

INFORMATIK BERICHTE

324 - 5/2005

2. GI/ITG KuVS Fachgespräch Ortsbezogene Anwendungen und Dienste

Jörg Roth (Editor)



FernUniversität in Hagen

Fachbereich Informatik
Postfach 940
D-58084 Hagen

Vorwort

Der Aufenthaltsort eines mobilen Benutzers stellt eine wichtige Information für Anwendungen aus den Bereichen *Mobile Computing*, *Wearable Computing* oder *Ubiquitous Computing* dar. Ist ein mobiles Endgerät in der Lage, die aktuelle Position des Benutzers zu bestimmen, kann diese Information von der Anwendung berücksichtigt werden – man spricht dabei allgemein von *ortsbewussten* oder *ortsbezogenen Anwendungen*. Eng verknüpft mit dem Begriff der ortsbewussten Anwendungen ist der Begriff der *ortsbezogenen Dienste* (engl. *location-based services*). Hierbei handelt es sich beispielsweise um Dienste, die Informationen über den aktuellen Standort übermitteln. Mittlerweile werden solche Dienste kommerziell eingesetzt und erlauben etwa, dass ein Reisender ein Hotel, eine Tankstelle oder eine Apotheke in der näheren Umgebung findet. Nicht zuletzt durch die Einführung von UMTS erwartet man auf dem kommerziellen Sektor einen Boom für ortsbezogene Anwendungen und Dienste.

Nach dem erfolgreichen ersten Treffen in Hagen im Jahr 2004 fand am 16. und 17.6.2005 das zweite Fachgespräch "Ortsbezogene Anwendungen und Dienste" in Stuttgart statt. Ziel des Fachgesprächs war, Wissenschaftlerinnen und Wissenschaftlern aber auch Vertretern der Industrie die Möglichkeit zu einem intensiven Gedankenaustausch zu geben. Grundlage dazu waren kurze Vorträge zu verschiedenen Forschungsarbeiten aus diesem Bereich, die in diesem Bericht zusammengefasst sind.

Jörg Roth im Juni 2005

Inhalt

Sitzung 1: Context-, Situation-Awareness und Positionsmodelle

<i>Addressing geographic objects of unique location areas</i>	7
Holger Kirchner, Daniel Glöckner, Gerald Bieber, Sven Gabrecht	
<i>Benefits of Context Models in Smart Environments</i>	12
Daniela Nicklas, Susanne Bürklen, Tobias Drosdol, Nicola Hönle	
<i>IN:SIGHT – Supporting Situation-Awareness in Mobile Applications</i>	16
Ulrich Meissen, Stefan Pfennigschmidt, Agnès Voisard, Tjark Wahnfried	
<i>TagAttack: Ein kontextbasiertes mobiles Mehrbenutzerspiel</i>	20
Hartmut Ritter, Rolf Winter, Jochen Schiller	

Sitzung 2: Positionsbestimmung

<i>Improvement of Accuracy for GPS</i>	24
Stephan Kopf, Thomas King, Wolfgang Effelsberg	
<i>A Location System based on Sensor Fusion: Research Areas and Software Architecture</i>	28
Thomas King, Stephan Kopf, Wolfgang Effelsberg	
<i>Modelling Location Sensors</i>	33
Jörg Roth	
<i>A Provider-Independent, Proactive Service for Location Sensing in Cellular Networks</i>	37
Andreas Hartl	

Sitzung 3: Plattformen und Rahmenwerke, ortsabhängige Konfigurationen und Profile

<i>Context-Driven Service Discovery in Mobile Commerce</i>	41
Philipp Bostan, Colin Atkinson, Oliver Hummel, Giovanni Falcone	
<i>A Research Platform for Location-Based Applications</i>	46
Dominic Heutelbeck, Matthias Hemmje	
<i>Profilbasierte Rekonfiguration kleiner Endgeräte</i>	50
Gerhard Fuchs, Sébastien Truchat, Falko Dressler	
<i>Verbesserung personalisierter Informationsdienste auf Mobilfunkterminals durch Integration Ortsbezogener Daten</i>	54
Hans Weghorn	

Sitzung 4: Sicherheit, Vertraulichkeit, digitales Rechtmanagement

<i>Mit IDM und Mittler zu mehr Privatsphäre in LBS</i>	58
Tobias Kölsch, Lothar Fritsch, Markulf Kohlweiss, Dogan Kesdogan	
<i>Ortsabhängiges Digitales Rechtmanagement</i>	62
Thomas Mundt	
<i>Datenschutzmechanismen für Ortsinformationen aus der Sicht zukünftiger Anwendungen</i>	66
Georg Treu, Axel Küpper	

Addressing geographic objects of unique location areas

Fachgespräch für “Ortsbezogene Anwendungen und Dienste“, Stuttgart, 2005

Holger Kirchner Daniel Glöckner
{Holger.Kirchner, Daniel.Gloeckner}@ipsi.fraunhofer.de
Fraunhofer IPSI
Dolivostrasse 15
64293 Darmstadt

Gerald Bieber, Sven Gabrecht
{Gerald.Bieber, Sven.Gabrecht}@igd-r.fraunhofer.de
Fraunhofer IGD
Joachim-Jungius Str. 11
18059 Rostock

Abstract: In this paper, we investigate a set of different location models that can be applied to location based services (LBS). Searching geographic information in an information space is not new at all. Many projects using positioning systems such as GPS search for information in their information space describing a query as a range. We will show that Natural Area Code (NAC) can be used to address objects in a geographic area without taking additional network traffic. Our idea is based on NAC’s concept of optimizing addressing by location area shifting. Local Area Code (LAC) improves the practical use for smaller areas like 7km by 5km.

Keywords: location-awareness, query geographical objects, mobile data management

1 Introduction

1.1 Motivation

Many devices became location aware with the advances in mobile hardware and increased distribution of mobile wireless infrastructures. For example, increasing functionalities and improvements allow mobile phones to provide up-to-date information. The quality of the resulting data is very poor (in terms of detail), functional support is limited and accuracy of positioning is not very high. When backend systems are not accessible (e.g. in rural areas), the user might pass the destination, before being informed that he/she had actually arrived. There are no applications that provide that specific navigational functionality. On the other hand, systems such as Route66¹ or AutoRoute² are available offline. Online updates are limited to certain information only - for example information about traffic jams. Such applications run ‘only’ on desktop computers, which is not really very practical for many mobile users. More sophisticated applications such as TomTom³ can be used on PDAs (personal digital assistance). The problem is that users have to decide what mapping data and related points-of-interests (PoI) are relevant to them and select the respective data, etc. that they want to bring along. If they change their minds and leave the previously selected geographic area, they have to go back to their desktop and preload the new information.

This can be improved by prefetching [Kir2005] or hoarding [Kub2002], based on a prediction scheme about the user’s planned route, as they increase cache hits and reduce response times. The system will automatically transfer data in advance depending on the user’s position and traveling direction. If the user needs the information, the data will already be available in his local cache. It is necessary to provide relevant information in time to address a geographical information space efficiently.

1.2 Outline

After we look in section 2 at different projects and how positioning systems and location models can be applied, we investigate the use of different location models. We review NAC and its benefits. In section 2.4 a concept of shifting location nodes is proposed before LAC is introduced, followed by the conclusions in section 3.

¹ <http://www.66.com>

² <http://www.microsoft.com/germany/autoroute/default.msp>

³ <http://www.tomtom.com>

2 Related Work

The need to access information while on the move is motivated by many projects such as Nexus [KLSR1999], XyberScout [BI2003] and eureauweb [KMK+04] to mention a few. The current position of mobile users is often used as the key parameter to retrieve relevant information. It is essential to query mobile objects within an area of interest [Rot2002, BBR2002]. In general, one of three different types of location systems can be applied: *cell-based*, *coordinate-based* and *semantic or logical locations*. The goal of using these location models is to define the scope of geographic validity and thus be able to provide a mechanism for searching or addressing geographic objects.

2.1 Positioning and Location Models

Cell-based locations are often used on mobile phones. They do not require any additional hardware and can be used when base stations are available. Accuracy usually ranges from between hundreds of meters and several kilometers depending on the cell size; with triangulation being able to improve accuracy. Further improvements could be achieved by also considering the signal strength, but this would require a fingerprint for the different signals and additional information in the database to compare the signals, which can only be done for small local areas. Otherwise it is necessary to communicate with backend systems to resolve the location issue.

Semantic or logical locations can be used in a wide range of applications. However, no positioning system can provide such information by itself, there is always some transformation effort necessary to compile e.g. cell-IDs or coordinates into semantic or logical location data. Figure 1 (left side) shows the user moving on the green layer. He currently has access to a base station describing the validity of the geographical objects for a particular range (hot spot). Leaving the area may result in the user: a.) moving into an area where there is no base station or b.) reaching an area where the base station cannot be identified and, consequently, the location is also not resolved. The transformation can only work if one of the well-known world wide addressing systems is used such as state.place.street.housenumber. A hierarchy of location names can be used to resolve names and to build a name space for a world-wide addressing scheme. The disadvantage is that such an infrastructure needs to be built from scratch and supported by all services.

The advantages of *coordinate-based locations* are obvious: they provide worldwide unique coordinates and accuracy is at the highest level possible. They are normally used for outdoor applications (support for indoor is very limited). Addressing the geographical objects of coordinates is often done by GIS, which requires access to a server at the backend to provide relevant information and to process the spatial query. There is very little research on mobile GIS.

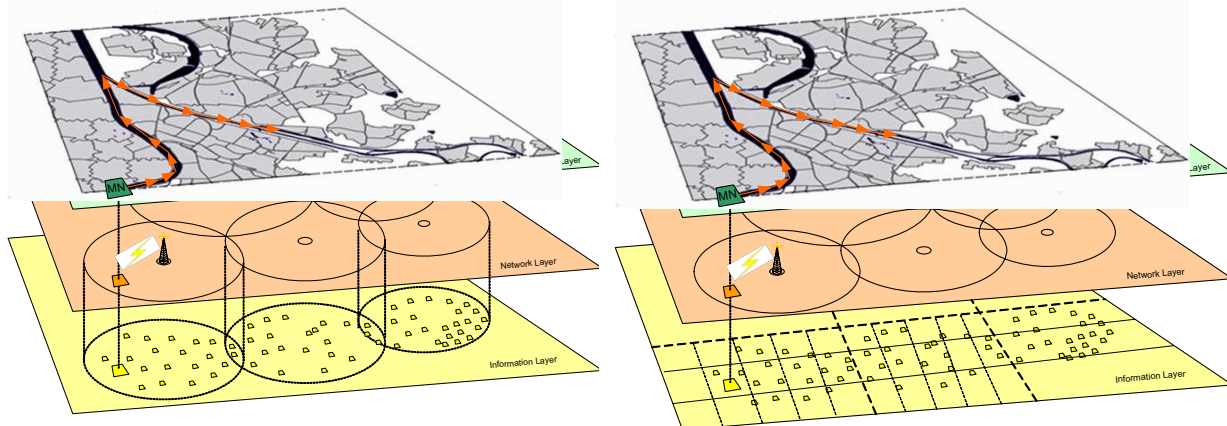


Figure 1. Location Models: scopes defined by a.) cells and b.) areas

2.2 Natural Area Code (NAC)

Another not very popular coordinate system is NAC⁴. NAC was originally created to identify unique geographic areas and has similar properties as the Gauß-Krüger coordinates. The world is represented by a set of characters from 0 to 9 and B to Z. Some letters do not get used (A, E, I, O, U, Y). The

⁴ <http://www.nacgeo.com/nacsite/>

coding system also allows the description of location areas (the more letters the smaller the area). The following lists examples of possible combinations:

NAC		Area	Size
NAC: H	R	Germany	1000km x 670km
NAC: HC	R8	Frankenstein and surrounding	33km x 22km
NAC: HC8	R8Z	Mountain top	1.1km x 0.7km
NAC: HC8G	R8Z1	Tower of Burg Frankenstein	36m x 25m
NAC: HC8GF	R8Z13	Platform of the tower	1.2m x 0.8m

Table 1. NAC zones.

An example for the tower of “Burg Frankenstein” in the south of Hessen can, therefore, be described as latitude N49°47.615' and longitude E8°40.114'. The NAC representation is HC8GF R8Z13. Another interesting effect, which is also supported by NAC, is that stripes can also be addressed. This means that a combination of e.g. 2C Q87M is possible and will address a zone of 33 km by 0.7 km.

2.3 Locations and Resolution

Obviously, there are many possibilities of how to use location models in different application areas and, of course, there are good reasons for choosing the right ones. Developing LBS on small resource limited devices requires location models which are efficient (in terms of computation) and do not take up too much network traffic. This allows for only a very limited amount of algorithms with complex operations. The scheme should not require too many resources because an increase in the number of objects would result in an equivocal multiplication of the necessary main memory; and the schema should be convenient to use.

[Rot2005] as proposed in Figure 2 provides a flexible positioning for LBS, but requires additional network traffic to resolve the coordinates into a global, unique, physical and semantic location. Our approach (figure 2b) is well-defined and does not require any extra communication cost to address a set of objects because each service can do the computations itself.

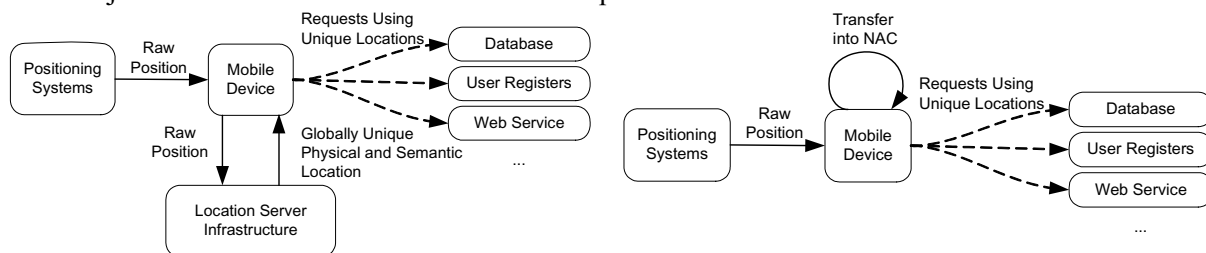


Figure 2. Location resolution: a.) hierarchical approach b.) approach using NAC

2.4 Shifting Location Areas

Some specific user situations might involve a user finding him at a location where objects are close enough to the border to interfere (see figure 2a) - to solve the problem there are three solutions possible:

- Defining a range of NACs to address a certain area; this requires the knowledge of the NAC areas which are in the user's neighborhood. An area of nine fields can cover the area around a user.
- Using the next higher NAC level (one letter less) and covering a range which addresses too many objects. In the worst case scenario, this does not have any effect when the scope of a small area is the same as it is for a large area. A major disadvantage is that lots of geographical objects may appear for large areas.
- Location areas can be applied to shift the user to the centre of an area. This is obviously not covering too many objects and additional efforts for managing objects are not necessary (see figure 2b). Adding or removing a fixed value for all geographical objects can simply be done by transferring the current position of a user from lat/long into NAC. Applying the reverse function from NAC to Gauß-Krüger will result in the corners of the respective NAC area. Measuring the distance to all sides provides the input values of the delta for which the NAC needs to be shifted.

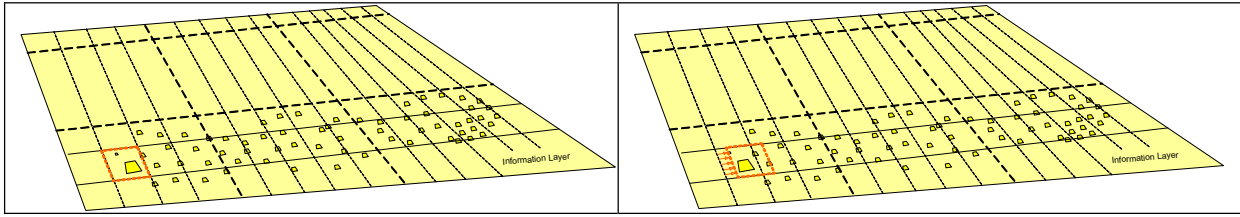


Figure 3. Location Models: scopes defined by a.) cells and b.) areas

One of the major advantages for any location model is that there is no additional communication necessary to update local devices with location knowledge and the need to distribute the location model to databases is not required (when compared to hierarchical models) to define a scope's validity. This is not required for NAC because any coordinate can be transferred into NAC.

2.5 Location Area Code (LAC)

The use of NAC is, in some cases, not practical enough because the location area does not cover areas of 7 km by 5 km. Another disadvantage of NAC is the usage of characters for the coding of the locations. The character representation does not allow an efficient implementation of the different algorithms in terms of computation, which is of crucial importance for mobile devices. That is why we adapt the approach for NAC and modify it: The LAC consists of two 3 byte strings, one for the latitude and the other for the longitude, which allow 6 levels of precision. The code is stored in the hexadecimal system, which results in a maximum of runtime and memory efficiency.

To calculate the LAC of a geodetic position, longitude and latitude are mapped to the interval [0,1]. In each LAC step, the interval is subdivided into 16 parts. The division is applied recursively to each subinterval again, resulting in greater precision. The first LAC level divides the interval into 16 parts, the second into 16^2 and so on. With 6 divisions, a precision of 1.8 x 1.2m is reached.

Transformation from (Longitude, Latitude, Altitude) into LAC			
LONG = (Longitude + 180)/360			
x1 = Integer part of (LONG*16)			
x2 = Integer part of ((LONG*16-x1)*16)			
x3 = Integer part of (((LONG*16-x1)*16-x2)*16)			
x4 = Integer part of ((((LONG*16-x1)*16-x2)*16-x3)*16)			
x5 = Integer part of ((((LONG*16-x1)*16-x2)*16-x3)*16-x4)*16)			
x6 = Integer part of ((((LONG*16-x1)*16-x2)*16-x3)*16-x4)*16-x5)*16)			
LAT = (Latitude + 90)/180			
y1 = Integer part of (LAT*16)			
y2 = Integer part of ((LAT*16-y1)*16)			
y3 = Integer part of (((LAT*16-y1)*16-y2)*16)			
y4 = Integer part of ((((LAT*16-y1)*16-y2)*16-y3)*16)			
y5 = Integer part of ((((LAT*16-y1)*16-y2)*16-y3)*16-y4)*16)			
y6 = Integer part of ((((LAT*16-y1)*16-y2)*16-y3)*16-y4)*16-y5)*16)			
LAC (hexadecimal)		Area	Size
LAC: 8	C	Germany	1875km x 1256.3km
LAC: 86	C6	Hessen	117.2km x 78.5km
LAC: 862	C6D	Frankenstein and surrounding	7.3km x 4.9km
LAC: 862A	C6D1	Burg Frankenstein	458m x 306m
LAC: 862A1	C6D14	Tower of Burg Frankenstein	29m x 19m
LAC: 862A10	C6D14A	Platform of the tower	1.8 x 1.2m

Table 2. Transformation from (Longitude, Latitude, Altitude) into LAC.

Compared to NAC, LAC is much more efficient in terms of computation. The code can be stored efficiently in 6 bytes without 'wasting' any bits. Due to the binary representation, the algorithms can be optimized more strongly and valuable CPU cycles on the mobile device can be saved. If we take a 5 character NAC for longitude and latitude, with a similar precision as LAC in the example above, the storage would take 40 Bytes (if a character was represented by a Byte).

3 Summary and Conclusion

Location models can be used in many application areas. Very often different models get used at the same time, based on a specific need and application. Cell-based positioning can be used to improve low accuracy but is not available worldwide. The requirements for using semantic or logical location models are often structured hierarchically and necessitate the resolution of a position into a location. Coordinate-based locations are very accurate and available worldwide. Coordinates like latitude or longitude are missing; therefore, the scope of an area cannot be described. NAC eliminates the addressing of a group of geographical objects and the coding fails to describe areas e.g. of 7 by 5 km. LAC allows the addressing of areas for smaller scopes. Comparing different location models by memory use shows that our approach offers the best benefit. Further work can be done to apply LAC in other application areas. Further investigation can also be done by using location area shifting.

4 References

- [Kir2005] Kirchner, Holger: *"Using context of a mobile user to prefetch relevant information"*. submitted for "Fourth International ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE 2005) in conjunction with SIGMOD/PODS 2005", Baltimore, Maryland, United States, June 12, 2005, <http://db.cs.pitt.edu/mobide05>.
- [Kub2002] Kubach, Uwe: *"Vorabübertragung ortsbezogener Information zur Unterstützung mobiler Systeme"*, Dissertation, Fakultät Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart, 2002.
- [KLSR1999] Hohl, F; Kubach, U.; Leonhardi, A.; Schwehm, M.; Rothermel, K.: *"Nexus - an open global infrastructure for spatial-aware applications"*. 5th Intern. Conference on Mobile Computing and Networking (MobiCom '99), Seattle, WA, USA, 1999. ACM Press.
- [BI2003] Bieber, Gerald; Ide, Rüdiger: *XyberScout: "A Platform for the Efficient Construction of Mobile Location Aware Information Systems"*. In: Sprague, Ralph H. (Ed.); IEEE Computer Society: 36th Annual Hawaii International Conference on System Sciences. Proceedings. Washington : IEEE Computer Society Press, 2003, pp. 297-305.
- [KMK+04] Kirchner, Holger; Mahleko, Bendick; Kelly, Mike; Krummenacher, Reto and Wang, Zhou: *"eureauweb – An Architecture for a European Waterways Networked Information System"*. Conference on Information and Communication Technologies in Tourism 2004 (ENTER 2004); Cairo, Egypt, 26-29 January 2004; Published by Springer, Wien, New York; ISBN 3-211-20669-8, pp 65-75, Feb. 2004.
- [Rot2002] Roth, Jörg: *"Context-aware Web Applications Using the PinPoint Infrastructure"*; IADIS International Conference WWW/Internet 2002, Lissabon, Portugal, 13.-15., IADIS Press, 3-10, Nov. 2002.
- [BBR2002] Bauer, M.; Becker, C.; Rothermel, K.: *"Location Models from the Perspective of Context-Aware Applications and Mobile Ad Hoc Networks"*, Personal and Ubiquitous Computing, Vol. 6, No. 5, 322-328, Dec. 2002.
- [Rot2005] Roth, Jörg: *"A Decentralized Location Service Providing Semantic Locations"*, Habilitation, Informatik Bericht 323, Fernuniversität Hagen, Jan. 2005.

