floating point operations.

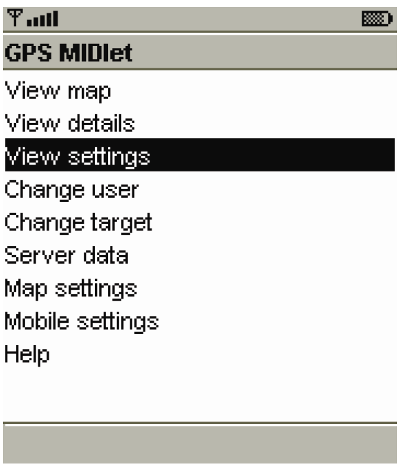

Figure 3a: Mobile Phone Client;
Main Menu

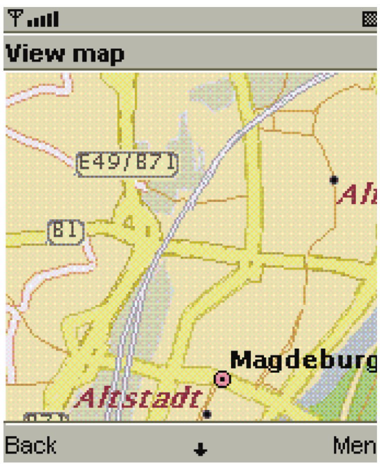

Figure 3b: Mobile Phone Client;
Displaying a Map

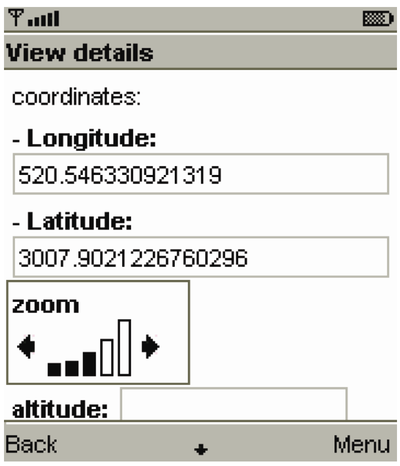

Figure 3c: Mobile Phone Client;
Displaying detailed information

1 http://www.in-fusio.com/fond_game/technologie_games.php

Installation: Compared to the installation routines known from personal computers, installing software on a mobile phone is a little more difficult. The software can either be copied from a PDA or PC, or it is installed using “over-the-air provisioning”. In the latter case, the J2ME compatible application (the so-called MIDlet) is downloaded from the internet. At first, only a `.jad` file that describes the application and its size is downloaded. The user can then decide whether or not to download the entire application as a compressed `.jar` file.

User interface: After starting the application the main menu (see figure 3a) is shown. With *View Map* the own position or the position of another user can be shown on a map (see figure 3b). If required, a reticule which points out the exact position can be plotted onto the map. The menu item *View details* displays additional information like longitude, altitude, speed or the number of available satellites (see figure 3c). *View settings* contains additional settings such as the update interval or the minimal change of position. As already mentioned, these parameters are useful to reduce transmission costs. The user management can be found at *Change User*. Besides the settings regarding the current user (username and password) it also includes the permission settings. *Change target* requests information (location and details) of other users. Settings regarding the connection to the server can be changed at *Server data*. Likewise, *Map settings* contains settings regarding the map server. *Mobile settings* allows to adapt the software to the mobile device by changing the display resolution and the number of colours. However, these parameters are usually determined automatically.

4.2 Desktop Client

Since mobile devices have a limited display size, not all details of the GPS data can be displayed in a reasonable way. We decided to implement a desktop client to solve this problem.

Programming Language: The desktop client is written in Java (J2SE 1.4) to guarantee platform independence. Every computer that can run a “Java Virtual Machine” can also run our application. As for the user interface, we use the Swing package which is rather easy and intuitive to use.

Functionality: The desktop client (see figure 4) can plot the position of mobile terminals on a map. A connection to the internet is required to run the software since both the users’ positions and the map are stored online. Position information is retrieved using the string-based protocol presented in chapter 3. The required map sections are returned as a GIF image by the map server on a simple

