

# Ortung von mobilen Geräten für die Realisierung lokationsbasierter Diensten

Felix Alcalá  
*mail@felixalcala.de*

Jöran Beel  
*joeran@beel.org*

Arne Frenkel  
*mail@arnefrenkel.de*

Béla Gipp  
*bela@gipp.net*

Johannes Lülff  
*mail@johanneslueff.com*

Hagen Höpfner\*  
*hoepfner@iti.cs.uni-magdeburg.de*

Otto-von-Guericke-Universität Magdeburg  
Postfach 4120, 39016 Magdeburg, Deutschland

**Abstract:** Die Entwicklung im Sektor von mobilen Endgeräten geht eindeutig in die Richtung, immer kompaktere Geräte mit immer mehr Funktionalität anzubieten. Schon heute können z.B. Laptops oder sogar PalmTops mit GPS-Kontrollern kombiniert werden. Die somit verfügbaren genauen Positionsinformationen ermöglichen die Realisierung von unterschiedlichsten lokationsbasierten Diensten. Ziel des in diesem Papier vorgestellten Projektes ist es, die Möglichkeiten derartiger Szenarien zu untersuchen. Dazu wird ein System vorgestellt, welche die ad-hoc Lokalisierung von Nutzern von mobilen Endgeräten mittels mobiler Endgeräte ermöglicht.

## 1 Einleitung und Motivation

Lokationsbasierte Dienste sind einer der aktuellsten und spannendsten Schwerpunkte im Bereich mobiler Informationssysteme. Diese Tatsache resultiert nicht zuletzt daraus, dass eine zentrale Eigenschaft der hierbei eingesetzten Endgeräte deren Kompaktheit und Leistungsfähigkeit ist. Nutzer mobiler Endgeräte wie Laptops, PDAs oder Handys können sich unabhängig von Büroräumen und Schreibtischen bewegen und dennoch die Funktionen ihrer Geräte ausschöpfen. In Kombination mit der kabellosen Vernetzung und der Lokalisierung derartiger Geräte ergibt sich aus Sicht der Anbieter von Informationssystemen daraus die Möglichkeit, dem mobilen Kunden Dienste anzubieten, welche auf dessen aktuellen Aufenthaltsort zugeschnitten sind. Die dazu notwendige Bestimmung des Ortes kann im Wesentlichen auf zwei verschiedene Arten [Eri01] erfolgen. Da die am weitesten verbreiteten und flächendeckend verfügbaren Funknetztechnologien (GSM [ETS93], GPRS [ETS98], HSCSD [ETS99]) einen zellulären Charakter aufweisen, kann zum einen über die zur Einwahl notwendigen Basisstationen mittels Triangulation die ungefähre Position eines Gerätes ermittelt werden. Zum anderen besteht die wesentlich genauere Möglichkeit der Positionsbestimmung unter Verwendung des GPS [Kap96]. Beide Verfahren haben Vor- und Nachteile, welche hier aber nicht tiefergehend erläutert werden sollen.

---

\*Gefördert durch die DFG unter der Fördernummer SA 782/3-2.

Das in dem Beitrag beschriebene Projekt wurde als Grundlage für die Weiterentwicklung des beim „Jugend-forscht“-Wettbewerb 2002 präsentierten und prämierten Systems „GSM-Schutzengel“ [GBAP03] konzipiert. Dieses System dient im Wesentlichen der automatischen Meldung der aktuellen Position eines Kraftfahrers via Handy im Falle eines Unfalls an eine Rettungsstelle. Auf diese Weise kann dem Verunfallten schneller geholfen werden. Der ursprüngliche Ansatz nutzt die Zell-Triangulation zur Positionsbestimmung. Um eine noch genauere Lokalisierung zu ermöglichen wurde im Rahmen der hier vorgestellten Arbeit untersucht, ob und wie sich das GPS eignet, um derartige lokationsbasierte Informationssysteme zu unterstützen.