Benefits of Context Models in Smart Environments

Daniela Nicklas, Susanne Bürklen, Tobias Drosdol, Nicola Höhle
Universität Stuttgart, Institute of Parallel and Distributed Systems
Universitätsstraße 38, 70569 Stuttgart, Germany
<firstname>.<lastname>@informatik.uni-stuttgart.de

Abstract: Ongoing technologic advances drive the emergence of *smart items*, everyday objects with embedded microcomputers and communication capabilities. In a *smart environment* a multitude of such smart items exist to assist its users. In this paper, we will show how smart environments can benefit from the concepts of the Nexus platform, an open pervasive computing system that supports various kinds of context-aware applications by providing a federated, potentially global context model.

1 Introduction

The vision of *ubiquitous computing* has drawn increasing attention of researchers in the past years. Ongoing technologic advances drive the emergence of *smart items*, everyday objects with embedded microcomputers and communication capabilities that can sense their current state or changes in their environment by integrated sensor systems. Such devices can range from smart household appliances to smart cups or picture frames. Smart items cooperating with each other can form a *smart environment*, in order to assist users in their daily work and simplify their life. Examples are smart homes that help elderly people [KOA+99] or smart offices that provide new methods for interacting with devices and information [SGH+99] [RC00].

Yet, the realization of such an environment is a difficult task. It typically incorporates a multitude of very heterogeneous devices that need to be set up and configured individually. Cooperation between these devices furthermore requires interoperability of data formats and exchanged communication messages. In the following, we will therefore show how smart environments can benefit from the concepts of the Nexus platform and which use cases can be easily developed using its support.

In its beginning, the Nexus project¹ [Nexus] focused on location-aware applications that support mobile users and take into account their spatial context, i.e. their position. Such applications typically run on a mobile client that is equipped with location sensors and is capable of wireless communication. The spatial range of these applications can be quite large, from campus up to city or country level. Examples are tourist guides [LKAA96] [CDM+00] or mobile navigation applications [BKW02] [HHT+01].

Unlike many other approaches in that area, the Nexus project focuses not on a specific application, but on platform support for all kind of location-aware applications. The main idea is to provide a global-scale context model that is federated over autonomous local context models [NM04].

In the following section, we briefly introduce the Nexus platform. Then we show how smart environments can benefit from context models that can be managed by the platform.

¹ The Nexus project was funded 1999-2002 by the German Research Association (DFG) under grant 200989 and is now continued as Center of Excellence (SFB) 627.

2 The Nexus Platform

In this section, we briefly describe our present context managing approach—the Nexus platform [NM04].

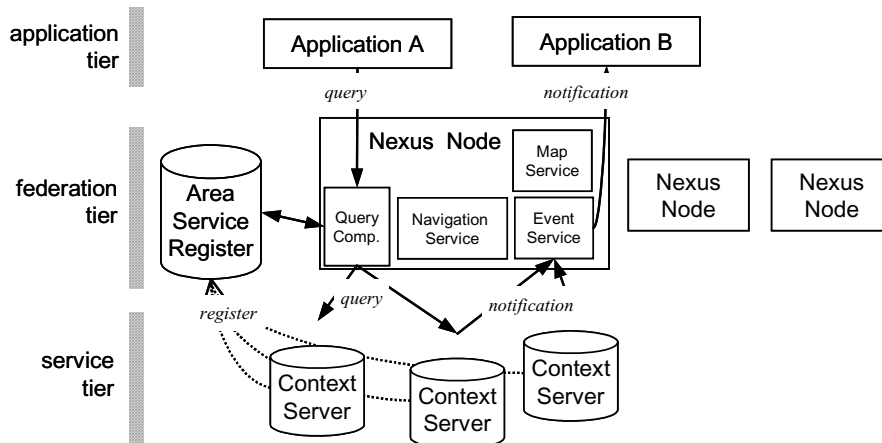


Figure 1: Architecture of the Nexus platform

The Nexus platform (Figure 1) is an open pervasive computing system that supports various kinds of context-aware applications by providing a federated, potentially global context model. To achieve this goal, local data providers (so called context servers) offer their context data in a standardized yet extensible way: they implement a given interface (the Augmented World Query Language, AWQL) and present their data according to a given model (the Augmented World Modeling Language, AWML). There can be many different implementations of a context server [GBH+05]. For providing large scale static models, we used a spatially enhanced database, but even small scaled sensor platforms or sensors itself can be used as context servers.

The model is object-based: there are data objects that have attributes. Up to now we have defined a basic set of object types, of which we think that they are essential for context-aware applications. This basic set includes object types for real objects as well as for virtual objects. We can model map data like streets and buildings, ground plans of buildings, and sensor objects with sensor measurement values. Objects can be given in different levels of detail.

All objects of a data provider lie within the boundaries (service area) of a so called Augmented Area. Augmented Areas can overlap in space and also in the objects they provide. Therefore, the same real world object can be part of different Augmented Areas and can be modeled using different data objects. We call them multiple representations. Context servers register their Augmented Areas and contained object types in the Area Service Register.

A federation node mediates between applications and context servers. It has the same interface as a context server, but does not store models. In addition to the query functionality, every federation node supports value-added services. They use the federated context model to implement advanced services having their own interfaces. The map service, for example, renders maps based on a selected area. With GeoCast, you can address a message to a geographic area, sending it to every mobile and stationary host that is currently located in this region.

3 Applying Context Models to Smart Rooms

In this section, we introduce some use cases for realizing smart environments by using context models. All these use cases could also be implemented and maintained without using a context model. However, the use of context models, as provided by the Nexus platform, offers appealing benefits. The context model acts as a layer of abstraction between applications and

the installed hardware, e.g., sensors. Thus, all interaction between different components is based on a common interface, easing the deployment of new items and allowing applications to be used likewise in any smart room. In the following, this is illustrated by means of different sample use cases:

- *Service discovery*: Discovering available services in a smart environment will typically rely on a service registry, which allows publishing and querying installed services, their offered features and the protocols they support. Instead of using a local registry for every smart room, this task could be carried out by a common context model as well. Defining the services and all their properties within such a model additionally enables applications to select an appropriate service based on their current context (e.g., selecting services with a certain spatial scope). For example, a context-aware search for a presentation service can always provide the output device that is currently located closest to the user.
- *Object Positioning*: Automated positioning of mobile entities is another important foundation for many features in a smart environment. This can be achieved by a wide range of different approaches. For example, RFID-readers can determine tagged items in their vicinity, installed cameras can identify moving objects through scene analysis, and a pressure-sensitive floor can locate users. All such techniques typically require matching the observed data with substantial reference values. Using a context model, such reference data must be maintained exactly once per object and can be shared across different smart environments. Furthermore, the accuracy of positioning can often be enhanced by utilizing additional context information. E.g., if an object is known to be in the room, then its position is bordered by the wall.
- *Object Identification*: In many situations, visitors of a smart room can be assisted by providing detailed information about certain objects (like a painting on the wall). Identifying the object of investigation should thereby be as easy as pointing to it. This could, for instance, be realized by detecting the target of a laser pointer with installed cameras. The selected entity at that position can then be determined by querying a detailed context model, which contains the locations of all available objects. This model can also provide detailed information about the object.
- *Integration of new hardware*: Without a common model, applications and services of a smart environment have to be tailored to the technical details of all installed hardware. This requires adaptation of the software whenever hardware is added or replaced. Using a context model, this integration can be done more easily in a context aware manner, since these details only have to be specified once in the context model.
- *Remote control of entities*: A lot of scenarios require remote access to the dynamic information and controls of a smart room. For example, during a teleconference one participant might wish to control the smart board of the distant party to visualize his statements. Using a common context model, no customized remote control for this particular item is required. Instead, it can be accessed directly through the Nexus platform.
- *Context events*: Another possibility of interacting with the Nexus platform is the asynchronous notification about the occurrence of certain event conditions. E.g., the user can register to be notified when the temperature in a room exceeds a certain value, or two persons meet. By integrating smart environments into the Nexus platform, this mechanism can be shared throughout all different environments.

4 Conclusion

The presented use cases show clearly, how the realization of smart environments can benefit from utilizing the Nexus platform. A shared context model provides a valuable abstraction of a specialized smart item. It thus forms the basis for interoperability between all kinds of smart items and allows for smooth interaction with them, superseding their manual adaptation and

configuration. This eases the deployment and modification of a smart environment as well as the integration of new items into it. Additionally, mobile devices can easily be used in very different smart environments, without prior tailoring of their applications. These applications can utilize the shared context model to gather information about their current environment and to interact with available services in a well defined way.

At the same time, this approach also offers the possibility to integrate a digital model of the smart environment into the Nexus platform without additional effort. This allows for accessing its dynamic information for other location-aware applications, that can seamlessly use the smart room context data together with other, spatially overlapping context models (e.g. from the building or the surrounding environment).

Within our Nexus-Lab at the University of Stuttgart, we are currently working on a prototypical realization of a smart room based on the Nexus platform. Amongst others, this will feature an Everywhere Display (a steerable video projector), a 3D-sound system, a smart board, several pivoted cameras, IR-beacons, RFID-readers, and a multitude of sensor systems. We are looking forward to provide an insight into the practical experiences gained throughout this project in the future.

References

- [BKW02] J. Baus, A. Krüger and W. Wahlster: A resource adaptive mobile navigation system. Proceedings of the International Conference on Intelligent User Interfaces, San Francisco, USA, 2002
- [CDM+00] K. Cheverst, N. Davies, K. Mitchell, A. Friday and C. Efstratiou: Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences. Proceedings of CHI 2000, Netherlands, 2000
- [GBH+05] M. Grossmann, M. Bauer, N. Hönle, U.-P. Käppeler, D. Nicklas and T. Schwarz: Efficiently Managing Context Information for Large-scale Scenarios. Proceedings of the 3rd IEEE Conference on Pervasive Computing and Communications (PerCom 2005), Kauai Island, Hawaii, USA, 2005
- [HHT+01] T. Höllerer, D. Hallaway, N. Tienna and S. Feiner: Steps towards accommodating variable position tracking accuracy in a mobile augmented reality system. IJCAI-2001 Working Notes: AI in Mobile Systems (AIMS), 2001
- [KOA+99] C. Kidd, R. Orr, G. Abowd, C. Atkeson, I. Essa, B. B. MacIntyre, E. Mynatt, T. Starner and W. Newstetter: The Aware Home: A Living Laboratory for Ubiquitous Computing Research. Proceedings of the Second International Workshop on Cooperative Buildings (CoBuild), 1999
- [LKA+96] S. Long, R. Kooper, G.D. Abowd and C.G. Atkeson: Rapid prototyping of mobile context-aware applications: The cyberguide case study. Proceedings of the Second Annual International Conference on Mobile Computing and Networking (MobiCom '96), Rye, New York, USA, 1996
- [Nexus] The Nexus Project Homepage: <http://www.nexus.uni-stuttgart.de>
- [NM04] D. Nicklas and B. Mitschang: On building location aware applications using an open platform based on the NEXUS Augmented World Model. Software and Systems Modeling, 2004
- [RC00] M. Roman and R.H. Campbell: GAIA: Enabling Active Spaces, Proceedings of the 9th ACM SIGOPS European Workshop, Kolding, Denmark, 2000
- [SGH+99] N. A. Streitz, J. Geißler, T. Holmer, S. Konomi, C. Müller-Tomfelde, W. Reischl, P. Rexroth, P. Seitz and R. Steinmetz: i-LAND: An interactive Landscape for Creativity and Innovation. ACM Conference on Human Factors in Computing Systems (CHI'99), Pittsburgh, Pennsylvania, USA, 1999

IN:SIGHT – Supporting Situation-Awareness in Mobile Applications

Ulrich Meissen
Stefan Pfennigschmidt
Agnès Voisard
Tjark Wahnfried

Fraunhofer Institute for Software
and Systems Engineering (ISST)
Mollstraße 1, 10178 Berlin, Germany

{ulrich.meissen, stefan.pfennigschmidt,
agnes.voisard, tjark.wahnfried}@isst.fraunhofer.de

Abstract

In recent years, there has been considerable development to support personalized services that take the situation of a user into account. One of the major problems in this area is to ensure an optimal information supply and to prevent information overload. Comparing situations predicted by the system with the expectations of a user yields information about “so-far-unexpected” changes the user should be informed about. With IN:SIGHT, this paper describes an approach to support situation-awareness and introduces an underlying model that additionally handles information from various sources.

Keywords: *Situation awareness, pervasive computing, mobile information services, demand-oriented information supply, information logistics.*

1. Introduction

Demand-oriented information supply is one of the key functionalities for efficient mobile information services. Many approaches exist that consider personalization and that adapt information supply to a user context in order to deliver appropriate information (e. g., location-based services [9], situation-dependent services [2]). Information and communication overload has been recognized to be a severe problem in such applications especially in situations where the user has only limited receptiveness.

In [8] the term “situation-awareness” was borrowed from the areas of aviation and car driving to describe a new class of applications in information logistics that are able to recognize a user’s situation and to use it to provide him or

her with appropriate information. Usually such applications adapt themselves to a user’s current and future situations, hence are able to react situation-dependent. However, beyond that, they can be used as a means to support situation-awareness of the user him- or herself. Imagine that a user is driving down highway A9 in Germany from Berlin to Munich and that severe weather conditions – like a hailstorm crossing the road – have been detected. That means the user will get into a so far unexpected situation “driving through a hailstorm”. He or she should be made aware of that upcoming situation. Similar application scenarios can be found in various domains, where it is necessary to keep a user informed about the changing of plans, about dangerous, unpleasant or in general “interesting” situations. Questions arising with respect to the problems sketched above are: *when to inform the user* and *how to inform the user*. The solution proposed in the present paper provides a simple yet powerful model as a basis to provide answers to these questions.

The foundation for the work presented here has been laid by work done in the field of context-awareness (esp. [4]) and information logistics (e. g., [8, 7]). Areas that are also strongly related to this work comprise the domains of situation theory [3], and situation calculus [6]. In addition, work done in the field of temporal logic (esp. [1]) has proved to be valuable as basic component of our approach.

The paper is structured as follows: Section 2 first gives an overview of our approach and presents the general idea. With the situation model and its operators, Section 3 describes the instruments necessary to provide the solution outlined. Section 4 finally draws our conclusions and discusses future research perspectives.

2. Supporting Situation-Awareness

Our approach to support situation-awareness is based on two ideas: First, certain types of information directly relate to some situation of a user, for instance, information about a new or changed appointment or – as in our example – information about bad weather conditions. Such information can be used to predict situations a user will get into. Second, the user anticipates or expects certain situations. Comparing these expectations with the predicted situations yields information about unexpected changes the user should be made aware of.

According to [8], a single situation is a time interval associated with a set of invariant properties during that time. The basic idea presented here is illustrated in Figure 1. We distinguish between two strands of situations, the first one representing situations that are predicted or expected by the system as user situations to come up, the second one representing the user’s expectation. More accurately, user’s expectation means what the system assumes the user to know because, for example, the user was earlier informed by the system.

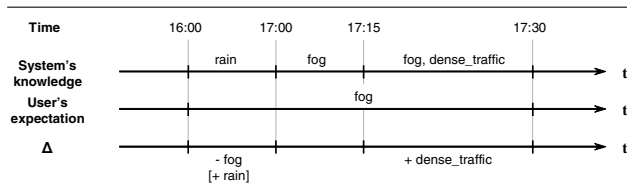


Figure 1. Example of knowledge gap between system’s and user’s situation sequence.

In the example, the user expects fog during his or her trip between four o’clock and half past five in the afternoon. If the system knows about rain occurring between four and five o’clock then the system’s supplementary knowledge is that there will be not fog but rain. This generally leads to a difference Δ between the system and user situation sequence. Δ represents an information gap where any single change in a basic property could potentially be worth a message. To achieve this kind of a demand-oriented information supply we suggest the following basic procedure:

1. Filter all irrelevant characteristics out of both strands of situations. If, for example, the user is interested only in dangerous situations where fog comes along with dense traffic ($\text{‘fog} \wedge \text{dense_traffic’}$) then ‘fog’ and ‘dense_traffic’ , as atomic phenomena, remain in the filtered strand but ‘rain’ is filtered out.
2. Compute the difference Δ between both strands. (Note that such a difference may contain the change

‘+dense_traffic’ even though dense traffic alone does not satisfy the pattern $\text{‘fog} \wedge \text{dense_traffic’}$.)

3. Calculate the set of *relevant* changes using Δ and the user knowledge strand. For example, the condition $\text{‘fog} \wedge \text{dense_traffic’}$ is a relevance pattern. Then ‘-fog’ is relevant if the user formerly expected fog *and* dense traffic. This would result in “no longer danger caused by dense traffic in conjunction with fog” during the dedicated time interval and we also know that the reason is: no longer fog. Analogously, ‘+dense_traffic’ is relevant in a situation when the user formerly expected fog already.
4. For each relevant information unit (the changes from step 3 above):
 - (a) Calculate the time of alert.
 - (b) Inform the user. Optionally give reasons.
 - (c) Update the user’s strand by incorporating the difference.

There are a couple of inherent problems that, by refinement, arise from the procedure above. To show the bandwidth of the information logistics challenges these are mentioned here in brief although not all of them are addressed in this paper.

- The problem of modeling context and situations: This has been tackled in [8]. We assume this understanding of situations and their modeling (see further).
- The problem of comparing situations and situation sequences: We need some formalism to be able to compute Δ as well as to describe the filter semantics indicated in Step 3 above.
- The problem of calculating the time of alert for a relevant message: This includes considerations about utility and acceptance of messages in different situations which means modeling *suitable* and *disturbing* effects of messages. The concept of *situation-based message rating* [7] is applicable at this point though its actual application is not presented here.

In this paper, we focus on a solution for the difference calculation issue, that is, we mainly attend to the second problem. Our solution is based on set-oriented operations over situation characteristics in order to describe Δ , the filtering process, and functions to update the system and user knowledge strands.

3. Computing Information Differences

With the situation model and its operators, this section describes the basic instruments to provide the solution outlined in Section 2. The terms and operations are defined informally here in order to support comprehension of the ap-

proach presented. A more formal definition would go beyond the scope of this paper and is not included here. The situation model was originally described in [8]. We extend this description here for the purpose of situation anticipation and comparison.

We first give informal definitions of the notions of *situation* and *situation sequence*. This is followed by basic relations namely *compatibility*, *consistency*, and *synchrony*, which are introduced because they are needed to define the operators.

Situation. A *situation* s is a triple (t_b, t_e, C) where t_b denotes the begin time, t_e the end time of the situation, and C denotes a set of logical propositions – called *characteristic features* of s – that hold during the interval $[t_b, t_e)$ ¹. Because of the time interval being an inherent part of the situation description the interval-based relations (*Before*, *Meets*, *Overlaps*, *Starts*, *During*, *Finishes*, etc.) defined in [1] can be easily applied to situations and will be used with the same semantics. A characteristic feature is modeled as a simple predicate that is defined over concepts from a taxonomy (called dimension structure). The taxonomy is defined as a directed acyclic graph with a subsume relation. Every predicate is considered to form a dimension of its own, i. e., there are no explicit dependencies between them. The predicate allows simple deductions by utilizing the relationships between concepts of the taxonomy. The model of characteristic features is purposely kept simple, because it first serves as a kind of interface between user and system – both, humans and machines should be able to interpret and comprehend the knowledge contained in a situation’s description. On the other hand this simplicity ensures the determinability of the necessary logical operations.

Situation Sequence. A *situation sequence* is a well ordered set of non-overlapping situations, i. e., every point in time belongs either to 1 or 0 situations of the sequence. The situations in a sequence are ordered by their time intervals.²

Compatibility. Two situations are said to be *compatible* if and only if they do not overlap or their characteristics do not contradict each other. Two situation sequences S_1, S_2 are compatible if and only if for all pairs $s_1 \in S_1, s_2 \in S_2$ s_1 and s_2 are compatible.

Synchrony. Two situations are *synchronous* if and only if they share the same time interval. Two sets of situations S_1 and S_2 are synchronous if and only if for all situations $s_1 \in$

S_1 there exists a situation $s_2 \in S_2$ that is synchronous to s_1 and vice versa.

Consistency. A set of situations is said to be *consistent* if all situations are pairwise compatible.³

Difference. The difference of two situation sequences S_1 and S_2 is the situation sequence describing the “situational knowledge” contained in S_1 that do not appear in S_2 . We write $S_1 \setminus S_2$. Because the concrete definition and results of this operation depend on the representation of situation sequences chosen, Figure 2 first illustrates the results intended for this and the following operations using two simple situation sequences.⁴ The representation required is developed below.

Intersection. The intersection of two situation sequences S_1 and S_2 is the situation sequence describing the “situational knowledge” contained in S_1 as well as in S_2 . We write $S_1 \cap S_2$.

Union. The union of two compatible situation sequences S_1 and S_2 is the situation sequence describing the combined “situational knowledge” of both sequences. We write $S_1 \cup S_2$. Two incompatible sequences can be unified by first removing contradicting knowledge.

Delta. The Delta between two situation sequences S_1 and S_2 describes the overall differences in “situational knowledge” contained in S_1 and S_2 and is defined as the tuple (S^+, S^-) , where S^+ – called add-list, following [5] – is a situation sequence itself and describes those parts of S_1 which do not appear in S_2 , and S^- – called delete-list – describes the parts of S_2 not appearing in S_1 . Thus Delta is defined as $\Delta(S_1, S_2) := (S_1 \setminus S_2, S_2 \setminus S_1)$.

Sequence Representation. In order to produce the intended results (cf. Figure 2) the operations defined above require a certain representation of situation sequences. First, both operands are synchronized, i. e., the original intervals will be split such that both sequences are afterwards synchronous. Every single situation now has a direct pendant from the other operand (provided we use imaginary situations with empty characteristics to represent the gaps within a sequence). Thus prepared the resulting sequence is build element-wise iterating over all corresponding pairs. The respective set-operation (\setminus , \cup , and \cap) is applied to the sets of characteristic features.

1 For formal reasons the end time point does not belong to the situation (cf. [1]). This is necessary to enable a proper definition of situation sequences, but has otherwise no application relevant impact.

2 Every situation sequence can be transformed into a *complete* sequence by introducing so-called *imaginary* situations with empty characteristics to represent possible gaps.

3 Any consistent set of situations could be used to form a situation sequence. Any situation sequence can be represented as a consistent set of situations.