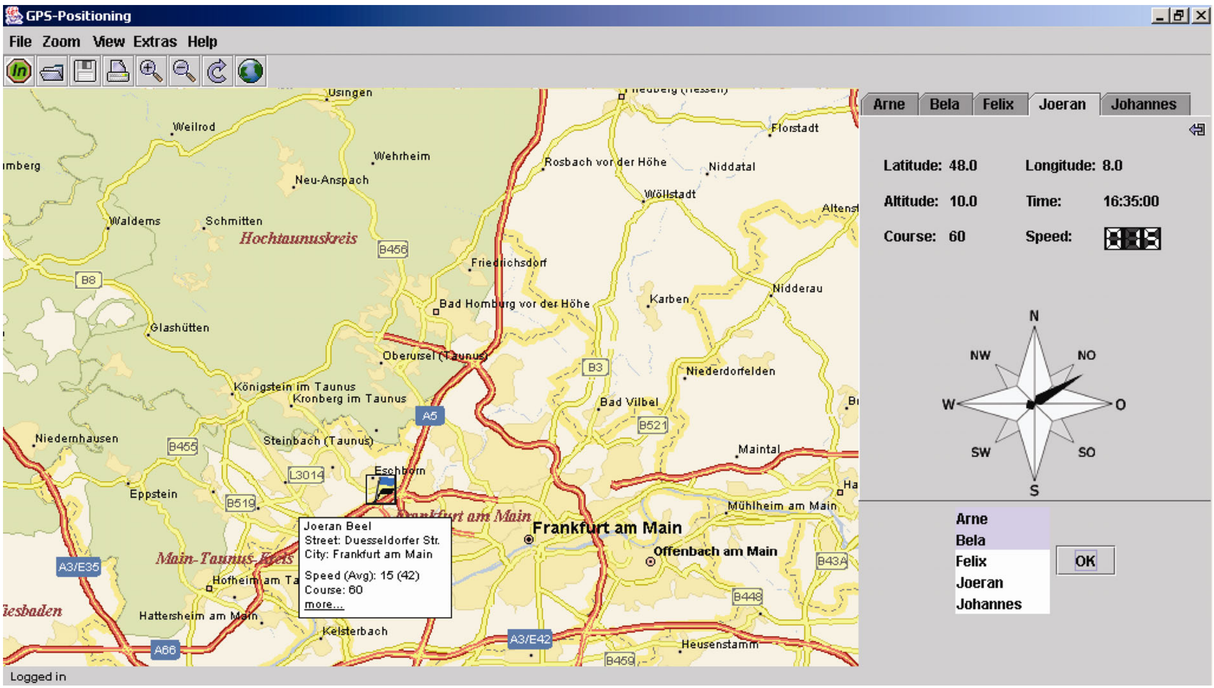

Figure 4: Desktop Client

HTTP/URL request that contains longitude and latitude. The map server also plots the positions of the selected users onto the image. This ensures optimal readability of the captions. As a consequence, the users' positions have to be submitted to the server in a separate HTTP-request.

After starting the application the users have to authenticate (username and password) themselves. If authentication is successful, the position server transmits a list of all users that have granted localization rights to the user that logged in. The user can now select one or more of them; their positions are displayed on a map that automatically zooms to show all users.
A tab shows additional statistical information such as altitude or speed for each user. The direction of the user's movement is also displayed in a compass rose whose needle grows in correlation to the speed. The map can be zoomed either stepwise or by using predefined levels of detail (street, city, region, country). Another feature of the desktop client is the possibility to playback routes. On the position server, each position dataset is assigned to exactly one route. As the dataset also contains a timestamp, a reconstruction of the route is possible. On replay, the user can choose between real time and accelerated playback.

5 Summary and prospects

A mandatory condition for the realization of location based services is locating mobile devices. Within the scope of the presented project we developed a system which enables to determine the position of a device equipped with GPS. In addition to a server that manages user rights and position information, two client applications were implemented. They allow accessing the data from a mobile phones and a desktop PC, respectively. The overall architecture was presented in this paper as well as aspects of the implementation and the usage of the system.
In fact, the system is a prototype and leaves space for further development, such as (1) scalability of the server in case of large amounts of users and (2) encrypted communication between the participating components. Alongside technical questions, possible application fields were a central question. As already mentioned the main goal was to increase accuracy for localization of the "GSM-Schutzengel" through GPS.

Acknowledgement: Our gratitude goes to Infineon Technology AG, NETSOLUT GmbH, as well as SOS r.AG, which supported us through financial and material capital. We also want to state that the research of Hagen Höpfner is supported by the DFG under grant SA 782/3-2.

References

[1] A. Jagoe, *Mobile Location Services: The Denitive Guide*. Prentice Hall PTR, Dec. 2002.
[2] ETSI, "European digital cellular telecommunications system; Attachment requirements for Global System for Mobile communications (GSM) mobile stations," European Telecomunications Standards Institute, Sophia Antipolis Cedex, Technical Basis for Regulation, Nov. 1993.
[3] —, "Digital cellular telecommunications system (Phase 2+) (GSM); General Packet Radio Service (GPRS); Service description; Stage 1 (GSM 02.60 version 6.1.1)," European Telecomunications Standards Institute, Sophia Antipolis Cedex, European Standard (Communication series), Nov. 1998.
[4] —, "Digital cellular telecommunications system (Phase 2+); Attachment requirements for Global System for Mobile communications (GSM); High Speed Circuit Switched Data (HSCSD) multislot mobile stations; Access (GSM 13.34 version 5.0.3)," European Telecomunications Standards Institute, Sophia Antipolis Cedex, European Standard (Communication series), Mar. 1999.
[5] E. Kaplan, *Understanding GPS: Principles and Applications*, ser. Artech House Telecommunications Library. Artech House, Mar.1996.
[6] B. Gipp, J. Beel, F. Alcala, and L. Petersen, "Der GSM-Schutzengel..., web page" 2003, `http://www.gsm-schutzengel.de`.
[7] NETSOLUT GmbH, "Map24 - online route planner and interactive street maps for UK and Europe, web page," 2003, `http://www.map24.de/index.php?language=en`.
[8] Sun Microsystems, Inc., "J2ME - Java 2 Platform, Micro Edition, web page", 2003, `http://java.sun.com/j2me/`.