Die Aufgabenstellung wurde im Rahmen eines Softwarepraktikums am Institut für Technische und Betriebliche Informationssysteme der Fakultät für Informatik an der Otto-von-Guericke Universität Magdeburg bearbeitet. Da die vollständige Integration der GPS-Positionsbestimmung den Rahmen dieser Veranstaltung gesprengt hätte und der positive Abschluss eines Softwarepraktikums ein lauffähiges System voraussetzt, wurde das in Abschnitt 2 vorgestellte Ersatzszenario implementiert. Die hierbei verwendeten Konzepte lassen sich aber - bei vorhandener Hardware - mit etwas zusätzlichem Aufwand in den GSM-Schutzengel integrieren.

## **2 Gesamtarchitektur**

Grob gesagt, war das Ziel des Projektes die Ortung von mobilen Geräten durch mobile Geräte. Bei dem zu ortenden Gerät handelt es sich um ein GPS-fähiges Endgerät, welches über eine Internet-Anbindung verfügt. Es ermittelt permanent seine Position und aktualisiert sie in einer zentralen Datenbank. Für die Abfrage dieser Position aus der Datenbank existieren eine Standard-Desktopanwendung und eine Applikation für Java-fähige Handys. Beide Applikationen ermöglichen die Darstellung der Position des zu ortenden mobilen Gerätes anhand einer Karte. Dazu werden die in der Datenbank gespeicherten Informationen gelesen und in das Kartenmaterial eingeblendet. Neben der Darstellung der Position, welche in unterschiedlichen Detailstufen möglich ist, können auch Informationen zur aktuellen Bewegungsrichtung und Geschwindigkeit angezeigt werden. Um den Missbrauch der verfügbaren Daten zu verhindern, enthält das System auch Mechanismen zur Rechtevergabe. Auf diese Weise kann festgelegt werden, welcher Nutzer welche mobilen Geräte orten darf. Im Folgenden wird die Grob-Architektur des Systems vorgestellt.

### **2.1 Beteiligte Komponenten**

Das System (siehe Abbildung 1(a)) besteht aus vier Knoten, zwei Servern und zwei Clients, wobei die Clients in mehreren Instanzen vorliegen.

Die Funktionalität der Clients unterteilt sich in die Komponenten Positionsanzeige (ortender Nutzer) und Positionsermittlung (georteter Nutzer). Letztere ist mit einem GPS-Controller verbunden und sendet die aktuelle Position des Gerätes an den Positionsserver. Die Übermittlung geschieht über TCP/IP, was beliebige Anwendungsprotokolle ermöglicht. Ideal ist eine paketbasierte Technik wie GPRS. Um die übertragene Datenmenge möglichst

gering zu halten, kann man die Übermittlung in frei definierbaren Zeitintervallen erfolgen lassen sowie Mindestpositionsveränderungen definieren. Der Positionsserver trägt alle übermittelten Positionen in einer Datenbank ein. Gespeichert werden dabei neben der Position auch weitergehende Informationen wie Geschwindigkeit, Datum und Zeit. Die Komponente Positionsanzeige kann dann - entsprechende Rechte vorausgesetzt - diese Informationen für beliebige Nutzer vom Server abfragen. Das zur Anzeige der Position benötigte Kartenmaterial wird vom Kartendienst Map24 [NET03] geladen. Dieser ermöglicht das Anfragen beliebiger Kartenausschnitte innerhalb von Europa über eine entsprechend kodierte URL. Die Karte wird dabei als statisches Bild geliefert. Das Sequenzdiagramm in Abbildung 1(b) verdeutlicht den Ablauf einer Positionsabfrage.

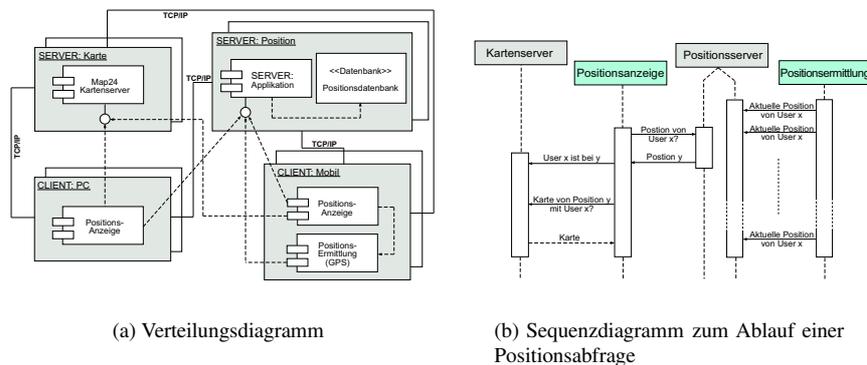


Abbildung 1: Gesamtarchitektur

Alle Knoten (abgesehen vom Map24-Kartenserver) wurden in Java implementiert. Da kein mobiles Gerät mit GPS-Funktionalität zur Verfügung stand, wurde für die Positionsermittlung auf einen Laptop mit angeschlossenem externem GPS-Controller zurückgegriffen.