4 The sequences S_1 and S_2 from Figure 2 contain different (and partially contradicting) knowledge about the situations of a person. The labels represent the characteristics and should be interpreted as follows: a = fog, b_1 = dense traffic, and b_2 = normal traffic. Additionally the knowledge ($b_1 \rightarrow \neg b_2$ and vice versa) is supplied by a traffic taxonomy. The gaps mean that there is no situational knowledge available.

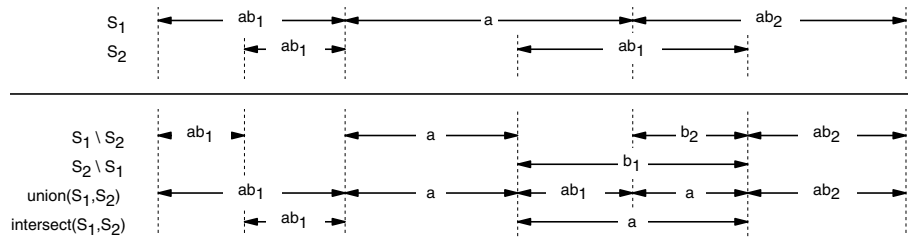


Figure 2. Intended results of operations.

Let S_1 and S_2 be situation sequences and S_{sync_1} and S_{sync_2} the two synchronous sets of situations resulting from synchronizing S_1 and S_2 . The following formula describes the procedure for the union-operation (\cup).

$$S_{sync_{res}} = \{(t_b, t_e, C) \mid \exists (s_1 \in S_{sync_1}, s_2 \in S_{sync_2}) \cdot \\ s_1 = (t_b, t_e, C_1) \wedge \\ s_2 = (t_b, t_e, C_2) \wedge \\ C = CR(C_1 \cup C_2)\}$$

where CR (conflict resolution) removes contradicting facts from a set of propositions. After that, the resulting sequence will be normalized meaning that all neighboring situations of $S_{sync_{res}}$ with identical characteristics will be melted into a single one thus reversing the synchronization process.

4. Conclusion

In this paper, we presented an approach to support a certain class of mobile services, that aim at keeping users up-to-date about information related to their current and upcoming situations. To the best of our knowledge, there is no approach addressing this problem with a general, unified framework. Our concept allows to compare the system knowledge – based itself on facts or on anticipation of events – with a user’s expectations.

The simplicity and genericity of our model allows to support different scenarios and to integrate different information sources based on *characteristics*, *patterns* and *situations*. In addition, only the necessary amount of update information is delivered to the user. Moreover, the model is close to human thinking and permits further enhancements such as a function that simulates *forgetting* by retracting situational knowledge from the user’s strand that has been acquired some time ago.

In order to become an integrated solution, the approach presented has to be supplemented by situation-dependent assessment, delivery, and presentation of information. Furthermore we are currently considering the aspects of fuzziness in describing situations and situation changes. These issues have not been discussed in this paper and are subject to current development.

References

- [1] J. F. Allen and G. Ferguson. Actions and events in interval temporal logic. Technical Report TR521, Computer Science Department, University of Rochester, Rochester, New York 14627, July 1994.
- [2] M. Amberg, S. Figge, and J. Wehrmann. A cooperation model for personalised and situation dependent services in mobile networks. In G. Goos, J. Hartmanis, and J. van Leeuwen, editors, *Advanced Conceptual Modeling Techniques: ER 2002 Workshops, ECDM, MobiMod, IWCMQ, and eCOMO, Tampere, Finland, Revised Papers*, volume 2784 of LNCS, pages 136–146, Berlin/Heidelberg/New York, 2003. Springer Verlag.
- [3] R. Cooper, K. Mukai, and J. Perry, editors. *Situation Theory and its applications*, volume I of *CSLI Lecture Notes No. 22*. CSLI: Center for the Study of Language and Information, Stanford University, California, 1990.
- [4] A. K. Dey. *Providing architectural support for building context-aware applications*. PhD thesis, College of Computing, Georgia Institute of Technology, 2000.
- [5] R. Fikes and N. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. In J. F. Allen, J. Hendler, and A. Tate, editors, *Readings in Planning*, pages 88–97, San Francisco, 1990. Morgan Kaufmann.
- [6] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. Report Memo AI-73, Department of Computer Science, Stanford University, Stanford, California, Nov. 1968.
- [7] U. Meissen, S. Pfennigschmidt, K. Sandkuhl, and T. Wahnfried. Situation-based message rating in information logistics and its applicability in collaboration scenarios. In R. Steinmetz and A. Mauthe, editors, *Proceedings of the 30th Conference on EUROMICRO*, pages 484–491. IEEE Computer Society, 2004.
- [8] U. Meissen, S. Pfennigschmidt, A. Voisard, and T. Wahnfried. Context- and situation-awareness in information logistics. In W. Lindner, M. Mesiti, C. Türker, Y. Tzikzikas, and A. Vakali, editors, *Current Trends in Database Technology – EDBT 2004 Workshops: PhD, DataX, PIM, P2P&DB, and ClustWeb, Heraklion, Greece, Revised Selected Papers*, volume 3268 of LNCS, pages 335–344, Berlin/Heidelberg/New York, 2004. Springer Verlag.
- [9] J. Schiller and A. Voisard, editors. *Location-based Services*. Morgan Kaufmann, San Francisco, 2004.

TagAttack: Ein kontextbasiertes mobiles Mehrbenutzerspiel

Hartmut Ritter
Freie Universität Berlin
Takustr. 9
14195 Berlin
hritter@inf.fu-berlin.de

Rolf Winter
Freie Universität Berlin
Takustr. 9
14195 Berlin
winter@inf.fu-berlin.de

Jochen Schiller
Freie Universität Berlin
Takustr. 9
14195 Berlin
schiller@inf.fu-berlin.de

1. Einleitung

Mit der preiswerten Verfügbarkeit immer leistungsfähigerer Mobilfunkgeräte und insbesondere dem Zusammenwachsen von PDA und Handy wird der Trend zum mobilen Spielen weiter gestärkt. Dabei ist anzunehmen, dass sich ähnlich wie im Bereich der PC-Spiele insbesondere diejenigen Spiele großer Beliebtheit erfreuen werden, die es erlauben, in Gruppen oder einzeln mit oder gegen andere zu spielen.

Mobile Spiele erlauben dabei im Gegensatz zu klassischen PC-Spielen eine enge Interaktion zwischen dem Spieler und seiner Umgebung und damit eine Verbindung von realer und virtueller Welt zu einem umfassenderen Spielerlebnis.

Mit dem Spiel *TagAttack* wurde ein Spiel entworfen und implementiert, das auf einem ad-hoc Netz aufbaut; aber zusätzlich mit tags kommuniziert, die an verschiedenen Punkten in der realen Welt plaziert werden. In diesem Beitrag wird dieses Spiel vorgestellt und insbesondere hinsichtlich der Erfassung und Verwendung von erweiterter Ortsinformation und zusätzlicher Sensorik diskutiert.

2. Verwandte Arbeiten

Der wohl bekannteste Prototyp eines Spieles, das die reale und die virtuelle Welt miteinander verbindet, ist das Spiel *Pirates!* aus dem Jahr 2001 [1]. Bei *Pirates!* ist die Spielidee; dass sich mobile Spieler mit PDAs in einem Raum bewegen; in dem an verschiedenen Stellen per Kurzstreckenfunk erreichbare Module installiert sind. Im Spiel navigieren die Spieler mit einem Schiff durch ein

Meer, die Module, die an verschiedenen Stellen des Raumes sichtbar werden, werden im Spiel als am Horizont auftauchende Inseln visualisiert. Damit liefert *Pirates!* bereits ein sehr gutes Beispiel der Integration von realer und virtueller Welt. Allerdings handelt es sich nur eingeschränkt um ein mobiles Spiel, da zwar die Endgeräte mobil sind, sie aber alle über WLAN direkt mit einem Server verbunden sind, der Teil der festen Infrastruktur ist.

Viele spätere Ansätze basieren eher auf dem Ansatz der *augmented reality*, z. B. *Computer-augmented tabletop games* [2] oder *ARQuake* [3].

Das vorgestellte Spiel *TagAttack* ähnelt von dem grundlegenden Ansatz der ortsbezogenen Spielidee dem Spiel *Pirates!*. Allerdings handelt es sich bei *TagAttack* um ein völlig serverloses Spiel. Im Folgenden werden zunächst Spielidee und Architektur des Spieles *TagAttack* vorgestellt und im Anschluss daran werden die geplanten Weiterentwicklungen diskutiert.

3. Spielidee

Ziel des Spieles *TagAttack* ist es, die Mehrheit der in der Umgebung verteilten tags für die eigene Gruppe zu erobern und zu halten. Sind alle tags erobert, endet das Spiel und diejenige Gruppe gewinnt, die die meisten tags für sich verbuchen kann.

Der Startbildschirm des Spieles ist in Abbildung 1 zu sehen. Er gibt den Grundriss des Innenhofs zwischen dem Informatikgebäude und dem Nachbargebäude der FU Berlin wieder mit charakteristischen Elementen wie den Bäumen und

der Kunstinstallation in der Mitte. Auf der rechten Seite sind die Spielergruppen und ihre momentan aktiven Mitglieder sichtbar.

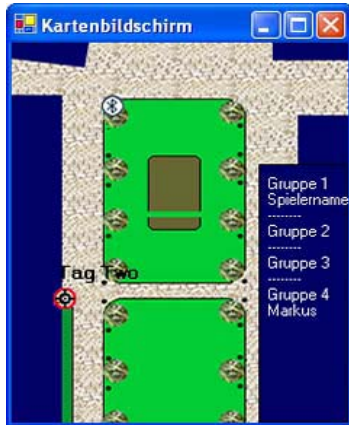


Abbildung 1: Startbildschirm des Spiels TagAttack

Die runden Markierungen kennzeichnen die auf dem Spielfeld verteilten tags. Der Benutzer kann nur mit tags interagieren, die sich in der direkten Funkreichweite befinden, der Status weiter entfernter tags wird ihm aber ebenfalls mitgeteilt. Wie in Abbildung 2 dargestellt, kann ein tag vier verschiedene Zustände annehmen: Neutral, wird angegriffen, freundlich, feindlich.

-  neutral
-  wird angegriffen
-  freundlich
-  feindlich

Abbildung 2: Zustände eines tags

Das Erobern eines tags funktioniert wie folgt: Bei neutralen tags gehört ein tag zunächst derjenigen Gruppe, deren Mitglied als erster in der Funkreichweite des tags war und ihn übernimmt. Das Übernehmen eines tags setzt also immer physikalische Nähe in der realen Welt voraus. Ist ein tag von einer Gruppe übernommen worden, so kann diese Gruppe zum Schutz des tags vor den anderen Gruppen eines von drei Elementen Feuer, Wasser oder Eis auf dem tag plazieren. Ähnlich wie bei dem bekannten Spiel „Papier, Stein, Schere“ ist eine Ordnung festgelegt, in der diese

Elemente zueinander stehen. Diese Reihenfolge ist in Abbildung 3 dargestellt.



Abbildung 3: Reihenfolge der Elemente

Ein Angreifer einer anderen Gruppe muss sich zunächst in die Nähe des für ihn feindlichen tags begeben. Dann kann er auf einem nur dann verfügbaren „Kampfbildschirm“ (s. Abbildung 4) ein Element wählen und gegen das auf dem tag hinterlegte Element antreten. Entsprechend der Reihenfolge der gewählten Elemente verbleibt der tag bei der bisherigen Gruppe oder wechselt den Besitzer.



Abbildung 4: Kampfbildschirm für Übernahmeversuche

Allerdings erfährt die Gruppe, die den tag zum Zeitpunkt des Übernahmeversuchs besitzt, von dem Übernahmeversuch und kann ihn verhindern, indem sie sich seinerseits mit dem tag über Funk verbindet und ihn „verteidigt“. Diese Verteidigung setzt aber die physikalische Nähe voraus. Die virtuelle Verteidigung des tags ist also nur bei realer Nähe zum tag möglich.

4. Realisierung

Das Spiel wurde vollständig in der Programmiersprache C# implementiert und ist auf Windows PDAs und Windows smartphones mit touch-

screen spielbar (z.B. auf dem MDA/VPA), aber auch unter Windows XP.

Als tags wurden bis zu acht selbst entwickelte Bluetooth-Module eingesetzt, von denen eines in Abbildung 5 dargestellt ist.



Abbildung 5: Bluetooth tag

Durch den Einsatz von Bluetooth mit seiner geringen Reichweite von ca. 10 m wurde die Ortsbezogenheit des Spiels erreicht. Die tags speichern die Zugehörigkeit zu einer Gruppe und das Element, das auf Ihnen gesetzt wurde. Sie bieten aber darüber hinaus noch weitere Möglichkeiten bis hin zur Anbindung von Sensorik, die weiter unten diskutiert werden.

Die Kommunikation zwischen den Teilnehmern des Spiels erfolgt über WLAN im ad-hoc modus. Dazu wurde eine eigene Implementierung des ad-hoc Routingprotokolls AODV [4] für Windows mobile vorgenommen. Auch diese Implementierung erfolgte in C# und ist damit portabel zwischen Windows PDAs und PCs unter Windows XP.

Problematisch an dem Einsatz von AODV ist, dass es ohne Erweiterungen zunächst keine multicast-Unterstützung bietet. Stattdessen wird zur Zeit die Benachrichtigung aller Mitglieder einer Gruppe über einen netzwerkweiten broadcast realisiert, der von den Spielinstanzen, die nicht zur angesprochenen Gruppe gehören, ignoriert wird. In Anlehnung an [5], [6] wird Multicast-Unterstützung für AODV momentan in die Implementierung integriert.

5. Weiterentwicklung

5.1 Einbeziehung von Sensordaten

Das Spiel *TagAttack* ist insofern kontextbasiert, als dass die räumliche Nähe, die für das Spiel entscheidend ist, als wesentlicher Kontext des

Handelns der virtuellen bzw. realen Charaktere angesehen werden kann. Denkbar ist jedoch eine viel weitergehende Einbeziehung der realen Umgebung in die virtuelle Welt. So wäre es denkbar, die Helligkeit der realen Umgebung oder beispielsweise auch deren Temperatur einzubeziehen. Im einfachsten Fall könnte bei einer hohen Temperatur oder bei starker Helligkeit das als Spielelement eingesetzte Eis schneller schmelzen oder das Wasser verdunsten.

Die technische Realisierung der Bluetooth tags erlaubt die einfache Erfassung solcher Umgebungsdaten, indem die tags über einen Zweidrahtbus mit Sensorknoten verbunden werden können. Das würde es sogar erlauben, die tags als Zugangspunkte zu einem verteilten Sensornetz aufzufassen, das sich je nach räumlicher Ausdehnung auf mehrere tags erstrecken kann. Damit könnten beispielsweise auch Bewegungen anderer Mitspieler, die gar nicht in direkter Sicht- und Funkreichweite eines tags bzw. menschlichen Spielers sind, erfasst und als Kontext an das Spiel übermittelt werden. In Abbildung 6 ist ein an der FU Berlin entwickelter ScatterWeb Sensorknoten [7] abgebildet, der über I2C-Bus mit dem Bluetooth tag verbunden wurde. Der Sensorknoten erlaubt den Aufbau von multihop-Sensornetzen und die Erfassung von Temperatur, Helligkeit, Vibration, Lautstärke und Bewegung.



Abbildung 6: ScatterWeb Sensorknoten

5.2 Annotation der Umgebung als Kontext

Die verteilten Sensorknoten ermöglichen nicht nur die Erfassung von Sensordaten und damit die ortsabhängige Anpassung des Spieles an die Umgebung, sondern auch umgekehrt die Erweiterung des jeweiligen Ortes um Daten der virtuellen Welt. So ist es beispielsweise möglich, die tags

und Sensoren dahingehend zu erweitern, dass sie Daten wie beispielsweise den Spielstand speichern oder dass sie Nachrichten speichern, die einzelne Spieler an Mitspieler oder Gegner hinterlassen. Ebenso ist es leicht vorstellbar, dass im Spiel ortsabhängig verschiedene komplexere Aufgaben zu lösen sind, deren Zwischenergebnisse auf den lokalen tags abgelegt werden und Mitspielern, eventuell aber auch Gegnern zur Verfügung stehen. Damit umfasst die Ortsinformation nicht nur die geographischen Daten, sondern vor allem auch die Bedeutung dieses Ortes und des bisherigen Geschehens an diesem Ort für das Spiel.

Wie in Abbildung 7 angedeutet könnte ein Beispiel für eine solche mit zusätzlicher Information angereicherte Lokation eine U-Bahn-Station sein, die Teil eines über das gesamte U-Bahn-Netz einer Stadt verteilten Spieles ist. Während der Name und Ort einer U-Bahn Station für einen menschlichen Spieler leicht erkennbar sind, könnte über die in der Station verteilten tags ortsbezogene Information gespeichert werden, die sich auf das Spiel bezieht und beispielsweise das Abschneiden der vorigen Gruppe am gleichen Ort speichert und dem neu hinzugekommenen Spieler mitteilt. Sind innerhalb des Spiels Aufgaben an einem Ort zu erledigen, die nicht von einer Gruppe alleine, sondern nur von mehreren Gruppen gemeinsam erledigt werden können, wird durch die Information über die Vorarbeit einer anderen Gruppe die Bedeutung der zunächst rein topographischen Ortsinformation grundlegend erweitert.

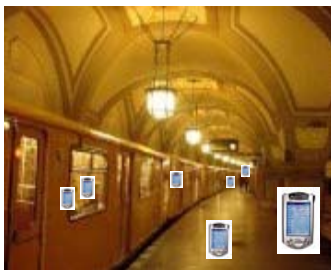


Abbildung 7: U-Bahn-Station als tpyische Lokation

6. Zusammenfassung

Ausgehend von der Beobachtung, dass mobile Spiele voraussichtlich weiter an Bedeutung gewinnen werden, wurden Konzept und Implementierung des mobilen Spieles *TagAttack* beschrieben, das Ortsinformationen als Kontext für die Spielewelt nutzt, aber auch die an einem Ort gespeicherte Information aktiv beeinflusst.

Davon ausgehend wurde diskutiert, wie eine Verbindung zwischen Lokationsdaten und weiteren Sensordaten erfolgen kann. Außerdem wurde beschrieben, dass aus Sicht der Autoren Lokationsinformation nicht nur die reine physikalische Ortsinformation beinhalten muss, sondern auf die Geschichte eines Ortes und seiner Bedeutung im Rahmen einer Anwendung, im gewählten Beispiel eines Spiels, sinnvoll erweitert werden kann.

7. Referenzen

- [1] Björk, Falk, Hansson, Ljungstrand: Pirates! using the physical world as a game board. In Proceeding of Interact 2001 Conference on Human-Computer Interaction, Tokio, Japan, Juli 2001
- [2] Magerkurth, Engelke, Memisoglu: Augmenting the virtual domain with physical and social elements. In Proceedings of the International Conference on Advancements in Computer Entertainment Technology, Singapur, Juni 2004
- [3] Piekarski, Thomas: ARQuake: The outdoors augmented reality system. Communications of the ACM, 45 (1), Januar 2002
- [4] Perkin, Royer: Ad-hoc On-Demand Distance Vector Routing. Proceeding of the 2nd IEEE Workshop on mobile computing systems and applications, New Orleans, Januar 1999.
- [5] Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol. In Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, Seattle, August 1999
- [6] Royer, Perkins: Multicast Ad-hoc On-Demand Distance Vector (MAODV). IETF Internet Draft draft-ietf-manet-maodv-oo.txt, 2000
- [7] ScatterWeb Projekt der FU Berlin. Information unter <http://scatterweb.mi.fu-berlin.de>

Improvement of Accuracy for GPS

Stephan Kopf, Thomas King, Wolfgang Effelsberg
Praktische Informatik IV
University of Mannheim
{kopf|king|effelsberg}@informatik.uni-mannheim.de

Abstract: We analyze typical positioning errors of the GPS system and present an approach to improve the accuracy of position information obtained from standard GPS receivers. Based on a basic movement model, we pre-calculate a position estimate and compare it with the position delivered by the GPS receiver. This approach is validated by means of simulations and real world measurements and shows a reduction of the positioning errors of ten percent.

1 Introduction

The *Global Positioning System* (GPS) is the most relevant technology to identify the position in outdoor scenarios [1,2]. An alternative European technology (*GALILEO* [3]) is expected to be operational in a few years. GPS is used for the navigation of planes and ships, land surveys and to analyze changes of the earth's crust, but it is also available in daily life through mobile phones, PDAs or car navigation systems. A direct line of sight to GPS satellites and a positioning error of fifteen meters or more are the major disadvantages of this system.

Figure 1 depicts the changes of position data that were recorded by a stationary GPS receiver. More than 6000 data sets were stored within a few hours during a slightly cloudy day on the roof of a building. These values are plotted in a diagram [4] and show a maximum positioning error of fifteen meters.

Differential GPS (DGPS) [5,6] improves the accuracy of positions significantly, and hence the expected positioning error drops below five meters. A reference station on the surface of the earth analyzes the GPS signals and transmits correction information, which is used by DGPS receivers to adjust the measurements. Compared to standard GPS receivers, DGPS systems are more expensive and larger due to additional antennas.

The goal of this paper is to present an approach to improve the position information with standard GPS receivers. The idea is to identify and filter single positioning errors and improve the position information by interpolating the current position with previous position values.

This paper is structured as follows. In the following Section, typical positioning errors of a GPS system are analyzed. In Section 3, values for the accuracy of position are derived from the GPS receiver, which define the weights for the interpolation. This approach is validated in Section 4 with several simulations and real world measurements. The paper finishes with a conclusion and an outlook.

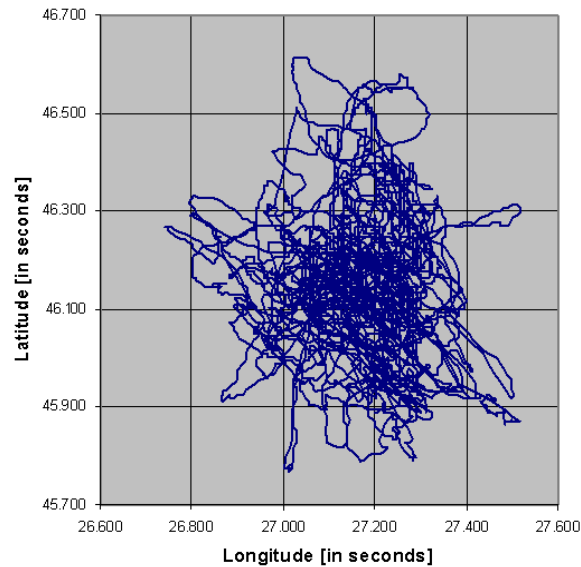


Figure 1: Position information of a stationary GPS receiver

2 Positioning Errors

In this section, we analyze the typical positioning errors of GPS systems. These errors may be caused by satellites, the atmosphere, the GPS receiver or the environment where the GPS receiver is located [7,8].