## 2.2 Anforderungen an die Endgeräte

Das ortende mobile Endgerät muss Java nach der Connected-Limited-Device-Configuration-Spezifikation (CLDC) [Sun03] unterstützen und eine Internet-Verbindung aufbauen können. Für das ortende nicht-mobile Endgerät gelten die gleichen Voraussetzungen. Jedoch wird hier anstatt der eingeschränkten J2ME (Java 2 Micro Edition) die Java Virtual Machine 1.4 oder höher benötigt, welche einen stark erweiterten Funktionsumfang liefert.

An das zu ortende mobile Endgerät werden die folgenden Anforderungen gestellt:

**Möglichst permanente Anbindung an das Internet:** Damit die Positionen des Nutzers die notwendige Aussagekraft besitzen, müssen sie in möglichst kurzen Abständen an den Server übermittelt werden. Hierfür ist eine dauerhafte Internet-Anbindung des Endgerätes, beispielsweise über GPRS, sinnvoll. Bei GPRS wird nicht nach Zeit sondern nach übertragendem Volumen abgerechnet. Da die Positionsdaten nur wenig Übertragungskapazität benötigen, sind die anfallenden Kosten relativ gering.

**Integration von GPS im Endgerät:** Damit das Gerät seine Position ermitteln kann, muss ein GPS-Empfänger integriert sein bzw. Zugriff auf ein externes GPS-Gerät bestehen.

**Fähigkeit zum Ausführen von Java:** Die Übermittlung der Position an den Server erfolgt über ein Java-Programm. Damit das Gerät dadurch nicht blockiert wird, muss es multitasking-fähig sein.

Derzeit erfüllen keine Geräte diese Anforderungen vollständig. So hatten die Handelsketten Aldi und Real im Sommer 2003 zwar GPS-fähige PDAs im Angebot, diese benötigten jedoch zusätzlich ein Mobiltelefon, um eine Verbindung zum Internet herzustellen. Andererseits gibt es von Benefon<sup>1</sup> oder Garmin<sup>2</sup> Mobiltelefone mit integriertem GPS-Empfänger, diese Geräte sind aber nicht java-fähig. Es ist jedoch zu erwarten, dass diese Anforderungen bald von einer zunehmenden Anzahl von Geräten erfüllt werden. So bietet Global Locate bereits heute einen extrem kleinen und stromsparenden GPS-Chipsatz<sup>3</sup> für Mobiltelefone an, so dass es nur noch eine Frage der Zeit (und des Preises) ist, bis dieser in Mobiltelefone integriert wird.

### 3 Vom Server bereitgestellte Funktionalität

Als zentrales Bindeglied zwischen positionssendendem und positionsabfragendem Benutzer fungiert ein in Java implementierter Server. Dieser verwaltet Positions- und Benutzerdaten sowie die Zugriffsrechte in einer relationalen Datenbank und gewährleistet die Konsistenz dieser Daten. So ist beispielsweise eine Umrechnung der Positionsdaten notwendig, da der positionssendende Benutzer die Daten des GPS-Gerätes in einem anderen Format ausliest als sie später für das Kartenmaterial benötigt werden. Aufgrund der Beschränkungen der J2ME ist eine Umrechnung direkt nach dem Auslesen auf dem mobilen Endgerät nicht sinnvoll möglich.

Daneben ist es mangels *Serializable-Implementation* seitens der J2ME nicht möglich ein GPS-DataPiece<sup>4</sup> als Objekt zu übertragen. Daher werden die GPS-DataPieces bzw. Positionsdaten vom Client in einen String konvertiert, welcher über eine Socket-Verbindung an den Server geschickt wird. Dieser trägt die Daten nach erfolgreicher Benutzeridentifikation in die Datenbank ein. Die Stringgenerierung für die Kommunikation folgt einer logischen und leicht erweiterbaren Syntax:

[action] × [condition] × [user] × [dataID 1] × [dataRecord 1] × ... × [dataID *i*] × [dataRecord *i*]

wobei × als (vorher festgelegtes) Trennzeichen fungiert. Beispielsweise liefert

```
getGdp × latest × ownUserData × UsernameToLocalize
```

die letzte Position eines Benutzers zurück. Damit der Server prüfen kann ob die Berechtigung zur Lokalisierung vorliegt, müssen die eigenen Nutzerdaten mit übertragen werden.

<sup>1</sup>Modelle: ESC!, Track Pro NT, Track One NT; <http://www.benefon.com/products/>

<sup>2</sup><http://www.garmin.com/>

<sup>3</sup>[http://www.globallocate.com/chip\\_set\\_main.shtml](http://www.globallocate.com/chip_set_main.shtml)

<sup>4</sup>Datenstruktur, in der die GPS-Daten gespeichert werden

Das Einfügen von GPS-DataPieces für den Benutzer „ownUserData“ kann beispielsweise wie folgt realisiert werden:

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

Auf eine `condition` kann dabei verzichtet werden, was durch die beiden direkt aufeinander folgenden Trennzeichen dargestellt wird.

Momentan erfolgt die Kommunikation im Testsystem unverschlüsselt. Für den Einsatz in einem Produktivsystem ist eine Verschlüsselung insbesondere der personen- und positionsbezogenen Daten zwingend gefordert. Je nach verwendeter Java Virtual Machine kann es diesbezüglich aber auf den mobilen Geräten zu Problemen kommen. Serverseitig ist eine Verschlüsselung der übertragenen Daten hingegen problemlos möglich.

### 3.1 Datenbank

Der Zugriff auf die Datenbank erfolgt ausschließlich über den Server (vgl. Abbildung 2). Zum einen ist es systembedingt (Verwendung von J2ME ohne plattformabhängige Erweiterungen) technisch nicht möglich, direkt von einem mobilen Gerät auf Datenbanken zuzugreifen. Andererseits ist ein direkter clientseitiger Zugriff aus Gründen der Datenfluss- und Zugriffskontrolle nicht sinnvoll.

Der Server greift über die Standardschnittstelle JDBC auf die Datenbank zu, die Teil der Standard-API von J2SE (Java 2 Standard Edition) ist. Die Abfrage der Datenbank erfolgt somit über Standard-SQL-Anweisungen.

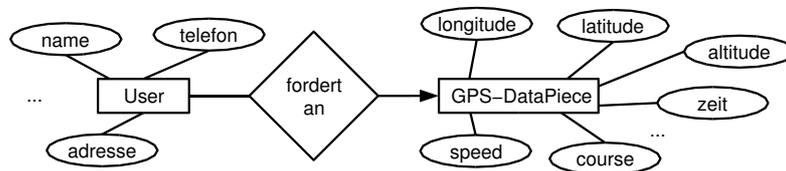


Abbildung 2: Ausschnitt aus dem ER-Schema: Anfordern von Positionsdaten

Die Datenbank besteht im Kern aus 3 Relationen. Zum einen werden positionsbezogene Informationen gespeichert, welche später vom Server und Client als GPS-DataPiece-Objekte verwendet werden. Entsprechende Daten sind die genaue Positionsangabe in Longitude, Altitude und Latitude, der Zeitpunkt, die Geschwindigkeit, der zur Position gehörende Benutzername, der Kurs und zudem der Zeitpunkt, zu welchem die Daten in die Datenbank eingefügt wurden. Letzteres kann genutzt werden, um Zeitunterschiede zwischen Auslesen der Daten aus dem GPS-Gerät und dem Einfügen in die Datenbank festzustellen. Als Primärschlüssel wird der eindeutige Benutzername und die Zeit des GPS-DataPieces verwendet. Dies ist insofern sinnvoll, als dass ein und der selbe Benutzer kaum Positionsdaten zu einer identischen Zeit liefern wird - und wenn doch, ganz offensichtlich ein Fehler im System vorliegt. Zudem werden in der Datenbank sämtliche Nutzerdaten gespeichert,

wie Benutzername, Vor- und Nachname, Anschrift und natürlich die Mobilfunknummer. Eine letzte Tabelle enthält die Zugriffsrechte der einzelnen Benutzer, wobei aus dieser Tabelle auch die „Buddylist“ generiert wird, welche dem Benutzer zeigt, wen er lokalisieren darf.