Satellite orbits: Although the satellite orbits are usually very precise, small changes are possible due to the gravity of the sun and moon. Temporary errors caused by the orbit may generate positioning errors of several meters.

Inaccurate calculation based on satellite positions: It is much harder to determine a precise position if the available satellites are close to each other. Very small calculation or measurement errors can lead to significant positioning errors.

Changes in the atmosphere: The velocity of propagation of the GPS signal is reduced in the ionosphere and troposphere. Solar wind influences the ionosphere and can cause positioning errors up to 5 meters. Current weather conditions and the amount of water in the troposphere generate additional errors.

Quality of GPS receivers: The constructions of the antennas, the quality of the components and the algorithms which calculate the position have a great impact on the accuracy of the position.

Errors based on timers: Even very small differences of the timers in the satellites or the receivers may cause positioning errors of several meters.

Assumption of a wrong altitude: If only three satellites are in range of a GPS receiver it determines its position with the 2D position-fix approach which assumes a fixed altitude. Especially in mountain regions, the assumption of the altitude is problematic and may cause positioning errors of several hundred meters. The 3D position-fix approach is used if data from at least four satellites is available.

Reflection of GPS signals: The GPS signals can be reflected from the environment (e.g., buildings) and the distance between the satellites and the GPS receiver increases compared to a direct line of sight. An error of several meters may occur.

If different positioning errors are aggregated a deviation from the correct position of fifteen meters or more is possible. The recorded positions in Figure 1 of the stationary GPS receiver confirm this deviation. DGPS reduces the effects of the ionosphere, the changes of the orbits and timer errors, so that the positioning error is usually reduced to less than five meters.

3 Approach

Usually, GPS receivers provide the position information (longitude, latitude, altitude) in a standardized format [9,10]. This format may contain additional information like date, time, available satellites, the technology (GPS or DGPS), as well as a value called *dilution of precision* (DOP) which measures the accuracy of the position. The DOP value makes a prediction about the expected positioning error [11]. With a probability of 95 percent the real position is within a certain range (based on the DOP value) of the GPS position.

We assume a high correlation of the position, direction and speed of the GPS receiver within a short time interval. An *estimated position* is calculated based on a linear regression model from the previous position values. We compare the position of the GPS receiver (*GPS position*) with the estimated position, weight both positions based on the current accuracy (DOP) and get a final *interpolated position*.

The weights for the estimated and GPS position are based on the quality measurement values. If the DOP indicates very precise position information the GPS position is weighted by 75 percent, and it drops to 25 percent for low values. Additionally, we reject the estimated values if the distance between the GPS and estimated position exceeds a predefined threshold. This rejection usually corresponds with significant changes of the direction or speed of the GPS receivers.

The quality of the interpolated values is significantly influenced by the number of changes in direction or speed. A major factor is the current speed because even a slight modification of the direction has a large impact on the absolute position of fast moving GPS receivers. Latest GPS receivers calculate position information twice or even four times a second and hence make our interpolation approach applicable for higher velocities.

In our approach, the current direction and speed is estimated from previous positions. The algorithm adapts much faster if only two values are used for the estimation. On the other hand, if a constant speed and direction is expected additional values will increase the accuracy. In our implementation, the number of selected positions (2-5) depends on the refresh rate and speed of the GPS receiver.

4 Experimental Results

We have conducted a large number of simulations and measured positions in real world scenarios to evaluate our proposed algorithm. We assume that very fast changes of the direction are not possible (a change of direction within one second should not exceed 60 degrees). The simulated GPS positions are influenced based on given DOP values.

In our simulation, we analyzed the effect of the interpolation with the standard GPS approach first and assume a deviation of fifteen meters or less [12]. In low-speed scenarios (less than 50 km/h), the interpolation improved the accuracy of the position by 0.45 meters on the average (more than ten percent), but the accuracy gain drops with increasing speed. In high-speed scenarios (75-120 km/h or faster) the advantage of the interpolation is reversed to a disadvantage: the interpolation reduces the accuracy and the data should not be interpolated anymore.

A different probability distribution is used to analyze the behavior for DGPS. In this case, the positioning error is usually below 5 meters, even slight direction changes may cause obvious positioning errors and only a very low speed is advantageous.

If no valid GPS signal is available (e.g., in tunnels or between high buildings) the interpolation is mandatory.

We validated our simulation results in real world measurements and used the following setup: Ten points were marked in a distance of 5 meters on a straight line of an open field. Three additional measuring points with a distance of five meters and an angle of 30 degrees were tagged, followed by ten additional points on a straight line. The weather condition was fine

which caused low DOP values between two and three (high quality). Twelve runs were performed and analyzed with a standard GPS receiver. The position could be improved in more than 90 percent, and the average positioning error decreases by 0.7 meters for each measurement point. These results show that our simulation model and the simulation results are valid.

5 Conclusion and Outlook

Especially in scenarios with low speed and standard GPS receivers the interpolation of the position improves the accuracy of the position significantly. This advantage drops with higher speed and more precise GPS receivers. On the other hand, if the GPS device provides data more often than once per second the interpolation is even applicable for higher velocities.

The interpolation is quite useful in tunnels and other places without direct line of sight to satellites. This approach provides position information until the next data record is available. The relevance of the interpolated value should also increase if the quality of the position accuracy decreases significantly. With very low additional computational costs an up to ten percent improvement of the accuracy is possible.

Acknowledgements

This work was supported by grants from the Ministry of Science, Research and the Arts of Baden-Württemberg and the Landesstiftung Baden-Württemberg.

References

- [1] B. Hofmann-Wellenhof, H. Lichtenegger and J. Collins. GPS, Theory and Practice, 1997.
- [2] M. Bauer. Vermessung und Ortung mit Satelliten. GPS und andere satellitengestützte Navigationssysteme (5th edition). Wichmann Verlag, Heidelberg, 2003.
- [3] GALILEO – European Satellite Navigation System. http://europa.eu.int/comm/dgs/energy_transport/galileo/index_en.htm, (last checked: 04/29/2005).
- [4] D. Hurrelmann. Evaluation von GPS-Empfängern für den Einsatz mit Handhelds und Implementierung eines Positionsbestimmungsdienstes auf einem Linux-basierten Handheld, Studienarbeit, Universität Mannheim, Juli 2003.
- [5] EGNOS und WAAS - Modernste D-GPS Satellitensysteme, http://www.environmental-studies.de/Teilflächenbewirtschaftung/EGNOS_WAAS/3.html, (last checked: 04/29/2005).
- [6] Federal Aviation Administration - Satellite Navigation. gps.faa.gov, (last checked: 04/29/2005).
- [7] GPS - Infos, <http://www.kowoma.de/gps/index.htm>, (last checked: 04/29/2005).
- [8] Trimble - All about GPS, <http://www.trimble.com/gps/>, (last checked: 04/29/2005).
- [9] NMEA 0183, <http://www.kh-gps.de/nmea-faq.htm>, (last checked: 04/29/2005).
- [10] The national marine electronics association (NMEA), <http://www.nmea.org/>, (last checked: 04/29/2005).
- [11] R. J. Milliken and C. J. Zoller. Principle of Operation of NAVSTAR and System Characteristics, Navigation, Vol. I, Washington, 1980.
- [12] S. Graber. Genauigkeitssteigerung der GPS-Positionsbestimmung durch Lineare Regression, Studienarbeit, Universität Mannheim, März 2005.

A Location System based on Sensor Fusion: Research Areas and Software Architecture

Thomas King, Stephan Kopf, Wolfgang Effelsberg
Praktische Informatik IV
University of Mannheim
{king,kopf,effelsberg}@informatik.uni-mannheim.de

Abstract: Information about the current position of a mobile device is required by context-aware applications. An inexpensive, pervasive, all-purpose location system is not yet available. This paper discusses approaches of different location systems and provides an insight in our ongoing research work, in particular for indoor locations.

1 Introduction

Position has long been identified as key information required for context-aware and location-based applications. In recent years a significant amount of research effort has aimed at inventing an all-purpose location system. Unfortunately, only the Global Positioning System (GPS) has achieved public acceptance. GPS is an outdoor location system. It is easy to use, free of charge, GPS receivers are relatively inexpensive, and it provides adequate accuracy for most outdoor applications. However, an inherent flaw of GPS is that it does not work indoors where people of developed countries spend most of their time.

Indoor location systems are still a research issue today. Many of the research systems do not scale well because they require an expensive infrastructure or a complex calibration process. Only few indoor location systems offer an accuracy of a few meters. For a large class of location-based applications, such as finding a book in a bookshelf, this accuracy is still insufficient.

New research directions in indoor location systems try to exploit the existence of more than one sensor to determine the location of a mobile device. This technique is called sensor fusion, and first publications in this area are promising. Another interesting research area is how to integrate Wireless LAN and Bluetooth access points as sensors into location systems. Using readily available infrastructure is a great benefit, especially Wireless LAN and Bluetooth show nearly complete coverage of areas where people of industrialized countries live. Additionally, other sensors that could be employed to sense the location of a mobile device might be integrated into a location system based on sensor fusion.

This paper is organized as follows. In Section 2, we discuss related work and present current research areas. Section 3 outlines interesting location sensors, location determination algorithms and sensor fusion algorithms. In Section 4 we provide an overview of the software architecture we are using to implement our research ideas in the area of location systems. Finally, we conclude the paper in Section 5.

2 Research Areas and Related Work

Indoor location systems have been an active research area since the Active Badge [1] project in 1992. Since then, several different indoor location systems have been invented as a counterpart of GPS to fill the gap in indoor positioning systems. These indoor location systems offer various capabilities but also different disadvantages [2] [3]. Most of them require additional infrastructure to be operational. In this paper we focus on location systems

and location sensors that are available as commodity products, as all the specialized hardware-based approaches are too expensive and hence do not scale well.

Only a few all-purpose location systems utilizing one technology to sense the position of a mobile device are available [4]. One of the recently proposed location systems called Place Lab [5] supports different sensor technologies and until now sensors based on Groupe Spécial Mobile (GSM) cell phones and wireless LAN technology are evaluated.

If a location system is split into coherent pieces, two major parts can be identified: sensors to sense a certain physical attribute and an algorithm to compute a position from the values delivered by the sensor. Recently published location systems add as a third part a sensor fusion algorithm that combines position estimates obtained from different location determination algorithms and different sensors. The idea is that the results generated by a sensor fusion algorithm are more precise than a position estimate provided by one kind of sensors.

3 Sensors and Location Determination Algorithms

Our goal is to integrate multiple sensors into our location system to determine the location of a mobile device. We assume that the mobile device in question is equipped with the appropriate sensors. As an outdoor sensor we have chosen GPS, Wireless LAN and Bluetooth technology were selected as indoor sensors.

3.1 Global Positioning System

The well-known Global Positioning System (GPS) was designed for the usage of the U.S. military to compute the position, speed and time of a GPS receiver [6]. The GPS infrastructure is funded and controlled by the U.S. Department of Defense, and is fully operational since 1995 as a worldwide, satellite-based navigation system. The nominal GPS Operational Constellation consists of 24 satellites which orbit the earth at a height of approximately 20,000 km. Usually, more than 24 satellites are operational as new satellites are launched to replace old ones. The Standard Positioning Service of GPS provides an accuracy of a few hundred meters for civil usage.

To improve the accuracy of GPS a technology called differential GPS was developed. The GPS signal is usually blurred by noise, bias, and blunders. To calculate correction data for mobile receivers positioning errors are measured at well-known locations, and the correction data is spread via satellites, FM sub-carrier broadcasts and others. This correction data is used by mobile receivers to adjust their own position measurements. With this improvement a positioning accuracy up to one meter can be achieved. A clear line of sight to at least four satellites is required to calculate the position (longitude, latitude, and altitude) and time of a GPS receiver. GPS signals can be blocked by obstacles such as walls, foliage, and clouds. For example, "urban canyons" formed by skyscrapers prevent GPS receivers from receiving signals emitted by GPS satellites.

Within our research work we integrate GPS as one sensor to our sensor fusion based location system. To create an all-purpose location system an outdoor sensor is needed.

3.2 Wireless LAN

An early approach of Wireless LAN technology as a sensor for location systems was published by Bahl and Padmanabhan during their work for the RADAR project [7]. In their approach mobile devices continually measure the signal strength of beacons that are periodically emanated by fixed access points. To obtain usable results each point of the terrain must be covered by at least two access points. Before the system can be used for user tracking, a database with signal strength values at well-known locations must be built. The location of a mobile device is estimated by comparing the strength of the last measured

sample with values stored in the database. The authors propose a single nearest neighbour in signal space, a k-nearest neighbours and a Viterbi-like algorithm to estimate the location of a mobile device. Real world indoor evaluations have shown that a median error distance of 2.37 meter can be achieved [8].

In [9] the authors are mainly interested in location determination algorithms that perform well in case of minimal calibration data. They used a technology called Place Lab as a metropolitan-scale Wireless LAN location system. The idea behind Place Lab is to utilize the huge databases created by the war-driving community to estimate the locations of access points. The term war-driving was firstly defined by Pete Shipley [10] and means driving around looking for wireless networks. One of the largest war-driving databases available free of charge contains more than 2.26 million entries of access points worldwide [11]. Place Lab utilizes the positions of access points in communication range of a mobile device to infer the location of the device. The benefit of this approach is that it spreads the costs and effort to build the database over a large community. A Bayesian algorithm proposed by the authors has shown a median error distance in a real world outdoor evaluation of 10-30 meters.

Based on the promising results of Wireless LAN technology as a sensor for location systems, we investigate how war-driving databases can be employed to create a highly accurate indoor location system. In addition to that, we integrate the Place Lab technology as an outdoor sensor to improve the accuracy.

3.3 Bluetooth

Bluetooth was developed in 1999 as a cable replacement technology. The usual communication range of Bluetooth devices is only a few meters. However, it is an interesting technology for location systems because it might provide fine-grained position estimates. In [12] the authors investigated the usability of Bluetooth for location systems. Depending on the class of Bluetooth devices used and the velocity of the moving device, the amount of time needed to find surrounding devices can be higher than the time the devices are in communication range of each other.

We believe it is the right time to methodically investigate how Bluetooth can be used for location systems. Especially, since the Bluetooth Class 1 and Enhanced Data Rate specifications are available. Communication range is increased to 100 meters by the former specification, and the latter specification enhances the data rate to 2.2 MBit/s.

3.4 Sensor Fusion

In practice, a location system may consult many different location sensors to determine the location of a mobile device. In many cases, though, sensor data is noisy and misleadingly influenced by the environment. Even worse, position estimates from different sensors may conflict with each other.

Sensor fusion is a promising technique to solve position estimate conflicts reported by different location sensors. Additionally, this technique may improve the position accuracy. In [13] a Bayesian network algorithm shows how to minimize the number of queried sensors while simultaneously achieving a certain level of accuracy. Unfortunately, in the evaluation process only homogenous sensors are used. In contrast, the Place Lab project utilized Wireless LAN access points, GSM mobile phone cell towers and a sensor fusion algorithm based on particle filters to improve the accuracy and coverage of their location system significantly [11].

4 Software Architecture

The software architecture of our location system is divided into two parts: a small piece of native code and a Java 2 Micro Edition (J2ME) part. Sensor data such as signal strength values are usually not directly accessible through Java because these parameters are stored in an operation system dependent way and managed inside the hardware drivers. Therefore, a small native code library gathers sensor data and offers the values to the Java applications. Location determination and sensor fusion algorithms are implemented using Java to exploit the write-once-run-everywhere capabilities of Java. We have chosen J2ME because for most Laptops, PDAs, Smart phones and mobile phones a J2ME virtual machine is available. The native code implementation and the Java application are tied together with the Java Native Interface.

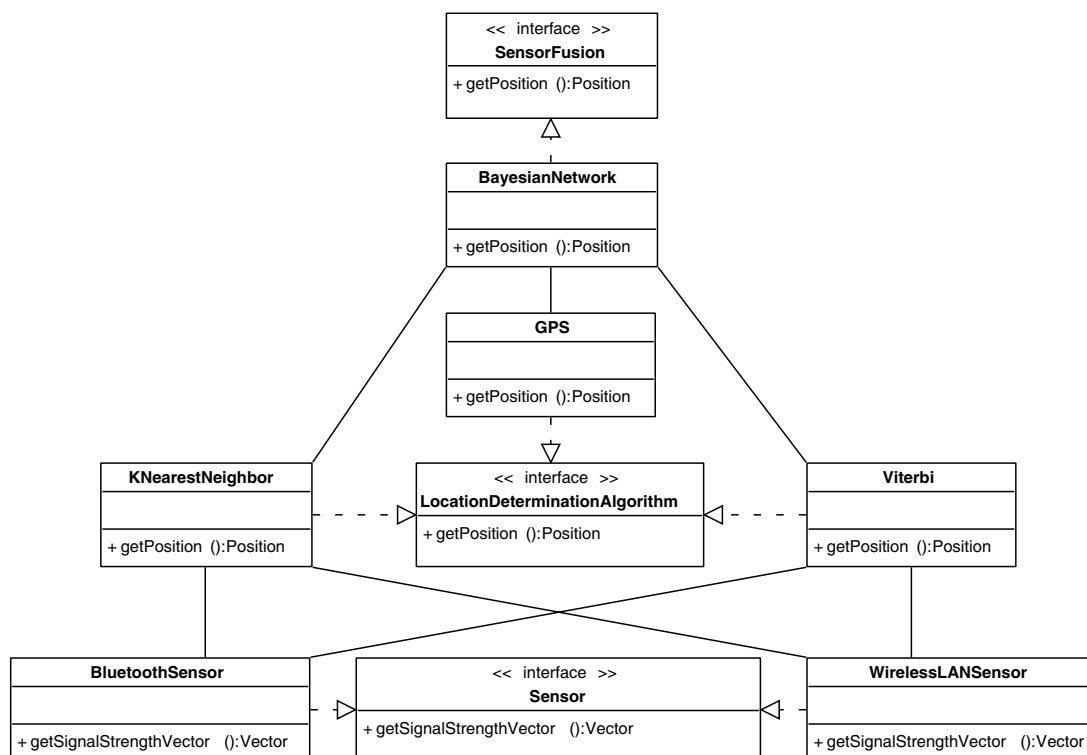


Figure 1: Software architecture

Figure 1 shows the software architecture as an UML class diagram. The three elements of a location system, as identified in Section 3, divide also the software architecture. The sensors, location determination algorithms and sensor fusion algorithms are represented by the *Sensor*, *LocationDeterminationAlgorithm*, or *SensorFusion* interface, respectively. Each algorithm and sensor implementation has to realize the interface to which it belongs. The interfaces define generalized methods that can be used for a standardized data exchange. For instance, the method *getPosition* is defined in the *LocationDeterminationAlgorithm* interface and implemented by location determination algorithms, such as *KNearestNeighbour*, *Viterbi* and *GPS*.

5 Conclusions

Similar to other wireless technologies such as FM radio, satellite television, or mobile phones, an all-purpose location system has to provide full coverage in areas where people live. A

distinction between out- and indoor is inappropriate for its future use, especially if location-based and context-aware applications raise the markets as forecasted by various research institutes.

In this paper, we have presented promising technologies for position determination and we have shown that an all-purpose location system based on sensor fusion will be feasible. Currently, we are working on an implementation of such a location system as part of our on-going Mobile Business project.

Acknowledgements

This work was supported by grants from the Ministry of Science, Research and the Arts of Baden-Württemberg and the Landesstiftung Baden-Württemberg.

References

- [1] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10 (1): 91-102, January 1992.
- [2] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *Proceedings of the Sixth Annual ACM International Conference on Mobile Computing and Networking*, Boston, MA, USA: 32-43, August 2000.
- [3] R. Orr and G. Abowd. The smart floor: a mechanism for natural user identification and tracking. In *Proceedings of the 2000 Conference on Human Factors in Computing Systems*, The Hague, Netherlands: 275-276, April 2000.
- [4] J. Krumm, G. Cermak, and E. Horvitz. RightSPOT: a Novel Sense of Location for a Smart Personal Object. In *Proceeding of UbiComp 2003*, Seattle, WA, USA: 36-43, October 2003.
- [5] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place Lab: Device Positioning Using Radio Beacons in the Wild. Technical Report IRS-TR-04-016, Intel Research, October 2004.
- [6] E. Kaplan. *Understanding GPS: Principles and Applications*. Artech House, March 1996.
- [7] P. Bahl and V. Padmanabhan. RADAR: An In-Building RF-based User Location and Tracking System. *Proceedings of IEEE Infocom 2000*, Tel-Aviv, Israel: 775-784, March 2000.
- [8] P. Bahl, V. Padmanabhan and A. Balachandran. Enhancements to the RADAR User Location and Tracking System. Technical Report MSR-TR-2000-12, Microsoft Research, February 2000.
- [9] Y. Cheng, Y. Chawathe, A. LaMarca and J. Krumm. Accuracy Characterization for Metropolitan-scale Wi-Fi Localization. Technical Report IRS-TR-05-003, Intel Research, January 2005.
- [10] WarDriving.com, <http://www.wardriving.com/about.php>, last visited: 04/2005.
- [11] Wireless Geographic Logging Engine – Plotting WiFi on Maps, <http://www.wigle.net>, last visited: 04/2005.
- [12] Y. Yoneyama, M. Makino, and S. Shinoda. A Location System with Bluetooth. *Proceedings of the 2002 International Technical Conference on Circuits/Systems, Computers and Communications*, Phuket, Thailand, July 2002.
- [13] P. Castro and R. Muntz. Managing Context Data for Smart Spaces. *IEEE Personal Communications*, 7 (5): 44-46, October 2000.