### 3.2 Zugriffskontrolle und Benutzermanagement

Allgemeines ist davon auszugehen, dass nicht alle Nutzer eines solchen Services jeden beliebigen Nutzer lokalisieren dürfen. Damit nur berechtigte Benutzer Positionen abfragen oder eintragen können, werden bei jedem Kommunikationsvorgang zwischen Client und Server der eigene Benutzername und das Passwort übermittelt. Der Server überprüft stets, ob der jeweilige Benutzer auch die entsprechenden Rechte besitzt. Ist dies nicht der Fall, bricht der Server die Bearbeitung ab und generiert eine Fehlermeldung. Durch das zustandslose Protokoll vermeiden wir ein aufwändiges Sessionmanagement, welches im Mobilfunkbereich durch häufig wechselnde IP-Adressen zusätzlich verkompliziert würde.

## 4 Client-Applikationen

Wie eingangs erwähnt wurde, bestehen die Möglichkeiten, die Ortung von Nutzern mobiler Endgeräte entweder von einem Desktop-PC oder von einem anderen mobilen Endgerät aus zu realisieren. Im Folgenden werden die beiden Client-Anwendungen vorgestellt.

### 4.1 Mobiltelefon-Client

Für die Umsetzung des Prototyps wurden exemplarisch Mobiltelefone als eine Klasse mobiler Endgeräte genutzt. Prinzipiell ist jedoch eine Portierung der Client-Anwendung auf z.B. PDAs oder Smartphones ohne Probleme möglich, solange diese Geräte die in Abschnitt 2 vorgestellten Anforderungen ausreichend erfüllen.

**Wahl der Programmiersprache:** Eine wichtige Anforderung bei der Wahl der Programmiersprache für die auf dem Mobiltelefon laufende Software war die Kompatibilität. Die Software für das Mobiltelefon sollte möglichst ohne größere Anpassungen auf allen gängigen Mobiltelefonen sowie PDAs lauffähig sein. Daher kamen proprietäre Systeme wie z.B. ExEn von In-Fusio<sup>5</sup> für diese Entwicklung nicht in Frage. Nahezu alle neu auf dem Markt erscheinenden Handys und PDAs verfügen jedoch inzwischen über eine eingebaute „Java Virtual Machine“. Die Java-Spezifikation unterteilt sich in mehrere Varianten, die unterschiedliche Zielgeräte abdecken. Neben dem sehr verbreiteten Standard (J2SE) und der Enterprise Edition (J2EE) bietet Sun für kleine Geräte die Java 2 Micro Edition (J2ME) an. Dieser Standard unterteilt sich je nach Anforderungen in weitere Untergruppen. Die Variante, die auf aktuellen Mobiltelefonen zum Einsatz kommt, ist MIDP, das Mobile Information Device Profile. Diese verwendet die Kilobyte Virtual Machine (KVM)

---

<sup>5</sup>[http://www.in-fusio.com/fond\\_game/technologie\\_games.php](http://www.in-fusio.com/fond_game/technologie_games.php)

und begnügt sich mit nur 32 bis 512 kb Arbeitsspeicher. Dieses Mobile Information Device Profile bietet unter anderem Netzwerkunterstützung nach dem HTTP 1.1 Standard, eine graphische Benutzeroberfläche, die sich automatisch den Gegebenheiten des verwendeten Gerätes anpasst sowie die Möglichkeit, Daten permanent zu speichern. Für die Kommunikation mit der GPS-Hardware sind die Java APIs für Bluetooth von Interesse. So wäre es möglich, mit der GPS-Einheit zu kommunizieren wenn diese beispielsweise in dem Akku des Mobiltelefons integriert wäre, was wiederum ein einfaches Nachrüsten der Technik erlauben würde.