Modelling Location Sensors

Jörg Roth
University of Hagen
58084 Hagen
Germany
Joerg.Roth@Fernuni-hagen.de

1 Introduction

Location-aware applications rely on location sensor input to provide location-dependent output. Many frameworks assume ideal location sensors with high accuracy, availability and coverage and ignore potential measuring costs. Real location sensors, however, have certain limitations concerning location input and lead to certain costs (e.g. drain the battery or lead to monetary costs). In this paper, we present a framework to model location sensors. The framework is included into the Nimbus platform for location-based services [9].

2 Location Sensor Models

A typical domain for modelling location sensors is robotics. Mobile robots have to determine their location with the help of sensors such as distance sensors, odometers, or visual sensors. The concept of sensor data fusion is strongly related to location sensor models: to increase availability, coverage and accuracy, the output of multiple location sensors can be combined to a single output. There are three major ways to model location sensor output: *points*, *position probabilities* and *areas*.

Points in space: The simplest model of a location is a single point in space. Ignoring accuracy and uncertainty issues, a measurement could be described by a single coordinate such as N51°22.579/E7°29.615/169 m. Although it is very unlikely that the measured and the true location are identical, many applications simply take the measured value for further processing without considering the measurement error. As some positioning systems have a high accuracy, this approach is often suitable.

Position probabilities: More advanced approaches model location output by a statistical distribution of measurements. For two dimensions we get a probability distribution for a certain measurement. For a small grid element (e.g. of 10 m x 10 m), the distribution expresses the probability for a user to reside in this grid element. In this context we talk about the position probability of a mobile user. Statistical methods based on Bayes' rule, the Kalman Filter or Position Probability Grids can be used to fuse multiple sensor input.

Areas: As a last model for location output, areas could be used. An area describes the set of possible user locations when a certain measurement is performed. In particular, location output based on cell of origin paradigm (e.g. GSM cell positioning) can easily be described by areas. The area model is suitable to represent the locations as we can easily specify whether a certain location belongs to a cell or not. The area model could be viewed as a simple statistical representation where the area border separates two regions with a uniform distributed position probability – inside the area the sum of probabilities is 1, outside it is 0.

3 The Nimbus Location Sensor Model

To model the different attributes of location sensors, we introduce the *Virtual Positioning System (VPS)* which hides the specific details of underlying real positioning systems. With this component, the framework can use arbitrary positioning systems. These systems range from satellite positioning systems (e.g., GPS), positioning with cell phone networks (e.g., GSM) to indoor positioning systems (e.g. our PalmSpot system [9]). In addition, a simulation tool can simulate sensor information and can be used to set up realistic test scenarios without going to the respective locations.

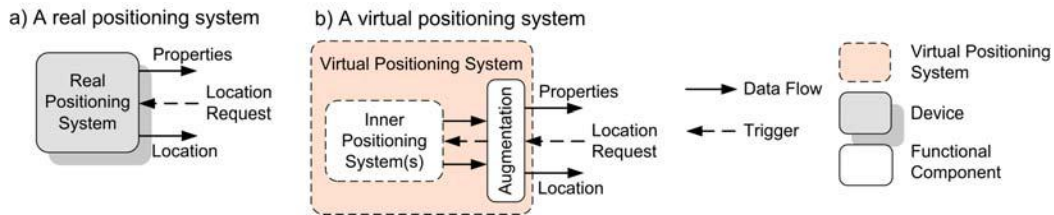


Fig. 1: Real and virtual positioning systems

A real positioning system (fig. 1a) is modelled as a black box that produces a location output on location request. The main idea of a virtual positioning system is to provide location output with higher quality (e.g., higher availability, greater coverage) and, at the same time, to preserve the general interface of location, properties and location request. Fig. 1b shows a virtual positioning system component. It contains one or more inner positioning systems and an augmentation component. The augmentation component takes the location output of the inner systems and generates new location output with improved characteristics. Typical types of augmentations hide details of inner systems, provide access functions and run a capturing protocol, convert one type of location information into information with a richer meaning or a standardized format or merge the output of several positioning systems into a single output. In principle, virtual positioning systems can be arbitrarily nested. Several combinations are conceivable which can even be established at runtime. Nimbus uses a fixed structure that is a direct result of the initial project goals. It contains the following augmentation components (from the innermost VPS to the outermost):

- *Access*: integrate real positioning system via a driver framework.
- *Mapping*: convert local location data into location data with a global meaning.
- *Selection & Collection*: provide a single location measurement even though different positioning systems are available.
- *Resolution*: convert physical into semantic locations using Nimbus semantic resolution capabilities [5, 9].

These components lead to the structure of nested VPSs as presented in fig. 2.

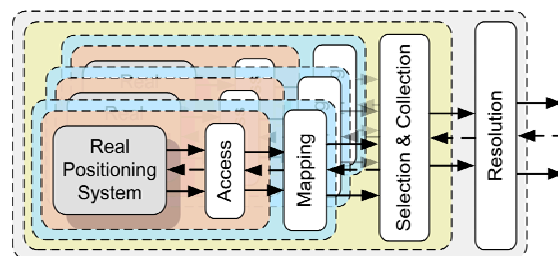


Fig. 2: Virtual positioning systems in Nimbus

3.1 Modelling Locations

From the alternatives presented above, Nimbus uses the area model to model location sensor output. Often, positioning systems are delivered without any detailed information about the probability distribution. To find out the actual distribution by measurements is very cost-intensive, so usually the distribution is simply estimated. In such cases, the area representation is a reasonable alternative to the exact probability distribution. Using the area model to represent systems with normal distributed location values is not optimal, but the region that covers 95% of the possible locations can be used for the area model without losing too much information.

Compared to detailed statistical models, areas are a simplification: first, the area does not identify a specific position with the highest position probability. Second, a strong border often cannot be sharply defined, as there is usually no most distant possible location for a certain measurement. As an advantage, an area is a very compact representation of a location sensor output. In addition, this model leads to very efficient processing algorithms.

3.2 Selecting Sensors

Currently, many projects in the area of location-based services or applications assume a single positioning system. Future mobile end-user devices will probably have access to several positioning systems. In a typical future scenario, a user has access to satellite navigation via GPS, indoor positioning systems which cover some buildings and positioning via a cell phone network, whenever more accurate systems fail. Applications, however, do not want to deal with various positioning systems and require a single location. There are two mechanisms to deal with several positioning systems:

- a subset of all positioning systems currently connected to the mobile device is activated (*selection* component), and
- the location information of all activated positioning systems that currently provide output is merged into a single output (*collection* component).

Collection: A simple approach to merge output of different location sensors is to simply choose the most accurate one. In many cases this approach is effective. Whenever an accurate system is available, less accurate systems do not contribute substantial information. As an example we consider two positioning systems: the Cricket indoor system (accuracy approx. 0.3 m) and GSM positioning using cell IDs (accuracy some hundreds meter in cities). The latter system does not provide any useful information as long as the Cricket system is available. Collecting the most accurate system for output requires only low computational resources and is thus a suitable approach for weak clients.

An improved collection component uses sensor data fusion. Whenever two or more positioning systems provide similar accuracy, the fusion can significantly improve the accuracy. According to the area sensor model, a user location has to be inside each of the provided areas, thus the geometrical intersection of all outputs describes all potential user locations.

Selection: In principle, the selection component can activate all positioning systems that are currently connected to the mobile system. In this case, all available location data are collected. This simple approach is appropriate if positioning is free of charge. Unfortunately, the user is sometimes charged for positioning services, for example when mobile phone networks are involved. For mobile systems also non-monetary costs such as power consumption have to be considered. A selection component can for example deactivate an unused positioning system to save battery power.

As a major problem, the selection cannot determine, if a positioning system produces output before the system is activated. E.g. the selection can decide to activate GPS (which drains the battery), but the GPS receiver does not receive any satellites. As the availability status may undesirably change after a selection, the algorithm can only try to select the most appropriate systems. An algorithm presented in [9] makes a selection based on former measurements. It maximizes the accuracy and availability, but the expected positioning costs do not to exceed a certain cost limit. Fig. 3 shows a simulation of this algorithm.

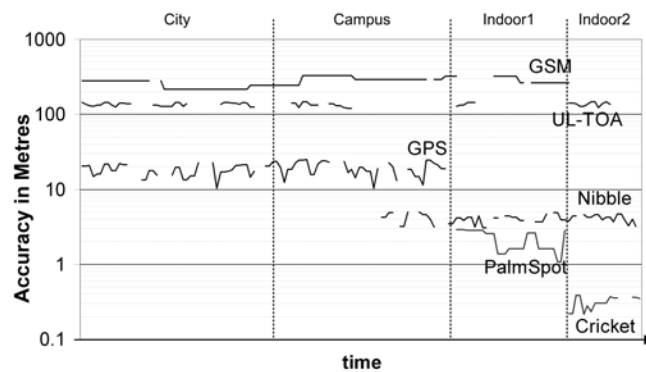


Fig. 3: Simulation of the selection component

4 Conclusion

In this paper, an approach to deal with different kinds of positioning systems is presented. Virtual positioning systems model all stages of abstractions from raw location output generated by real positioning systems up to high-level location output containing physical areas and semantic locations. The concept of drivers in particular simplifies the integration of new positioning systems in the future. Simulated positioning systems providing realistic testing scenarios are used during the application development phase.

Often, location output is considered as a single point in space. In reality, this assumption leads to misleading results, as many positioning systems only provide inaccurate location information. A location sensor model sketched in this paper addressed this issue.

1. Jörg Roth: *Mobile Computing*, dpunkt-Verlag, 2002
2. Jörg Roth: *Context-aware Web Applications Using the PinPoint Infrastructure*, IADIS International Conference WWW/Internet 2002, Lisbon (Portugal), Nov. 13-15, 2002, IADIS Press, 3-10
3. Jörg Roth: *Flexible Positioning for Location-based Services*, IADIS International Conference e-Society, Lisbon (Portugal), June 3-6, 2003, Vol. I, IADIS Press, 296-304
4. Jörg Roth: *Semantic Geocast Using a Self-organizing Infrastructure*, Innovative Internet Community Systems (I2CS), Leipzig, June 19-21, 2003, LNCS 2877, Springer-Verlag, 216-228
5. Jörg Roth: *Accessing Location Data in Mobile Environments – the Nimbus Location Model*, Mobile HCI 03 Workshop on Mobile and Ubiquitous Information Access, Udine (Italy), Sept. 8, 2003, LNCS 2954, Springer-Verlag, 256-270
6. Jörg Roth: *Flexible Positioning for Location-based Services*, IADIS Journal on WWW/Internet, Vol. I, Nr. 2, Dec. 2003, IADIS Press, 18-32
7. Thomas Hadig, Jörg Roth: *Proximity Services with the Nimbus Framework*, IADIS International Conference Applied Computing 2004, Lisbon (Portugal), March 23-26, 2004, IADIS Press, 437-444
8. Jörg Roth: *Data Collection*, in Jochen Schiller, Agnès Voisard (eds), *Location-Based Services*, Morgan Kaufmann Publishers, May 2004
9. Jörg Roth: *A Decentralized Location Service Providing Semantic Locations*, Habilitation Thesis, University of Hagen, Informatik Bericht 323, Jan. 2005

A Provider-Independent, Proactive Service for Location Sensing in Cellular Networks

Andreas Hartl

Telecooperation Group, Darmstadt University of Technology
andreas@tk.informatik.tu-darmstadt.de

Abstract

Using cell information for locating users in cellular networks is a common way to implement location based servers. However, implementations usually rely on infrastructure on the provider's side like special lookup servers. We present a light-weight solution that communicates the cell information to Web services. Our software is event based and therefore well suited for pro-active location based services. While still requiring network specific data, it is provider independent and extensible easily.

1 Introduction

Location is perhaps one of the most obvious context sources for mobile devices and mobile users, respectively. For deriving the location of a user there exists a wide range of location systems with different precision, coverage, technology, and price.

A simple way to determine coarse grained location in cellular networks is by identifying the cell. This paper presents a Web services' based approach for leveraging this cell information. Unlike other approaches, our software is decentralized and independent from the cellular provider. With this we want to lower the entrance barrier for programming location based services, giving more developers the opportunity to create tomorrow's killer application.

2 Related work

The GSM 03.71 Standard [2] defines a way for accessing location information in GSM based networks. The standard requires the network provider to operate a *Mobile Location Center*. This MLC collects cell information from the mobile switching center (MSC) currently responsible for the handset's base station and from the home location register (HLR) that stores subscription data and routing information for each handset. Clients that want to get location data send a request to the MLC. After the MLC has checked the HLR for the client's permission, it queries the handset's MSC for the location. Usually a client must actively query a user's current location, although the standard also specifies *deferred responses* that are sent when the user's location changes.

The MLC has access to the mobile phone operator's infrastructure. It is therefore possible to use more sophisticated location sensing mechanisms to get the location of a handset within a cell. The standard explicitly mentions time of arrival and timing advance based positioning mechanisms to be used by the provider. Zhao [6] gives several more examples of positioning

systems in 3G networks. Next to cell ID based positioning he mentions time difference of arrival based location – possibly extended by intersecting the information from several cells – and the GPS based Assisted GPS.

Implementors of location based services must have a contract with each operator they want to support which might be difficult to obtain and/or expensive with contract prices starting at several thousand Euros.

There is surprisingly little work on retrieving the cell information on a cellular phone directly. The Context project [4] at the University of Helsinki used the cell information to log the usage of a cell phone, along with incoming and outgoing calls and short messages as well as profile changes. Log information was stored on the cell phone itself and anonymously analyzed later. Berkeley’s Garage Cinema project [5] uses some of the software from the Context project to automatically annotate photos with context metadata including the cell information.

3 Implementation

Our approach is based on software running on an individual user’s cell phone rather than on a central server that may be queried about the location of a user. The software runs on Smartphones based on Nokia’s Series 60 or Microsoft’s Pocket PC 2003 and monitors the cell information. Once the device enters a new network cell, it publishes the new cell information. For transmission, we chose a low-footprint UDP-based protocol, containing the relevant cell identifiers (mobile country code, mobile network code, location area code and cell-id), the IMEI for identifying the phone and therefore the user, and a checksum. A proxy server in the Internet receives the data and transforms it into a SOAP message containing the very same information. This SOAP message is then broadcasted on the network.

The proxy is transparent to the clients of the location service. A cellular phone could also transmit the SOAP message directly. We chose the proxy-based approach to keep the network usage low. This helps when cooperating with bandwidth intensive application that may be used simultaneously, keeps the cost at the common volume-based tariffs low, and might make our solution more attractive to the cost-sensitive end-user market. Specifically, our solution is considerably less expensive than the ones already existing on the German market [1]: about 0.15 € each 100 cell changes instead of 0.20 € for a single location lookup.

3.1 Cell IDs and geographical coordinates

The SOAP message published comes in two different flavours. On the lower level, it just contains the user’s identity and the raw cell information. This information already might be used to create services based on cell granularity, for example a pro-active “friend finder” application that notifies its users if other people from their buddy list are in the same cell. However, many advanced services might require the geographical coordinates of a cell. For such services, we extend the basic SOAP message by the WGS84 coordinates of the cell. While our solution is independent from providers, the cell-to-coordinates mapping is of course provider specific as cellular providers are free in how to number their cells. We currently employ two different solutions for this:

- As a general purpose solution, we are using a cell lookup service that subscribes to the basic SOAP messages and looks up the location of a cell from a database. The SOAP message is then augmented by the geographical coordinates and retransmitted. This solution is applicable to any network, but requires manual registration of cells in the database.

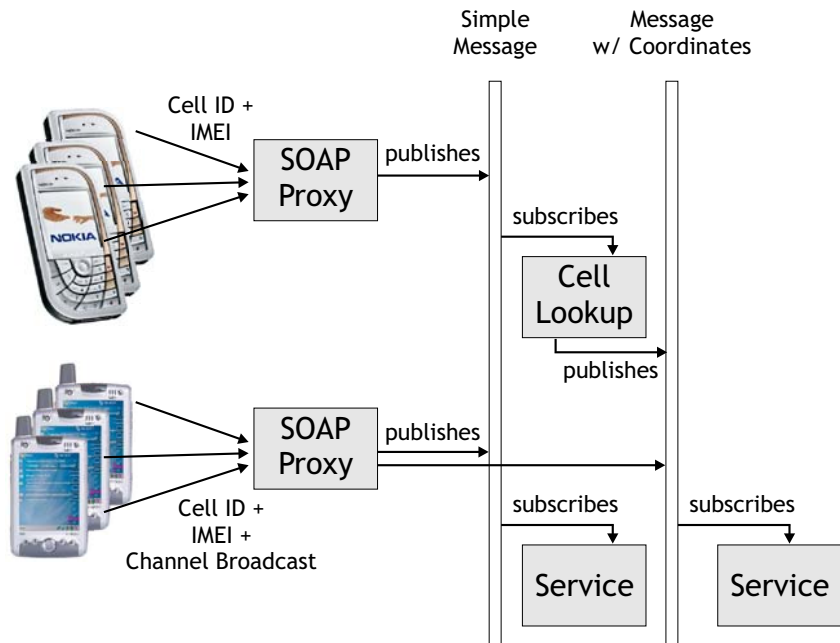


Figure 1: Example setup of the location sensing service with an event based routing transport.

- For the German provider O2 we also support another solution: we use an extended UDP protocol containing a field with the data from cell broadcast channel that contains the Gauss-Krüger-Coordinates of a cell. The proxy automatically converts this into the extended SOAP message.

As the low level cell information still might be of interest for some users, the extended SOAP message contains both the cell identification information and the cell's geographical coordinates.

3.2 Security considerations

Up to the moment our solution does not support any security or privacy. Location messages may be faked easily and location based clients may subscribe to any user's location information. While this eases testing and simulation, it is obvious that a production system must support secure transmissions and privacy.

The publisher of the SOAP message may ensure its integrity by signing it. If the proxy based approach is used, the UDP message might also be signed or symmetric key exchanged between proxy and handset.

Privacy might be ensured by encryption of location data in the SOAP message with a user specific key. Only those location based services trusted by a user would get her key and only those services could then decrypt the location data. The user ID would have to remain unencrypted in order to select the appropriate key for an incoming message. The event mechanism used by the system therefore would still allow the detection of movement of one user.

3.3 Programming location based service clients

Being a transport-independent protocol, SOAP does not specify the protocol used to deliver messages. We are using MundoCore [3], an event routing service that deals with the distribution of the location events transparently from the applications using it. Applications only need to

subscribe to the channel that is used for location message delivery to get an event when a user's cell changes.

As an alternative to MundoCore, applications may use the standard HTTP binding for SOAP-RPC to get information about location changes. Every time an event occurs, a callback function is invoked.

3.4 Sample applications

As a sample application, we built a location aware real-time train schedule. The service keeps track of the cell of a user by subscribing to the location information of a user. It determines the location of the nearest train station and keeps track of the train delays of there. The application may be queried explicitly using a WAP interface whereupon it shows the current timetable. Additionally, the schedule may use the data from the context aware calendar presented in [1] and notify the user when she should leave for a train.

4 Conclusion

We presented a way to implement a low-cost way for location sensing in cellular networks. The location information is based on the cell information that can be read by software on the mobile phone. The software is decentralized and provider-independent. For transmission of the data we chose a combination of a low-footprint UDP based protocol and a SOAP based event notification in order to balance between communication overhead and ease of use in location based client. For mapping the cell information to geographical coordinates we showed two possibilities: Looking up the position of a cell in a database or using provider specific features of the network.

References

- [1] G. Austaller. Web Services als Bausteine für kontextabhängige Anwendungen. In J. Roth, editor, *1. GI/ITG KuVS Fachgespräche - Ortsbezogene Anwendungen und Dienste*, Informatik Berichte 317 - 6/2004, pages 23–26. FernUniversität Hagen, 2004.
- [2] ETSI. 3GPP TS 03.71: Location Services (LCS); Functional description; Stage 2. Standard, 1998.
- [3] A. Hartl, E. Aitenbichler, G. Austaller, A. Heinemann, T. Limberger, E. Braun, and M. Mühlhäuser. Engineering Multimedia-Aware Personalized Ubiquitous Services. In *IEEE Fourth International Symposium on Multimedia Software Engineering (MSE'02)*, pages 344–351, Dec. 2002.
- [4] M. Raento. Mobile Communication and Context Dataset. Technical report, University of Helsinki, 2004.
- [5] A. Wilhelm, Y. Takhteyev, R. Sarvas, N. V. House, and M. Davis. Photo Annotation on a Camera Phone. In *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI 2004)*, pages 1403–1406. ACM Press, 2004.
- [6] Y. Zhao. Standardization of Mobile Phone Positioning for 3G Systems. *IEEE Communications Magazine*, pages 108–116, July 2002.

Context-Driven Service Discovery in Mobile Commerce

Philipp Bostan, Colin Atkinson, Oliver Hummel, Giovanni Falcone
University of Mannheim
68159 Mannheim, Germany
{bostan, atkinson, hummel, falcone}@informatik.uni-mannheim.de

Abstract: The promise of mobile commerce is to allow users to make the best possible use of their immediate environment with as little effort and inconvenience as possible. However, at present constructing such context-driven applications is a highly laborious and specialized task. In this paper we present a generic framework that simplifies the creation of context-driven service retrieval applications by allowing so called Service Discovery Services (SDSs) to be federated into arbitrary hierarchic configurations. Each SDS has the ability to infer user needs and desires from raw context information, find suitable services (or federated SDSs), and to integrate information into a single coherent result. This allows users to create, deploy and maintain sophisticated mobile commerce applications much more cheaply and effectively than it is possible today.

1 Introduction

Although mobile computing devices are now much more ubiquitous and powerful than just a few years ago, most people still primarily see them as communication and entertainment devices rather than as vehicles for engaging in “commerce” on the go. The expected explosion in mobile commerce transactions and services has so far failed to materialize, therefore, and the industry still awaits the so called “killer applications” that will convince customers to embrace mobile commerce en masse. These killer applications, when they do arrive, are likely to be location- and context-based systems which allow users to make the best use of their immediate environment given their prevailing requirements and circumstances. Mobile computing devices provide the optimal mix of communication and personalization capabilities needed to realize such applications. However, there are still some fundamental technical obstacles to their widespread deployment. These can roughly be divided into three groups.

The first group of issues can be categorized as security and privacy issues. These relate to the problem of prohibiting the misuse of highly personalized information[1],[2] collected by context-driven applications. Basically, the more “context-driven” an application becomes (in terms of automated determination and inference of user requirements) the higher the security risk associated with the inferred “context” information. Even the location knowledge needed to drive location-based applications is potentially highly sensitive.