Die geringen Hardwareanforderungen des MIDP führen jedoch zu Einschränkungen, die bei der Softwareentwicklung zu beachten sind. Da z.B. viele MIDP-unterstützende Geräte über keinen Fließkomma-Prozessor verfügen, können keine Variablen vom Typ `float` oder `double` verwendet werden.

**Installation:** Die Installation der Software kann auf zwei Arten geschehen. Entweder wird das Gerät mit einem Computer verbunden, um dann die Software auf das Mobiltelefon oder PDA zu kopieren, oder man verwendet das so genannte „Over-the-Air Provisioning“. Hierbei wird das MIDlet, so heißen Programme, die dem J2ME-Standard entsprechen, von einem Webserver geladen, was durch aktivieren eines Links auf einer WAP-Seite oder über eine normale HTML-Seite erfolgen kann. Zuerst wird nun nur eine `.jad`-Datei herunter geladen. Diese enthält unter anderem eine Beschreibung des Programms und die Dateigröße der Software. Anschließend kann der Anwender entscheiden, ob er die komplette Software herunterladen möchte. Diese enthält dann die kompilierten Klassen in einer komprimierten `.jar`-Datei.

**Programmbedienung:** Nach dem Starten der Software erscheint zuerst das Hauptmenü (siehe Abbildung 3(a)). Unter *View Map* kann die eigene Position oder auch die Position



(a) Hauptmenü

(b) Zusatzinformationen

(c) Kartendarstellung

Abbildung 3: Mobiltelefon-Client

von anderen Benutzern des Systems auf einer Karte angezeigt werden. Da die meisten modernen Mobiltelefone und PDAs über ein Farbdisplay verfügen, erfolgt die Kartendarstellung (vgl. Abbildung 3(c)) in Farbe. Auf Wunsch kann ein Fadenkreuz eingeblendet werden, welches die exakte Position darstellt. Unter dem Menüpunkt *View details* werden die Längen- und Breitengrade, der Zoomfaktor, die Breite des Kartenausschnitts, die

Höhe über dem Meeresspiegel, die Geschwindigkeit, die Richtung, die Anzahl der aktuell empfangenen Satelliten sowie einige weitere Daten (siehe Abbildung 3(b)) angezeigt bzw. eingestellt. *View settings* beinhaltet weitere Einstellungen, z.B. zur Aktualisierung der Kartendarstellung basierend auf einem Zeitintervall oder einer minimalen Positionsveränderung. Dadurch wird das übertragene Datenvolumen und somit die Kosten der GPRS Datenübertragung reduziert. Der Menüpunkt *Change User* beinhaltet die Benutzerverwaltung. Wie schon in Kapitel 3.2 erwähnt, dient sie der Zugriffskontrolle und der Benutzerauthentifizierung. Der Menüpunkt *Change target* ermöglicht - entsprechende Berechtigungen vorausgesetzt - das Anzeigen von Daten anderer Nutzer. Unter *Server data* werden Einstellungen bezüglich des Servers (z.B. Adresse und Port) vorgenommen. Dementsprechend, ermöglicht *Map settings* das Festlegen spezieller Parameter für den Zugriff auf den Kartenserver. Der vorletzte Menüpunkt *Mobile settings* umfasst spezielle Anpassung der Software an das verwendete Endgerät. Dazu gehören die Auflösung und die Farbdarstellung. Normalerweise ist dies aber nicht notwendig, da die Software anhand des Gerätes automatisch kalibriert wird. Der letzte Menüpunkt bietet eine umfangreiche Hilfefunktion.

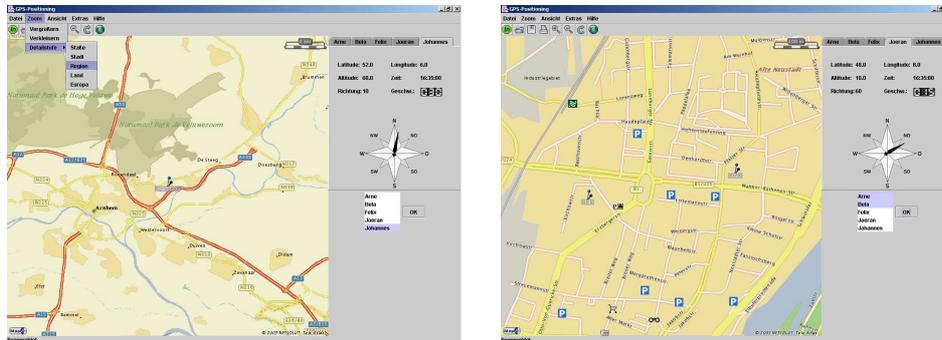
## 4.2 Desktop-Client

Da mobile Endgeräte oftmals beschränkte Darstellungsmöglichkeiten (Displaygröße) besitzen, können sie nicht alle Details der GPS-Daten sinnvoll gleichzeitig darstellen. Um dieses Problem zu lösen wurde zusätzlich zu dem in Abschnitt 4.1 vorgestellten Mobiltelefon-Client ein Desktop-Client entwickelt.

**Wahl der Programmiersprache:** Der Desktop-Client ist komplett in Java (JSDK 1.4) programmiert, da so die Plattformunabhängigkeit gewahrt bleibt und der Client auf jedem Computer, der über eine entsprechende „Java Virtual Maschine“ verfügt, benutzt werden kann. Die graphische Benutzeroberfläche ist in Swing realisiert, wodurch eine leichte und weitgehend intuitive Bedienung möglich ist.

**Funktionalität:** Der Desktop-Client (siehe Abbildung 4) ermöglicht es, die Position von mobilen Endgeräten auf einer Karte anzuzeigen. Für den Betrieb der Software ist eine Anbindung an das Internet erforderlich, da sowohl die Positionsinformationen als auch das Kartenmaterial hierüber bezogen werden. Der Zugriff auf die Positionsdaten erfolgt über das in Kapitel 3 vorgestellte string-basierte Protokoll. Der Zugriff auf das Kartenmaterial ist über HTTP realisiert: Eine einfache, aus dem vom Positionsserver erhaltenen Längen- und Breitengrad zusammengesetzte URL genügt, um dynamisch den jeweils benötigten Kartenausschnitt zu erhalten. Auf diese Anfrage wird dann die Karte im GIF-Format zurückgeliefert. Die Positionen der Nutzer werden dabei vom Kartenserver in das Bild eingefügt, was stets eine optimale Lesbarkeit der Beschriftungen sicherstellt. Hierfür müssen zuvor dem Server in einer gesonderten HTTP-Anfrage die jeweils aktuellen und einzuzeichnenden Positionen der Nutzer mitgeteilt werden.

Nach dem Programmstart ist eine Authentifizierung des Nutzers anhand des Nutzernamens und des Kennworts notwendig. Nach erfolgreicher Prüfung der Daten durch den Positionsserver wird die so genannte „Buddylist“ übertragen und im Client angezeigt. Diese Liste



(a) Detailstufe „Region“

(b) Detailstufe „Straße“

Abbildung 4: Desktop-Client

enthält alle Nutzer mobiler Endgeräte, die dem Anwender der Software ein Ortungsrecht eingeräumt haben. Über die „Buddylist“ erfolgt die Auswahl der Nutzer, die auf der Karte eingezeichnet werden. Dabei ist es möglich, beliebig viele Nutzer gleichzeitig anzuzeigen. Der Zoomfaktor wird automatisch so gewählt, dass alle Nutzer im Kartenausschnitt repräsentiert sind.

Zusätzlich zu der Position auf der Karte werden zu den ausgewählten Nutzern statistische Daten in einem Kartenreiter angezeigt. Diese Statistik enthält zum einen Angaben über die aktuelle geographische Position (Längen- und Breitengrad sowie die Höhe über Normal Null). Zum anderen werden auch Angaben über die aktuelle Geschwindigkeit, die aktuelle Bewegungsrichtung und über den Zeitpunkt der letzten Messung angezeigt. Diese Angaben sind alle in Textform gegeben. Zur besseren Übersicht werden Richtung und Geschwindigkeit noch durch eine Windrose dargestellt, bei der mit steigender Geschwindigkeit der Richtungszeiger wächst.

Das Zoomen der Karte ist über zwei verschiedene Methoden möglich. Zum einen kann einfach stufenweise herein- oder herausgezoomt werden. Zum anderen ist es möglich, über fünf vorgegebene Detailstufen (Straße, Stadt, Region, Land und Europa) auch bequem in größeren Sprüngen zu zoomen.