The second group of issues can be categorized as context-determination issues. These relate to the problem of automatically determining the context information necessary to make applications as responsive and helpful as possible. This, in turn, can be divided into two main subgroups – one concerned with the sensing of “raw” context data such as location, time or environmental conditions (e.g. weather) and the other concerned with context processing by inference[3] of higher level context information from this raw data (e.g. whether it is night time or day time etc.) and reasoning[4] of contextual information which means the aggregation of multiple context sources into new contextual information (e.g. day time and location is the user’s office building, means user is at work).

The third group of issues can be broadly characterized as “ease-of-creation” issues. These relate to the problem of providing a technology that can support the relatively straightforward

construction and deployment of applications with customized levels of context dependency. It is already possible today to build quite sophisticated context- and location-based services, but only with a great deal of effort and specialized knowledge. Widespread deployment of these kinds of applications is unlikely to occur without an easy-to-use framework that “content providers” can use to build and customize their applications according to their specific “content” organization and business models. In the same way that the definition of J2EE[18] and .NET[11], as powerful but relatively easy to use technologies for building web applications, boosted the emergence of e-commerce, a similar framework for building context- and location-driven applications would significantly boost mobile commerce and provide the foundation for killer mobile commerce applications.

The Mobile Business Research Group at the University of Mannheim is addressing all of these concerns with the context of the SALSA and LAMBADA projects[5]. In this paper we discuss our ideas for addressing this third group of issues – namely the provision of a generic framework for creating and deploying context-driven mobile commerce applications. We first discuss some of the key foundational concepts for context-driven mobile commerce applications and give our own new interpretation of the notion of context. We then provide an overview of our generic mobile commerce framework and describe the different types of federation scenarios that it supports.

2 Service Discovery Services

To empower users to make the best use of their immediate environment given their prevailing requirements and circumstances an application needs to find and display *services* that match their user profile, context and location. We therefore characterize such applications as Service Discovery Services (SDSs). We regard a service as an activity performed by an organization to improve the state of a user. A service therefore delivers value. In general the activity may include the production and/or provision of physical artefacts (e.g. a book, a cinema ticket) or the production and/or provision of information (e.g. a program, a phone number).

Since customers interact with different kinds of services in distinct ways it is helpful to identify distinct kinds of services. An important differentiation has to be considered between electronic services which are delivered exclusively by IT technology and non-electronic service which are delivered purely through physical (non-IT) means. By definition, electronic services are pure information manipulation services. Typical examples of electronic services are Google and Online Yellow Pages. We regard a web service as a specific kind of electronic service implemented using XML technologies such as SOAP[6] and WSDL[7]. Not all electronic services are web services. Standard web applications such as Google are electronic services but are not implemented as web services. Thus, web services are a specific kind of electronic services and electronic services are a specific kind of service.

Services are often composed of a combination of smaller services. If a service is composed of a mixture of electronic and non-electronic services we refer to it as a hybrid service. Thus, a hybrid service is a service which includes (i.e. is composed of) an electronic service, but delivers value which cannot be attained 100% electronically. An example for a hybrid service includes Amazon’s product retail service, where the ordering of the product is made electronically over the internet, but the delivery of the value is realized physically by transport services.

Since a mobile user does not usually wish to be restricted to electronic services, our generic framework supports all of these different service types in a uniform way. This is achieved by means of a general Service Meta Model which captures the taxonomic relationships between the different services types and defines their common and distinguishing characteristics. The Service Meta Model is influenced by well-known technologies like WSDL, OWL-S[8], ebXML[9], Kobra[10], QoS and UML[12]. It basically

constitutes a service description language which has both an XML serialization syntax and a UML-based visualization syntax.

3 Context as a Relative Concept

In the literature on context-aware computing most authors define the term “context” in a fixed way by simply listing the set of concepts that they regard as representing context [13],[14],[15]. Typical examples include location, environment, identity and time[16]. This approach only makes sense, however, if (a) there is also a non-empty set of concepts which are regarded as constituting context and (b) there is consensus on what concepts are regarded as context and what concepts are not. Unfortunately neither of these is true in general. Many authors use such general concept to define “context” that effectively nothing is ruled out, and in the cases where there is an effective distinction between “context” and “not context” there is very little consensus. Absolute definitions of “context” have therefore effectively created a term that is meaningless. To address this problem and try to develop a useful notion of the term “context” we have adopted a radically different definition. Instead of defining context in an absolute sense we define it in a relative sense:

“Context is information used to deliver a service which is not explicitly input by the service requestor, but becomes visible during the course of the service delivery.”

A good example is provided by Google which uses implicit information about a user’s IP address to prepare a localized response to a user’s request. This information is not entered explicitly by the user when issuing a request, but becomes visible to the user when the result is returned. This definition means that information like location can only be characterized as context (or not as the case may be) relative to a specific service (or transaction).

4 A Generic SDS Framework

The key to the simple and cost-effective creation of context-driven SDSs (expected to be one of the killer applications of mobile commerce) is to provide a service discovery framework that allows content providers to organize their “content” in the way that best matches their specific business model or their existing content organization. By “content” here we mean information about services. Defining context in a relative way is a key step in this direction because it allows a regular service (e.g. a weather service) to be turned into a context-driven service by embedding it within a “context wrapper” which automatically determines context information (e.g. location and time) unbeknown to the user. The other key idea is to base the generic architecture on the well known composite pattern of the GoF[17], thereby enabling an SDS to itself be “content” in other SDS’s. This enables SDSs to be combined and federated in arbitrary ways. Figure 1 provides a schematic representation of our canonical SDS architecture. We refer to it as “canonical” because it shows a single (non-federated) situation in which a single SDS object provides context-driven searches on a “flat” repository of regular (i.e. non-SDS) services. However, when deployed, SDSs will often be configured to refer to other SDSs and to perform compound searches in cooperation with them.

The core part of the SDS architecture is the Universal Service Registry (USR) that stores the descriptions of the services registered with the SDS. Access to the USR is controlled by three interacting components which together provide the core federation, context processing and service retrieval capabilities of the framework – the Service Orchestration Engine (SOE), the Context Resolution Engine (CRE) and the Service Search Engine (SSE). The SOE is responsible for presenting the results of a search from the SDS’s local USR, and in the case of a federated architecture, combing this with information from other subordinate SDSs. The CRE is responsible for context inference and reasoning which are technologies that were mentioned by examples in the introduction. Together these raise the precision of the service retrieval process.

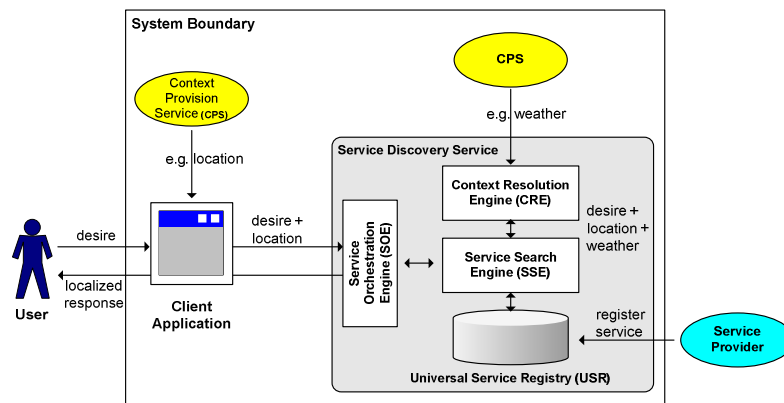


Figure 1. Context-Driven Service Discovery Architecture

The Service Search Engine (SSE) is the central coordinating component of an SDS since it communicates with all the other subcomponents. It is the basic matching and ranking engine which lies at the heart of the service retrieval process. When an external search request is received by the SOE it passes the request to the SSE which searches for suitable services from the USR. With the given service descriptions from the repository the SSE can now pass the request to the CRE which tries to receive necessary contextual information. With the received information the SSE ranks the services by service matching algorithms. The result that is returned to the client is a ranked list of services that are most likely to meet the user's needs.

A key feature of the SDS architecture is that SDS objects can be organized into federated hierarchies by virtue of the fact that SDS's can also be regarded as services in their own right. Four distinct federation models are supported:

- *User-Managed Linear Federation* in which the service requestor submits his desire to the client software that acts as service requestor by sending the request to the SDS. The SDS returns a list of SDSs which are presented to the user by client software. The user then chooses an SDS and interacts directly with it.
- *Client-Managed Linear Federation* in which the list that is returned by the SDS on a service request is processed automatically by the client software, which sends requests to the listed SDSs and returns a merged result set to the user.
- *Client-Mediated Linear Federation* which is similar to the User-Managed Linear Federation with the extension of downloadable components from service providers which are executable on mediating client software.
- *Server-Managed Hierarchical Federation* in which the SDS itself connects to suitable SDSs after a user has sent his service request. The SOE submits requests to the other SDSs and merges their results which are sent back in a result set to the client software that represents them to the user.

5 Conclusion

In this paper we have explained how a generic framework that supports the simple and cost effective creation of context-sensitive service retrieval applications will lay the foundation for the killer applications needed to drive the adoption of mobile commerce. We have presented the basic canonical architecture of our framework and have explained the two key ideas of defining context relative to specific services and of allowing SDSs to be federated in arbitrary way based on the business model or information architecture of content providers. We are currently implementing a prototype version of the framework and aim to apply it in the context of a small demonstrator case study.

Acknowledgements

This work was supported by grants from the Ministry of Science, Research and Arts of the state of Baden-Württemberg and the Landesstiftung Baden-Württemberg.

References

- [1] Jarvenpaa, S. L. et al: Mobile Commerce at Crossroads, Communications of the ACM, Vol. 46, No. 12, 2003, pp. 41-44.
- [2] Diezmann, N.: Payment – Sicherheit und Zahlung per Handy, in Kahmann, M. (Editor): Report Mobile Business – Neue Wege zum mobilen Kunden, Düsseldorf, 2001, p.155-178.
- [3] Dey A.: Providing Architectural Support for Building Context-Aware Applications, Ph.D. thesis, College of Computing, Georgia Institute of Technology, 2000.
- [4] Chen. H., Finin T., Joshi A.: An Ontology for Context-Aware Pervasive Computing Environments, Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review, Cambridge University Press, Vol. 18, No. 3, 2004, pp.197-207.
- [5] Mobile Business Research Group, University of Mannheim, <http://www.m-business.uni-mannheim.de>.
- [6] SOAP, “Simple Object Access Protocol”, <http://www.w3.org/TR/soap/>.
- [7] WSDL, “Web Services Description Language”, <http://www.w3.org/TR/wsdl>.
- [8] OWL-S, “Ontology Web Language”, <http://www.daml.org/services/owl-s/1.0/>.
- [9] ebXML, “Electronic Business using eXtensible Markup Language”, <http://www.ebxml.org>.
- [10] Atkinson C. and others, Component-Based Product Line Engineering with UML, Addison-Wesley, 2001.
- [11] Microsoft .NET, <http://www.microsoft.com/net/>
- [12] UML, “Unified Modelling Language”, <http://www.uml.org>.
- [13] Dey A.: Understanding an Using Context, Personal and Ubiquitous Computing Journal, Vol. 5(1), 2001, pp. 4-7.
- [14] Dey A., Abowd, G.: Towards a Better Understanding of Context and Context Awareness, 2000 Conference on Human Factors in Computing Systems, The Hague, The Netherlands, April 3, 2000.
- [15] Schilit B. N., Theimer M. M., Disseminating Active Map Information to Mobile Hosts, IEEE Network 8(5), 1994, pp 22-32.
- [16] Ryan N., Pascoe J., Morse D., Enhance reality Fieldwork: The Context-aware archaeological assistant, Computer Applications and Quantitative Methods in Archaeology, 1997; British Archaeological Reports, Oxford, 1998.
- [17] Gamma E., Helm R., Johnson R., Vlissides J.: Design Patterns – Elements of Reusable Object-Oriented Software, Addison Wesley, 1995.
- [18] J2EE, “Java 2 Platform Enterprise Edition”, <http://java.sun.com/j2ee/>

A Research Platform for Location-Based Applications

Dominic Heutelbeck and Matthias Hemmje
Computer Science Department, University of Hagen
{dominic.heutelbeck|matthias.hemmje}@fernuni-hagen.de

1 Introduction

In this paper presents a short overview of the platform ArWorx, a platform for location-based applications. Among others, it addresses distributed geographical data-sharing, geographical resource discovery, network services, and information visualization.

Location-based services are gaining an increasing importance. In literature, numerous different definitions for location-based services and applications exist, e.g., [1, 2, 4, 3]. They all have in common, that the location of a mobile device or geographical data is used for different tasks in an application to provide some valuable service to the user. In this paper, a very general notion for location-based applications is used:

Definition 1 *A **location-based application** is an application or network service that operates with geographical data from a part of the earth's surface. This part of the earth's surface is called the **service-area** of the location-based application.*

This definition includes all former definitions of location-based services. The word *application* is used instead of *service* to emphasize that the platform *ArWorx* does not only support location-based network services, but also arbitrary applications like geographic information systems, route planning, and location-based games. All these applications operate with geographical data and they may be used in a dynamic geographical context.

2 Requirements

As location-based applications gain acceptance, it becomes increasingly difficult for a user to find the right service. The number of services increases, the service-areas may be small and mobile, and the services may be only temporarily available. For a user, it is important to keep track of location-based services that can aid him in his current spatial context.

Requirement 1 *The platform must support the user in finding locally relevant location-based applications.*

A user may use more than one location-based application at the same time, e.g., he is using a navigation tool and has a friend finder application running at the same time. While this also implies requirements for the underlying operation system, it also means that these applications must be easily accessible to the user through a standard user interface.

Requirement 2 *The platform must support the presentation of multiple concurrent location-based applications in an accessible standard user interface.*

In addition to these requirements from the perspective of the user, it must also be easy for an application provider to publish a new location-based application. In this case, a location-based application may be a large scale information system or a single session of a mobile location-based multiplayer game.

Requirement 3 *The platform must provide an efficient way of publishing new location-based applications.*

The architecture of location-based applications is not limited to simple stand-alone applications or client-server systems. Location-based applications can be arbitrary distributed systems.

Requirement 4 *The platform must support applications with different distributed architectures.*

The hardware in this scenario is very heterogeneous and includes mobile phones, PDAs, Laptops, Tablet PCs, wearable computers, desktop workstations, and large servers. Besides their computational power and network connectivity, these devices have very different user interfaces. A single location-based service will be accessed via different kinds of devices. Thus, a matching user interface has to be provided for each platform. In addition, different classes of user interfaces like window based or augmented reality interfaces are based on completely different interaction metaphors. User interface plasticity [7], i.e., the automatic transformation of user interfaces, is an active research topic. However, in practice the user interface for a single application has to be redesigned for each type of target device.

Requirement 5 *The platform must support an extensible set of target devices and provide standard interaction widgets for each supported device type.*

A platform for location-based application should also support the basic location-based network services.

Requirement 6 *The platform must support basic location-based network services:*

- *Geocasting.*
- *Trigger services.*

These requirements are the key requirements addressed by the platform ArWorx. In addition, the following basic requirements apply:

Requirement 7

- *The platform must provide a driver infrastructure for location sensing and tracking hardware.*
- *The platform must provide a library for basic operations with different geographical coordinate spaces.*
- *The platform must provide a library for handling geometrical objects.*

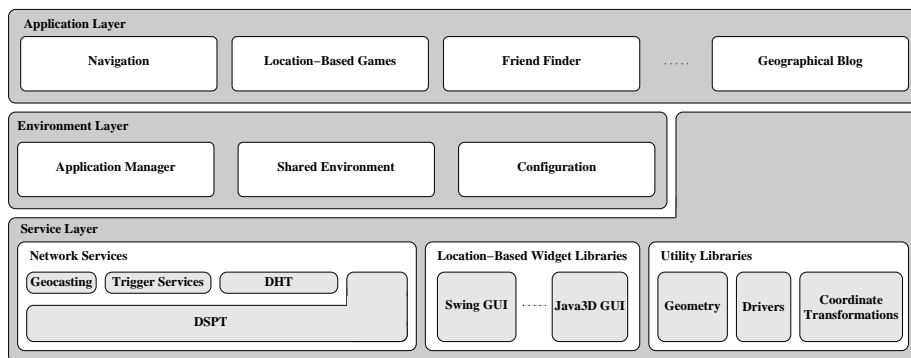


Figure 1: The architecture of ArWorx.

3 The Architecture of ArWorx

ArWorx is based on a three layer architecture as illustrated in figure 1. The base layer of ArWorx is the *service layer*. The service layer consists of three basic components:

- *Network services*: This component consists of the basic network services geocasting, trigger services, a distributed hash table (DHT), and a distributed space partitioning tree (DSPT) for storing geographical data.
- *Location-based widget libraries*: This component is a collection of user interface elements for different classes of mobile devices. The libraries implement location-based elements like interactive map widgets for tablet style devices, or 3D widgets for augmented reality systems.
- *Utility libraries*: This component is a collection of libraries that provide some fundamental functions for location-based applications, e.g., device drivers for positioning systems, tools for handling geometrical objects, and functions to transform locations between different standard geographical coordinate systems.

The second layer in ArWorx is the *environment layer*. The function of the environment layer is to manage the user’s personal environment. In ArWorx, the user’s environment is defined by the user’s spatial context and by the set of active location-based applications. The environment layer offers a *shared environment*. It is used to aggregate the location-based information from different simultaneously running location-based applications. In addition, the environment layer provides an application manager tool that allows to run, stop, and manage location based applications. The tool offers search for location-based applications according to their service-area. Finally, the environment layer also offers users the possibility to customize their user interface and to configure their environment, including available positioning systems. The role of the environment layer is similar to the role of a desktop environment for classical applications. The environment layer makes use of the services and libraries offered by the service layer. The final layer is the *application layer* that contains the actual location-based applications.

4 Comparison to Related Work

The two major existing platforms for location-based applications are Nimbus [6] and Nexus [5]. The approach of the platform ArWorx differs significantly. Some problems addressed by these platforms, such as geocasting and trigger services, are also addressed by ArWorx. However, the three platforms are based on three different architectures that have different strengths. RectNet is a peer-to-peer platform, Nimbus is a completely decentralized self-organizing infrastructure of fixed servers, and Nexus is a federation of servers with some central components. Nexus and Nimbus are both based on a distributed location model and the central purpose of these platforms is to support applications dealing with this model. The motivation for ArWorx is different. It is based on the thesis that the number of location-based applications and their individualization will increase drastically in the future. Thus, users will require support in handling these location-based applications. ArWorx offers this support. In addition, ArWorx introduces the concept of shared environments to help users to manage multiple simultaneously active location-based applications. These aspects are not addressed by the platforms Nimbus and Nexus.

5 Conclusion

The Platform ArWorx was realized using Java, and a number of interesting location-based applications were realized. The developers of the applications, such as a friend finder, a geographical messenger, a location-based Pong game, and a geographical blog, were relieved from many standard tasks, such as the mapping between geographical coordinate systems, or the management of distributed data. The developers were able to concentrate on the actual application logic. The shared environment proved to be a powerful concept to aggregate location-based information and applications. The application manager enabled users to search for location-based applications in an intuitive and efficient way.

In future publications we will describe the different solutions offered by ArWorx in detail. E.g., the shared environment, or distributed space partitioning tree (DSPT), a peer-to-peer datastructure for dynamic geometrical objects that is used to realize distributed geographical data-sharing.

References

- [1] T. Bychowski, M. Mbrouk, H. Niedzwiadek, J. Herring, and J.-F. Gaillet. OpenGIS Location Services (OpenLS): Part 6–Navigation Service. Technical report, Open GIS Consortium Inc., 2003.
- [2] GSM-World. GSM - Location Based Services. *PRD SE. 23*, 2002. www.gsmworld.com.
- [3] Ovum Ltd. *Mobile Location Services*. 2000.
- [4] M. Mabrouk, T. Bychowski, H. Niedzwiadek, Y. Bishr, J.-F. Gaillet, N. Crisp, W. Wilbrink, M. Horhammer, G. Roy, and S. Margoulis. OpenGIS Location Services (OpenLS): Core Services. Technical report, Open GIS Consortium Inc., 2004.
- [5] University of Stuttgart. Sonderforschungsbereich – Umgebungsmodelle für mobile Kontextbezogene Systeme. www.nexus.uni-stuttgart.de.
- [6] Jörg Roth. A Decentralized Location Service Providing Semantic Locations. Habilitation thesis, October 2004.
- [7] D. Thevenin and J. Coutaz. Plasticity of User Interfaces: Framework and Research Agenda, 1999.

Profilbasierte Rekonfiguration kleiner Endgeräte

"Roboter unterstützte Sensornetzwerke"

Gerhard Fuchs, Sébastien Truchat, Falko Dressler
Informatik 7, Universität Erlangen-Nürnberg
Martensstr. 3, 91058 Erlangen

{gerhard.fuchs, sebastien.truchat, dressler}@informatik.uni-erlangen.de

Zusammenfassung: In diesem Paper wird ein Vorhaben beschrieben, bei dem Sensorknoten von mobilen, autonomen Robotern rekonfiguriert werden sollen. Hierzu wird ein allgemeiner, profilbasierter Mechanismus vorgestellt, der für eine PDA/WLAN-Anwendung entwickelt wurde. Dieser Mechanismus wird mit besonderem Bezug auf die speziellen Anforderungen und Besonderheiten im anvisierten Szenario angepasst. Besondere Beachtung liegt dabei auf der Leichtgewichtigkeit des Mechanismus und aller beteiligten Operationen.

1 Einführung

Die primären Forschungsaspekte der Autonomic Networking Gruppe des Lehrstuhls für Rechnernetze und Kommunikationssysteme sind das Entwickeln und Beherrschen von selbstorganisierenden, selbst-konfigurierenden, und adaptiven Methoden für Kommunikationssysteme und interagierende autonome Systeme. Diese Aspekte werden im Rahmen des ROSES¹-Projekts (Robot assisted Sensor Networks) am Beispiel von mobilen Robotern und stationären Sensornetzen untersucht. Hierbei betrachten wir nicht nur die Teilsysteme (z.B. energiegewahre Aufgabenverteilung bei autonomen Robotern [3] oder bio-inspirierte Mechanismen zur Selbstorganisation von Sensornetzwerken [1]), sondern auch deren Kombination. Wir unterscheiden zwischen Sensornetzwerk unterstützten Roboterschwärmen, d.h., dass die Roboter das Sensornetzwerk z.B. zur präziseren Lokalisierung verwenden [2] und Roboter unterstützten Sensornetzen, d.h., dass die mobilen Roboter das Sensornetzwerks z.B. instand halten und warten.