Über ein Wiedergabe-Modul kann der Nutzer zudem sämtliche auf dem Positionsserver gespeicherte Routen von früheren Zeitpunkten wiedergeben. Hierzu wird die Route vom Server übertragen und in einer gewählten Geschwindigkeit abgespielt. Die Zuordnung der Positionen zu einer Route wird über einen Routennamen geregelt, die der Sender der Position festlegen kann. Alle Positionen mit dem gleichen Routennamen werden dann zu einer Route zusammengefasst.

In den Programm-Einstellungen lassen sich Deutsch und Englisch als Programmsprache wählen. Außerdem ist es möglich einen Standard-Nutzer anzulegen, damit die Anmeldung beim Start des Programms vereinfacht wird.

## 5 Zusammenfassung und Ausblick

Die Ortung von mobilen Endgeräten ist eine zwingende Voraussetzung für die Realisierung lokationsbasierter Dienste. Im Rahmen des vorgestellten Projektes wurde ein System entwickelt, das es ermöglicht, die Position eines mit GPS ausgestatteten mobilen Endgerätes zu bestimmen. Dazu wurden neben einem Server, der Positions- und Nutzerinformationen verwaltet, zwei Client-Anwendungen implementiert. Durch diese ist der Zugriff auf die Daten von einem Mobiltelefon bzw. einem Desktop-PC möglich. In dem Papier wurde die Gesamtarchitektur präsentiert und einige Aspekte der Implementierung sowie der Anwendung des Systems erläutert.

Bei dem vorgestellten System handelt es sich um einen Prototyp, der noch zahlreiche Punkte für zukünftige Arbeiten offen lässt. Darunter sind u.a. (1) Skalierbarkeit des Servers bei einer großen Anzahl von Nutzern und (2) verschlüsselte Kommunikation zwischen den beteiligten Komponenten. Neben den technischen Fragen, sind Anwendungen ein zentraler Punkt für Anschlussfragestellungen. Wie eingangs erwähnt, soll und wird in erster Linie die Genauigkeit der Lokalisierung des „GSM-Schutzengels“ durch die GPS-Ortung verbessert werden.

**Danksagung:** Unser Dank gilt der Infineon Technologies AG, der NETSOLUT GmbH sowie der SOS r. AG, welche die Entwicklung durch Geld und Sachmittel unterstützt haben.

## Literatur

- [Eri01] Ericson. Ortsbezogene Mobilfunkdienste - Location Based Services, October 2001. <http://www.ericsson.de/downloads/presenews/HintergrundpapierLBS.pdf>.
- [ETS93] ETSI. European digital cellular telecommunications system; Attachment requirements for Global System for Mobile communications (GSM) mobile stations. Technical basis for regulation, European Telecommunications Standards Institute, Sophia Antipolis Cedex, November 1993.
- [ETS98] ETSI. Digital cellular telecommunications system (Phase 2+) (GSM); General Packet Radio Service (GPRS); Service description; Stage 1 (GSM 02.60 version 6.1.1). European standard (communication series), European Telecommunications Standards Institute, Sophia Antipolis Cedex, November 1998.
- [ETS99] ETSI. Digital cellular telecommunications system (Phase 2+); Attachment requirements for Global System for Mobile communications (GSM); High Speed Circuit Switched Data (HSCSD) multislots mobile stations; Access (GSM 13.34 version 5.0.3). European standard (communication series), European Telecommunications Standards Institute, Sophia Antipolis Cedex, March 1999.
- [GBAP03] B. Gipp, J. Beel, F. Alcala, and L. Petersen. Der GSM-Schutzengel... Internetseite, 2003. <http://www.gsm-schutzengel.de>.
- [Kap96] E. D. Kaplan. *Understanding GPS: Principles and Applications*. Artech House Telecommunications Library. Artech House, March 1996.
- [NET03] NETSOLUT GmbH. Map24 - Online Routenplaner und interaktive Karten für Europa. Internetseite, 2003. <http://www.map24.de>.
- [Sun03] Sun Microsystems, Inc. J2ME - Java 2 Platform, Micro Edition. Internetseite, 2003. <http://java.sun.com/j2me/>.