Ein Beispiel einer Wartungsmaßnahme ist die profilbasierte Rekonfiguration der Sensorknoten unter Berücksichtigung des aktuellen Kontexts (momentane Aufgaben des Systems) durch die mobilen Roboter. Unser Ziel ist es, dieses Szenario mit dem im Mo.S.I.S-Projekt (Modulare Softwareentwicklung für Interoperative Systeme) erarbeiteten Mechanismus zu realisieren. Im Gegensatz zu den klassischen Client-Server-Ansätzen wie z.B. [5, 6] basiert die verwendete Architektur auf autonomen Knoten.

Das nachfolgende Paper gliedert sich wie folgt: Abschnitt 2, erklärt die theoretischen Grundlagen des Profiling Mechanismus, Abschnitt 3 die geplante praktische Umsetzung, Abschnitt 4 zeigt die bisherigen Erkenntnisse und die weitere Vorgehensweise.

2 Das Profiling Konzept

Im Rahmen des Mo.S.I.S. Projektes wurde an unserem Lehrstuhl ein Konzept für generische Rekonfiguration erarbeitet [7]. Ein Entwurfsmuster beschreibt einen Profiling Mechanismus, der im Zusammenhang mit einem leichtgewichtigen RPC Protokoll [8] eine

¹ Projekt-Homepage: http://www7.informatik.uni-erlangen.de/~dressler/projects/roses/roses_en.shtml

Plattformübergreifende "any-to-any" Rekonfiguration von mobilen Endgeräten ermöglicht (Abb. 1). Zielsetzung war es, einen sehr leichtgewichtigen Mechanismus zu entwickeln, um diesen auch für sehr kleine eingebettete Plattformen zu verwenden.

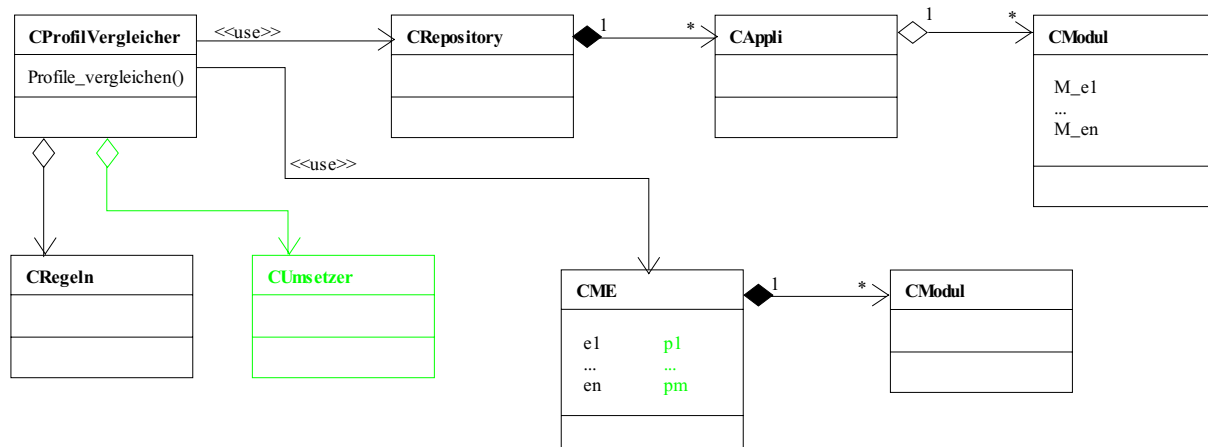


Abb. 1: Profiling Pattern

Das mobile Endgerät (Klasse CME) möchte sich automatisch mit den für ihn relevanten Softwaremodulen rekonfigurieren, die ein lokaler Server beherbergt (je nach Kontext, verschiedene Server mit unterschiedlichen Softwaremodulen: ortsabhängige Dienste). Das mobile Endgerät wird durch Profile charakterisiert, in denen seine Eigenschaften (e1 bis en oder p1 bis pm) stehen (Hardware, vorhandene Software, Benutzer Präferenzen). Im Repository des lokalen Servers (Klasse CRepository) liegen die konkreten Softwaremodule, dessen Eigenschaften (M_e1 bis M_en) auch durch ihre Profile beschrieben werden (für welchen Prozessortyp kompiliert, welches Display benötigt, ...). Applikationen (oder Dienste) bestehen aus mehreren Modulen. Eine Applikation (Klasse CAppli) wird auch durch ein Profil definiert, in dem die Eigenschaften der Applikation beschrieben werden (Version, ...) und die Liste der benötigten Module. Ein Applikationsmodul ist hier also ein abstraktes Konzept, welches konkret durch ein oder mehrere Softwaremodule realisiert wird (ein Applikationsmodul existiert z.B. als Softwaremodul für Prozessor xy und als Softwaremodul für Prozessor yz). Der Profile Matching Algorithmus (Klasse CProfilVergleicher) findet dann heraus, welche Softwaremodule vom lokalen Server dem mobilen Endgerät angeboten werden sollen, unter Berücksichtigung der schon vorhandenen Software auf dem mobilen Endgerät. Dazu benutzt er Regeln (Klasse CRegeln). Diesen Mechanismus kann man durch folgende Charakterisierungen verallgemeinern:

- Eigenschaften des Endgerätes: ME-Tupel = (eme1, ..., emen).
- Eigenschaften einer Applikation: App-Tupel = (eap1, ..., eapk, id1, ..., idq) wo eap1 bis eapn die Eigenschaften der Applikation sind, und id1 bis idq die Identifikatoren (z.B. Namen) der Applikationsmodule sind.
- Eigenschaften eines Softwaremoduls: Mo-Tupel = (id, emo2, ..., emom).

Der Profile Matching Algorithmus funktioniert folgendermaßen:

1. Welche Softwaremodule laufen überhaupt auf dem Endgerät? Regeln überprüfen, ob gewisse Anforderungen der Softwaremodule durch gewisse Eigenschaften des Endgerätes erfüllt werden (CUnsetzer übersetzt ggf. die Eigenschaften).
2. In dieser Untermenge von Softwaremodulen, wird jetzt für jede Applikation geprüft, ob auch noch alle nötigen Softwaremodule vorhanden sind, um die Applikation auf dem Endgerät zum Laufen zu bringen (nur lauffähige Applikationen werden behalten).
3. Von Modulen die zu einer Äquivalenzklasse gehören (selbe id) wird eines bevorzugt (z.B. ein Modul existiert in mehreren Versionen: die aktuellste wird bevorzugt). Applikationen können auch Prioritätsregeln haben. Hieraus entsteht die endgültige

Untermenge an Softwaremodulen, die dem Endgerät zum Runterladen angeboten werden.

3 Geplantes Szenario

3.1 Systembeschreibung

Bei dem geplanten Szenario sollen Sensorknoten von mobilen Robotern, die autonom durch das System fahren, rekonfiguriert werden. In unserem Labor kommen die von der Berkeley Universität entwickelten Mica2-Motes auf denen TinyOS² läuft und die von Fraunhofer AIS entwickelten Robertinos³ zum Einsatz. Bei der Rekonfiguration ist folgendes zu beachten:

- Die Sensorknoten übernehmen die Rolle des mobilen Endgeräts, die Roboter die des lokalen Servers.
- Auf einem Knoten läuft nur eine Applikation, die komplett ersetzt werden muss.
- Die Applikation besteht aus mehreren Codefragmenten, die auf dem lokalen Server ausgesucht und auf diesem vor der Übertragung zu einer Applikation kompiliert werden (Content Adaption).

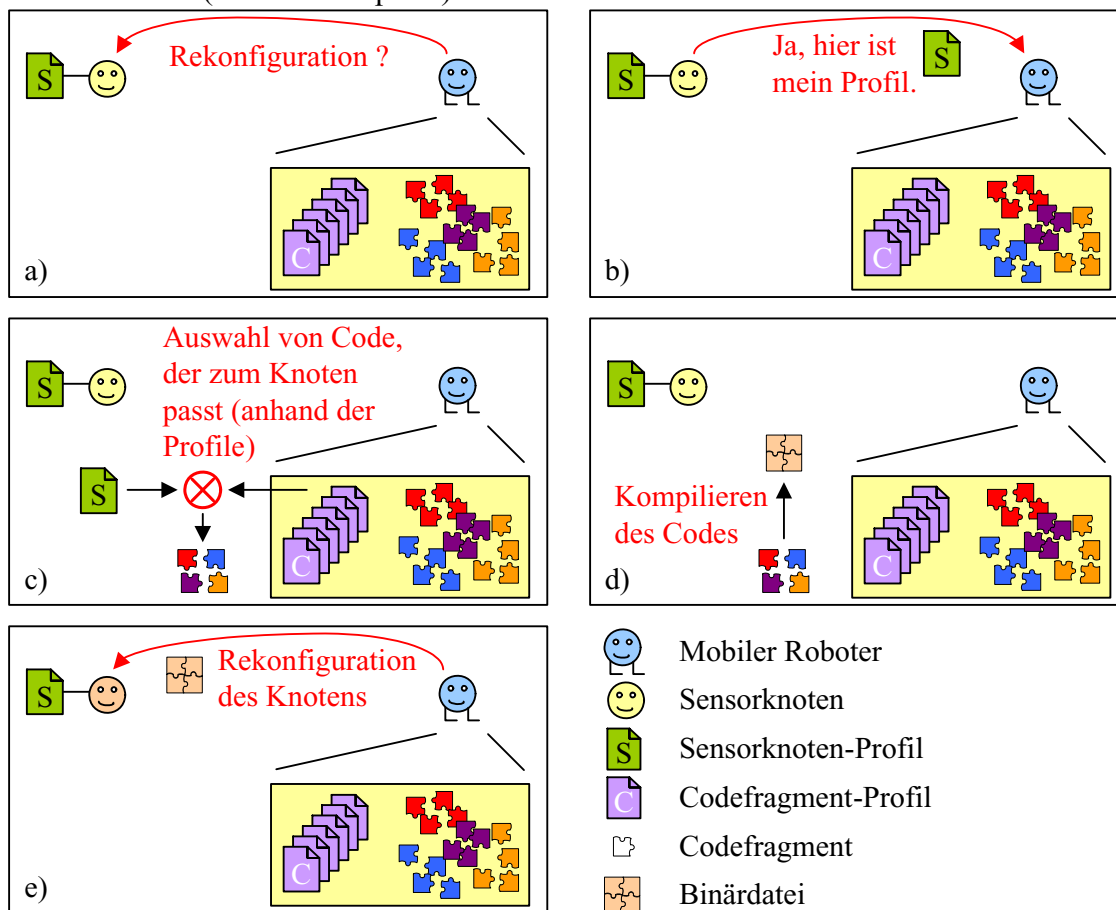


Abb. 2: Rekonfiguration eines Sensorknotens durch einen mobilen Roboter

3.2 Ablauf der Rekonfiguration

Ausgangssituation: Jeder Sensorknoten besitzt ein Profil, in dem seine Eigenschaften beschrieben sind (z.B.: Typ des Sensorknotens, Erweiterungshardware). Auf dem Roboter sind verschiedene Codefragmente abgelegt, aus denen eine neue Firmware für die Sensorknoten erstellt werden kann. Diese Codefragmente sind ebenfalls durch Profile

² Projekt-Homepage: <http://www.tinyos.org>

³ Projekt-Homepage: <http://www.openrobertino.org>

beschrieben, in denen die Anforderungen an das Endgerät vermerkt sind (z.B.: Typ Mica2 + Sensor MTS510).

Abb. 2 zeigt den Ablauf der Rekonfiguration eines Sensorknotens mittels eines mobilen Roboters:

- a) Der Roboter verkündet, dass eine Rekonfiguration nötig ist.
- b) Will ein Sensorknoten die Rekonfiguration durchführen, sendet er sein Profil an den Roboter.
- c) Der Roboter wählt anhand der Profile die Codefragmente aus, deren Anforderungen von den Eigenschaften des Sensorknotens erfüllt werden (*Profile Matching*).
- d) Die ausgewählten Codefragmente werden kompiliert, so dass als Ergebnis eine für den Knoten angepasste Software entsteht (*Content Adaption*).
- e) Die generierte Binärdatei wird auf den Sensorknoten übertragen, dieser startet neu, die Rekonfiguration ist abgeschlossen.

4 Erkenntnisse und weitere Vorgehensweise

Der hier vorgestellte Profiling Mechanismus wurde bereits prototypisch realisiert [4]. Die hierbei verwendeten mobilen Endgeräte waren im Vergleich zu den Sensorknoten (128kB Program Flash Memory, 38,4kBit/s Übertragungsrate) relativ mächtige PDAs mit WLAN-Anbindung. Es kamen ein symmetrisches Vokabular, externe Regeln und CC/PP (Composite Capabilities/Preference Profiles) zum Einsatz. Bei diesem Ansatz würde ein Sensorknoten für die komplette Beschreibung seiner Hardware / Konfiguration ca. 10% seines Speichers benötigen. Für das hier vorgestellte Szenario muss also eine ressourcenschonendere Umsetzung gefunden und die Content Adaption realisiert werden.

5 Literatur

- [1] F. Dressler, B. Krüger, G. Fuchs, and R. German, "Self-Organization in Sensor Networks using Bio-Inspired Mechanisms," Proceedings of 18th ACM/GI/ITG International Conference on Architecture of Computing Systems - System Aspects in Organic and Pervasive Computing (ARCS'05): Workshop Self-Organization and Emergence, Innsbruck, Austria, March 2005, pp. 139-144.
- [2] F. Dressler, "Sensor-Based Localization-Assistance for Mobile Nodes," Proceedings of 4. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze, Zurich, Switzerland, March 2005, pp. 102-106.
- [3] F. Dressler and G. Fuchs, "Energy-aware Operation and Task Allocation of Autonomous Robots," Proceedings of 5th IEEE International Workshop on Robot Motion and Control (IEEE RoMoCo'05), Dymaczewo, Poland, June 2005.
- [4] G. Fuchs, "Mobile autonome Dienste und ihr Profiling," Master's Thesis (Diplomarbeit), Dept. of Computer Sciences, University of Erlangen-Nuremberg, 2004.
- [5] T. Hakkarainen, A. Lattunen, and V. Savikko, "Flower - Framework for Local Wireless Services," in *ERCIM News*, vol. 50, pp. 51-52, July 2002.
- [6] M. Hillenbrand, P. Müller, and K. Mihajloski, "A Software Deployment Service for Autonomous Computing Environments," Proceedings of International Conference on Intelligent Agents, Web Technology and Internet Commerce, July 2004.
- [7] S. Truchat, "Interoperative Systems for Replenishment," Proceedings of Pervasive 2004, Doctoral Colloquium, Linz / Vienna, Austria, 2004.
- [8] S. Truchat and A. Pflaum, "Reconfigurable consumer direct logistics systems," Proceedings of 14. GI/ITG Fachtagung Kommunikation in Verteilten Systemen (KiVS), Kaiserslautern, Germany, March 2005.

Verbesserung personalisierter Informationsdienste auf Mobilfunkterminals durch Integration Ortsbezogener Daten

1. Einleitung

Gut eineinhalb Jahrzehnte nach ihrer Markteinführung haben digitale Mobiltelefone heute in Europa und insbesondere auch in Deutschland eine sehr hohe Marktdurchdringung erreicht. Aufgrund ihrer Handlichkeit hat sich für diese Geräte ja weitläufig auch die Kurzbezeichnung „Handy“ etabliert. Dass man mit diesen Geräten nicht nur telefonieren sondern auch Kurzbotschaften versenden kann, ist vor allem bei der jüngeren Generation wohlbekannt und wird auch intensiv genutzt. Andererseits haben so genannte Datendienste, die ebenfalls von Beginn an in den digitalen Funkkommunikationsstandard hineindefiniert wurden, bis heute weit weniger Beliebtheit am breiten Markt erzielt, als das allgemein erwartet wurde. Die mobile Datenkommunikation ist vergleichbar mit den Modemfunktionen über Analog- oder ISDN-Festnetzleitungen und ihre Anwendung ließe theoretisch ein Nutzungsspektrum wie für Internetzugriffe am Personal Computer zu.

Genau da liegt jedoch der Pferdefuß im System, weswegen Datendienste sich bisher nicht wirklich auf Mobilgeräten durchsetzen konnten. Schließlich wurde in einem direkten Ansatz die Idee des Web-Browsing mit der „Erfindung“ des WAP-Systems auf kleine Mobilgeräte übertragen. Dabei wurde jedoch völlig ignoriert, dass diese Kleinstgeräte für den Benutzer völlig andere Bedieneigenschaften aufweisen als leistungsstarke Personal Computer [1]: Neben einer eingeschränkten Rechenleistung und wenig Speicher ist der Bediener konfrontiert mit sehr kleinen Displays (~2% Bildpunkte eines Computerbildschirms), die oft wenig oder gar keine Farbtiefe aufweisen, und was die Eingabemöglichkeiten betrifft wäre die Bezeichnung „spartanisch“ auf vielen Geräten wohl noch eine positive Übertreibung.

Ein weiterer wesentlicher Faktor für den Benutzer von mobilen Diensten dürften wohl außerdem die absurden Tarife für die Datendienste darstellen. Während man in der aktuellen Festnetztechnologie (DSL) für ein Megabyte Übertragungsvolumen zurzeit ca. 1 Eurocent bezahlt, entstehen beim drahtlosen Zugriff Kosten, die um Faktoren von 100- bis 1000-mal höher sind. Darüber hinaus ist die drahtlose Kommunikation um ein bis zwei Zehnerpotenzen langsamer. Die Konsequenz ist, dass laut jüngerer statistischer Untersuchungen, weniger als 3% der Handybenutzer auch Datendienste regelmäßig einsetzen.

Auf der anderen Seite wären mobile Digitaltelefone gerade wegen ihrer enormen Marktverbreitung eine ideale Plattform für Datendienste. Deswegen sollen hier besonders solche Informationsdienste betrachtet werden, auf die von dieser Art Mobilterminals effizient zugegriffen werden könnte. Das sind z.B. Wetterkanäle, Verkehrsnachrichten, aber auch die Weiterleitung von elektronischer Post [2] kann in diese Kategorie fallen. Eine geschickte Kombination verschiedener Schlüsselkonzepte erlaubt für bestimmte häufige Anwendungsfälle, die Nutzung mobiler Informationsdienste sehr attraktiv zu gestalten: Einerseits lässt sich durch Personalisierung eine Minimierung der Nutzereingaben bei der Informationsabfrage erzielen. Weiterhin darf das Endgerät bei der Informationsübertragung keinesfalls auf das offene Internet zugreifen, da es von der Datenmenge völlig überfordert wäre. Vielmehr muss ein Agent (Abb. 1), der permanent am Internet zur Verfügung steht, die für den Mobilnutzer relevanten Informationen aus dem offenen Web herausfiltern und für die drahtlose Übertragung minimieren.

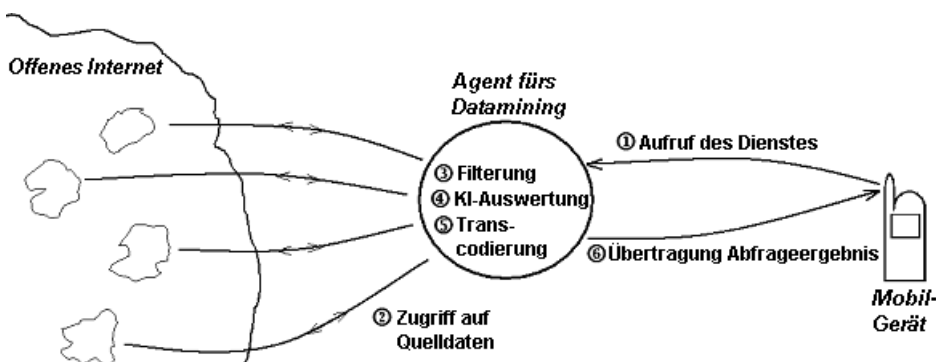


Abb. 1: Anstelle eines direkten Zugriffs auf das offene Internet kontaktiert das beschriebene mobile Informationssystem einen Softwareagenten für das sog. „Datamining“, der die relevanten Informationen aus dem Web oder von anderen Quellen herausfiltert. Dadurch wird die übertragene Datenmenge auf dem Funkweg minimiert, was die Zugriffskosten erheblich senkt und die Zugriffsgeschwindigkeit enorm steigert. Die Nummerierung zeigt die Reihenfolge der erforderlichen Einzelschritte.

Zum Beispiel interessieren Bahnpendler ausschließlich aus dem Web der Deutschen Bahn die Zugverspätungen am Abfahrtsbahnhof zum täglichen Ziel und nicht alle anderen Abfahrtsinformationen, die sich leicht zu mehreren hundert Kilobyte Quelltext aufsummieren (Abb. 2). Für PKW-Pendler, die täglich eine bestimmte Route benutzen, gelten vergleichbare Überlegungen. Durch den Einsatz von intelligentem Datamining können beim Mobilzugriff die notwendigen Bedieneingaben minimiert – in vielen Fällen sogar ganz wegrationalisiert – werden.



Abb. 2: Ein direkter Zugriff auf nützliche Webinhalte ist von mobilen Kleinstgeräten selten sinnvoll möglich, wie der linke Ausschnitt der Abfahrts-tabelle des Stuttgarter Hauptbahnhofs zeigen soll. Das Web-Dokument ist über 100 Kilobyte groß, weswegen die Übertragung enorme Kosten verursachen und selbst im echten UMTS-Netz mehrere zehn Sekunden dauern würde. Mit dem personalisierten Abfragedienst ist es dagegen möglich, die gewünschten Informationen auf kleinsten Systemen mit sinnvollen Kosten- und Zeitaufwand zuzugreifen. Dabei wird direkt ausschließlich die Information angezeigt, die den Benutzer interessiert.

Die Datenübertragungskosten werden in diesem Zugriffssystem dabei minimiert, weil nur das wirklich Wissenswerte über die Funkschnittstelle übermittelt wird, und dadurch wiederum wird der Datenzugriff erheblich beschleunigt. Dieses personalisierte Konzept erlaubt auch auf Kleinstgeräten, Informationsabfragen sinnvoll und ökonomisch durchzuführen. Als Beleg dafür seien Experimentserien berichtet [3], in denen mit dem entwickelten Prototypen eines Bahnabfragesystems für Pendler ein Abruf zeitaktueller Zugabfahrten typischerweise 10 Eurocent kostet und insgesamt etwa 15 Sekunden dauert (Abb. 2). In unseren Beispielapplikationen für dieses Konzept wurde die Terminalsoftware in wireless JAVA [4] implementiert, da dieses System mittlerweile Herstellerunabhängig auf sehr vielen Endgeräten zur Verfügung steht.

2. Mobiler Informationszugriff unter Ausnutzung der geographischen Position

In den beschriebenen Informationssystemen für den mobilen Zugriff verbleibt jeweils noch eine letzte Eingabevariable. Erst wenn auch diese automatisch bestimmen wird, kann die höchste Optimierungsstufe bei der Bedienung erzielt werden. Für die Stauabfrage ist dieser Parameter die Fahrtrichtung auf der gewählten Route, für das Bahnabfragesystem ist es die Frage, ob ein Pendler hin oder zurück fahren möchte. Diese Variablen können offensichtlich mit Hilfe der geographischen Position des Benutzer respektive seines/ihrer Anzeigeterminals aufgelöst werden: Befindet sich der Nutzer z.B. in der Nähe des täglichen Arbeitsplatzes, so gelten die nächsten Informationsabfrage automatisch der Rückfahrt, was einer bestimmten Fahrtrichtung bzw. einem Einsteigbahnhof entspricht. Auch die Weiterleitung wichtiger Nachrichten aus dem Büro kann auf Basis der geographischen Position des Bedienterminals vollständig automatisiert erfolgen.

Die Ausnutzung der geographischen Position des Nutzers erlaubt außerdem, von der sehr spezifischen Personalisierung in den mobilen Informationssystemen wieder etwas abzuweichen und ganz konkret allgemeinere Verkehrsinformationen anzubieten. Im Bahnabfragesystem wäre ja z.B. für eine Individualabfrage der Abfahrtsbahnhof und die Abfahrtszeit durch Ort und Zeit der Benutzung automatisch gegeben es müsste nur noch den Zielbahnhof eingeben werden. Als neues Beispiel könnte eine Verspätungsabfrage im Flugverkehr sogar ganz ohne Personalisierung auskommen, weil bei Start der Abfrage im ersten Schritt dem Nutzer eine Auswahlliste mit den Flughäfen angeboten werden kann, die nach umgekehrter Entfernung zum momentanen Aufenthaltsort sortiert ist. Dadurch steigt die Chance enorm, dass der Benutzer sehr schnell seine spezifische Wahl treffen kann. Wie nun die geographische Position für die mobilen Informationssysteme gemessen werden kann, wird in den nächsten Abschnitten diskutiert.

2.1. Das Satellitengestützte GPS

Die Bestimmung der geographische Positionen in elektronischen Geräten wird oft gleichgesetzt mit der Nutzung des Global Positioning System (GPS), dessen Infrastruktur vom US-amerikanischen Militär be-

trieben wird [4]. Sicherlich ist dieses System geeignet um in vielen Anwendungen verwertbare Positionsangaben zu erhalten, jedoch weist es auch Schwächen auf, die eine effiziente sowie eine zuverlässige Nutzung in manchen Anwendungsfällen verhindern. Z.B. kann innerhalb von Gebäuden oder in Ballungsregionen in Gebäudeschluchten der Empfang des Satellitensignals aussetzen und außerdem ist die Genauigkeit in bewegten zivilen Anwendungen ohnehin durch den Betreiber künstlich verschlechtert. Weitere Nachteile von GPS sind, dass dafür zusätzliche Hardware und dadurch wiederum ein höherer Stromverbrauch erforderlich wird. Falls es vorhanden ist, kann es sicherlich für die Steuerung der mobilen Informationsabfrage genutzt werden, jedoch müssten dann die geographischen Koordinaten noch in einen semantischen Kontext aufgelöst werden. Dieses Problem wurde jedoch bereits in Forschungsarbeiten ausführliche untersucht [6, S.11ff].

2.2. Cellbroadcast im Mobilfunknetzwerk

Jedes Terminal in digitalen Mobilfunknetzwerken verfügt automatisch immer über Positionsinformation, sobald es am Netzwerk angemeldet ist, weil das Mobilgerät sich ja mit einer Basisstation im Netzwerk verbinden muss. Die Funkzellenummer wird von der Netzwerkstation über so genannte Broadcast-Funkkanäle übermittelt. Genau genommen könnte diese sogar ohne Registrierung im Netzwerk empfangen werden. In Mobiltelefonen kann der Nutzer diese Zellenummer über Standardmenüs abfragen (Abb. 3). Im J2ME-System wurden die Cellbroadcast-Nachrichten ganz allgemein schon in der ersten Spezifikation des „Wireless Messaging“ berücksichtigt [7]. Da der Verbindungsaufbau in J2ME für alle Kommunikationsarten uniform erfolgt, kann z.B. die Mobilapplikation theoretisch über folgende Methode auf CBS-Kanäle zugreifen:

```
Connector.open ( "cbs://:0" );
```

Darüber könnte man dann die Zellnummer der Basisstation erhalten, an der das Terminal angemeldet ist, und diese wiederum könnte über Tabellen in eine geographische Position aufgelöst werden. Es existieren sogar Netzbetreiber wie O2, die auf einem dedizierten Cellbroadcast-Kanal die geographische Position der Basisstation übermitteln. In praktischen Experimenten hat sich jedoch herausgestellt, dass der Mechanismus, der seit einigen Jahren bereits spezifiziert wird, in den Geräten nicht unterstützt wird. Eine Portscanner-Software, die auf vier verschiedenen Mobiltelefonen namhafter Hersteller getestet wurde, war nicht in der Lage auf CBS-Kanäle zuzugreifen, obwohl das Experiment im J2ME-Simulator der Firma Sun einwandfrei funktionierte. Einen Handy-Prototypen aus neuesten Produktentwicklungen brachte die J2ME-Portscannersoftware sogar komplett zum Absturz. Um sicherzustellen, dass die Testsoftware selbst fehlerfrei war, wurde diese außerdem im SMS-Modus betrieben, der sich vom CBS-Zugriff nicht unterscheidet. Dieser Anwendungsmodus des Portscanners funktionierte auf den Endgeräten einwandfrei.

Zur Diebstahlverfolgung bieten übrigens manche Mobilnetzbetreiber die Möglichkeit, die eingebuchte Basisstation für das eigene Endgerät über einen Internetzugriff abzufragen. Dieser Service ist jedoch kostenpflichtig (laut Erfahrungsberichten in [6] 10 Eurocent pro Abfrage). Trotz der technischen Zugreifbarkeit macht es wohl im beschriebenen Konzept für mobile Informationssysteme keinen Sinn, doppelt für die Ermittlung einer Information zu bezahlen (Abfrage vom Netzbetreiber + Übertragung übers mobile Internet), die auf dem Endgerät schon vorhanden ist.

2.3. Location-Sensing über Proximity-Detektion mit Bluetooth

Mobile Geräte wurden in den letzten Jahren kontinuierlich um weitere Kommunikationsmethoden erweitert. Ein Funkstandard mit geringer Reichweite und geringen Stromverbrauch ist Bluetooth [8]. Auch Bluetooth ist bereits in die Programmierstandards für Endgeräte integriert, z.B. in J2ME nach Spezifikation JSR82 [9]. Als Konzepterweiterung im Sinne von ortsbezogenen Diensten kann das Endgerät im personalisierten Informationssystem mit Hilfe von Bluetooth nach anderen Geräten suchen. Wird z.B. der Büro-PC des Nutzers detektiert, ergibt sich automatisch eine verwertbare Positionsbestimmung, weil daraus das Abfragesystem für Verkehrsmeldungen die Fahrtrichtung schließen kann. Das Bahninfosystem kann so ebenfalls den nächsten Einsteigebahnhof bestimmen. Umgekehrt kann in pro-aktiven Systemen – wie der automatischen Weiterleitung wichtiger E-Mail- und Sprachnachrichten auf ein mobiles Terminal – ein stationärer Rechner (z.B. Computer im Büro) nach dem Endgerät mittels Bluetooth scannen und falls dieses nicht detektiert wird, werden die eingehenden Botschaften weitergeleitet.



Abb. 3: Das Mobiltelefon zeigt die eingebuchte Zelle (0731) und die Nachbarzelle (0730)

Nun ist Bluetooth-Kommunikation unter J2ME tatsächlich auf den Endgeräten funktional verfügbar, jedoch zeigen Experimente, dass die reine Gerätedetektion nicht aussagekräftig genug ist, weil anscheinend Cachingmechanismen in den Endgeräten der Applikationssoftware die Anwesenheit von Bluetooth-Geräten vorspiegeln, die schon aus dem Kommunikationsbereich entfernt wurden. Darüber hinaus kann natürlich aus Sicherheitsgründen die Sichtbarkeit von Bluetooth-Geräten in einer Kommunikationszelle abgeschaltet werden, wodurch dann eine Ortsbestimmung auf diesen Weg gar nicht mehr möglich wäre. Daher arbeiten wir zurzeit an einem Softwareprototypen, der an Stelle von Bluetooth-Geräten nach Diensten auf diesen sucht. Durch das Öffnen eines solchen Dienstes, der dann aber nicht benutzt wird, sollte sich eine sichere Erkennung von Bluetooth-Geräten im Umkreis und daraus wiederum die geographische Position ergeben. Unsere Experimente haben jedoch auch schon wieder einen erheblichen Nachteil der Bluetooth-Methode gezeigt, insofern als dass die Geräteerkennung mehrere zehn Sekunden dauern kann. Das würde die Zugriffszeit bei Stau oder Bahnabfragen erheblich verlängern, allerdings könnte sich durch eine Kombination mit Leastcost-Internet-Routing über Bluetooth bei der Informationsabfrage diese Nachteile evtl. überwinden lassen. Diese Annahme muss jedoch erst noch in zusätzlichen Untersuchungen nachgewiesen werden.

3. Zusammenfassung und Ausblick

Seit mehreren Jahren arbeiten wir an Konzepten sowie der Entwicklung von Informationssystemen, die Datenzugriffe auf mobilen Terminals in verschiedener Hinsicht verbessern [10]. Dabei ist dieses System besonders auf Kleinstgeräte zugeschnitten, die über das Mobiltelefonnetz auf Daten im Internet zugreifen. Erzielt wurden bisher durch eine Verringerung des auf dem Funkkanal übertragenen Datenvolumens eine Kostenreduktion und eine Erhöhung der Zugriffsgeschwindigkeit.

Als weiteres Merkmal ergibt sich durch Personalisierung eine stark vereinfachte Bedienung, welche nun durch die Integration von Ortsinformationen weiter optimiert werden kann. Für die Ortsbestimmung – im Prinzip die Messung der geographischen Position – existieren verschiedene Technologieansätze. Diese werden jedoch in den Konsumerprodukten nicht oder nur unvollständig unterstützt, weswegen sich die beschriebenen Verbesserungen noch nicht wirklich am Endgerät nutzen lassen.

Um nicht auf den Zeitpunkt warten zu müssen, ab dem die Gerätehersteller die Spezifikationen auch tatsächlich unterstützen, arbeiten wir bereits an „Workarounds“ in der Positionsbestimmungssoftware. Unsere bisherigen Applikationen wurden auf der Terminalseite in J2ME realisiert, jedoch besteht seit der immer breiteren Markteinführung von Geräten mit dem offenen Betriebssystem Symbian [11] die Möglichkeit, in C++-Applikationen alternative Gerätebibliotheken und damit eventuell die erforderlichen Funktionalitäten zu verwenden. Deswegen sollen im Rahmen entsprechender zukünftiger Machbarkeitsstudien die bestehenden mobilen Informationssysteme mit C++ nachgebaut und soweit wie möglich erweitert werden.

Quellenangaben

1. P. Johnson, „Usability and Mobility: Interactions on the move“, in First Workshop on Human Computer Interaction with Mobile Devices, University of Glasgow, GIST Technical Report G98-1, Mai 1998
2. H. Weghorn, „Electronic Mail Notification“, Studium&Praxis Nr. 1/2000, BA Stuttgart, 52ff, Juni 2000
3. H. Weghorn, „Efficient Information Access From Constraint Wireless Terminals“, in International Conference on Web Information Systems and Technologies, ICEIS Press, Miami, Mai 2005 (im Druck)
4. V. Piroumian, „Wireless J2ME Platform Programming“, Prentice Hall. Palo Alto, erste Ausgabe, 2002
5. <https://gps.losangeles.af.mil/>, Stand April 2005
6. J. Roth (Ed.), „1. GI/ITG KuVS Fachgespräch“, FernUniversität Hagen, 317, Juni 2004
7. JSR120, 2002. Wireless Messaging API (WMA), Version 1.0, Sun Microsystems Inc., Santa Clara, 2002.
8. Q. H. Mahmoud, „Wireless Application Programming with J2ME and Bluetooth“, <http://developers.sun.com/techttopics/mobility/midp/articles/bluetooth1>, Stand Mai 2005
9. JSR82, 2002. JAVA APIs for Bluetooth Wireless Technology, Vers. 1.0a, Motorola, Austin, April 2002.
10. H. Weghorn, „Teaching Wireless JAVA at the University of Cooperative Education“, in 2nd International Workshop on Wireless Information Systems WIS 2003, Ángers, Frankreich, 63ff, April 2003
11. www.symbian.com, Stand Mai 2005

Mit IDM und Mittler zu mehr Privatsphäre in LBS

Tobias Kölsch*

Lothar Fritsch[†]

Markulf Kohlweiss[†]

Dogan Kesdogan*

Zusammenfassung

In dieser Arbeit stellen wir eine Architektur und ein Protokoll für ortsbezogene Dienste vor, die die Privatsphäre der Benutzer schützen und den Verwaltungsaufwand zur Einrichtung eines Dienstes erheblich reduzieren. Dazu führen wir einen Ortsdatenvermittler ein, der Aufgaben der Ortsdatenverarbeitung übernimmt und so den Dienstanbieter von der Ortsdatenquelle trennt. Ein Großteil der Autorisierung und der Datenverwaltung wird dabei durch ein System zur automatischen Identitätsverwaltung geleistet. In dieser Arbeit betrachten wir passive Dienste, welche Datenschutzrechtlich problematischer sind, da die Benutzerposition bei ihnen über einen längeren Zeitraum verfolgt wird.

1 Einleitung und Problembeschreibung

Mit der weiten Verbreitung des Mobilfunks ist das Interesse an der Standortinformation von mobilen Telekommunikationsteilnehmern für kontextbezogene Dienste gewachsen. Viele Mobilfunkanbieter bieten bereits jetzt selbst oder über Drittanbieter ortsbezogene Dienste an. Diese reichen von ortsbezogener Tarifierung über ortssensitiven Auskunftsdiensten bis hin zur Flottenüberwachung. Allerdings ist die Einrichtung eines solchen Dienstes recht aufwändig. Da flexible, standardisierte Schnittstellen zur Übermittlung von Standortinformation von Benutzern fehlen, müssen diese häufig neu definiert werden. Außerdem müssen mangels existierender technischer Lösungen viele Aspekte, wie die Spezifikation der Zuständigkeit für die Benutzereinstimmung zur Datenweitergabe, auf vertraglicher Ebene geregelt werden. Dies führt zu einer erhöhten Einstiegsschwelle für neue Dienstanbieter. Eine Automatisierung dieser Schritte kann existierende Prozesse erheblich vereinfachen und

*{koelsch,kesdogan}@i4.informatik.rwth-aachen.de, RWTH Aachen

[†]{fritsch,kohlweiss}@whatismobile.de, JWG Universität Frankfurt am Main

erlaubt neue Geschäftsmodelle. Existierende Ansätze auf diesem Gebiet, wie z.B. [2] erfüllen nicht unsere Anforderungen an Datenschutz und Flexibilität.

In dieser Arbeit stellen wir eine Architektur vor, in der die Ortsinformation vom Mobilfunkanbieter nicht direkt an den Dienstanbieter weitergeleitet wird. Statt dessen fragt der Dienstanbieter die Daten über einen Mittler ab. Eine umfassende Begründung für die Sinnhaftigkeit eines solchen Mittler findet sich in [1]. Soweit möglich, verarbeitet der Mittler die Daten lokal und übermittelt nur das Ergebnis der Berechnung an den Dienstanbieter. Die Gewährleistung der Einwilligung durch den Benutzer und die Identitätsverwaltung wird dabei automatisch durch ein Programm erledigt.

2 Beschreibung der Architektur

Unsere Architektur beschreibt den Fall, dass wir einen Mobilfunkanbieter (MO) und einen von ihm getrennten Ortsbasierten Dienstanbieter (DA) haben, die dem Benutzer (B) gemeinsam einen Dienst zur Verfügung stellen. Der MO dient dabei sowohl als Kommunikationsanbieter, als auch als Positionsanbieter. Der DA implementiert die Funktionalität des Dienstes. Wie gesagt kommunizieren die beiden Anbieter nicht direkt, sondern über einen Mittler (LM). Der Aufbau ist auf der Grafik 1 dargestellt.

Die vier beteiligten Parteien sind mit einem automatischen Identitätsverwalter (IDM) ausgestattet. Konkret handelt es sich um den IDM, welcher im Rahmen des Prime Projektes [3] entwickelt wird.

2.1 Beschreibung des IDM

Ziel des Prime Projektes ist es ein Werkzeug zum ganzheitlichen Schutz der Privatsphäre und zur automatischen Verwaltung partieller Identitäten zu

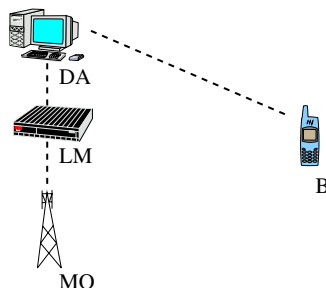


Abbildung 1: Der Benutzer fragt einen Dienst beim Dienstanbieter an. Dieser benutzt den Mittler um die Ortsinformation des Benutzers vom Mobilfunkbetreiber zu ermitteln.

erstellen. So trifft das System regelbasiert bei der Interaktion mit anderen Systemen Vereinbarungen darüber welche Daten über den Benutzer enthüllt werden. Das beinhaltet auch Abmachungen über Aufbewahrungsfristen der übermittelten Daten. Außerdem soll das System die Weitergabe von Informationen verfolgen. Anhand dieser protokollierten Daten kann der Benutzer sich dann darüber informieren, wem welche Daten über ihn bekannt sind.

Um Dienste zu ermöglichen, die es verlangen, dass bestimmte Eigenschaften des Benutzers bewiesen werden, bietet der IDM auch die Möglichkeit einzelne Aspekte eines Benutzers zusammen mit einer Zertifizierungsanstalt anonym zu beglaubigen. Bezahlung kann ebenfalls mittels des IDM erfolgen. Um die Privatsphäre des Benutzers auf allen Ebenen zu schützen, kann außerdem die gesamte Kommunikation anonymisiert werden.

2.2 Beschreibung des Protokolles

Es wird nun das Protokoll beschrieben, das in der vorhin beschriebenen Architektur die benötigte Funktionalität für ortsbasierte Dienste bietet. Wir betrachten hier nur passive, ereignisgesteuerte Dienste, wie z.B. einen ortsbasierten Stauwarndienst, oder einen Pollenwarndienst. Bei diesen wird der Benutzer gewarnt, wenn er sich innerhalb eines bestimmten Bereiches befindet.

Das Protokoll beinhaltet dabei zwei Einrichtungs- und einen Ausführungsschritt. Bei der Einrichtung veranlasst das Benutzergerät sowohl den MO als auch den LM dazu jeweils einen logischen Kanal zu erzeugen. Diese dienen von da an als Pseudonyme für den Benutzer. Zusätzlich werden noch Zugangsbeschränkungen für die Kanäle übermittelt. Außerdem erhält der LM noch einen Kryptoschlüssel.

Will der Benutzer einen Dienst initiieren, verbindet er sich zum DA, nehmen wir an es handle sich dabei um einen Pollenwarndienst, und übermittelt eine Liste seiner Allergien. Dann übermittelt er dem Anbieter die Zugangsdaten zu dem logischen Kanal bei dem LM und ein verschlüsseltes Datenpaket in dem sich die Zugangsdaten zu dem logischen Kanal des MO befinden. Dieses Paket ist mit dem Schlüssel der sich bei dem LM befindet verschlüsselt worden.

Beachte, dass die Kommunikation zwischen dem Benutzer und dem Anbieter anonymisiert werden muss, da der MO ansonsten über die Datenleitung erfahren kann, welche Dienste der Kunde nutzt.

Der DA berechnet nun die Regionen, die für den Benutzer von Interesse sind und verbindet sich zu dem Kanal der durch den LM geöffnet wurde. Er autorisiert seinen Zugriff mit den Daten, die ihm vom Benutzer mitgegeben wurden und übermittelt sowohl die Allergengebiete, als auch das

verschlüsselte Datenpaket.

Der LM kann nun das Paket entschlüsseln und sich anhand der in ihm befindlichen Daten zum MO verbinden. Die Zugriffsbeschränkungen des Kanals bestimmen, in welchem Maße dem LM Daten über den Benutzer übermittelt werden. Es kann auch die Genauigkeit der Übertragenen Daten eingeschränkt werden. Wenn die Verbindung steht und der LM sich autorisiert hat, kann er in dem benötigten Intervall die Benutzerposition abfragen. Diese vergleicht er dann mit den Allergengebieten und teilt dem DA mit, wenn er eine Übereinstimmung gefunden hat. Dieser kann dann die betreffende Information daraus ziehen und seinerseits den Benutzer informieren.

Zum Unterbrechen des Dienstes, teilt der Benutzer dem MO mit, dass er den Kanal schließen soll. Die Schließung wird dann weiterpropagiert.

3 Bewertung

Dieser Ansatz bringt auf verschiedenen Ebenen Vorteile. Zum einen abstrahiert der Mittler von der eigentlichen Ortsdatenquelle. Ein Dienstanbieter verbindet sich zu ihm unabhängig davon, ob die Daten aus einem WLAN-, einem GSM-Netz, oder vom Mobilgerät des Benutzers selbst stammen. Es könnten theoretisch sogar verschiedene Datenquellen gleichzeitig genutzt werden, um die Genauigkeit zu erhöhen. Außerdem erfährt die Datenquelle wenig über den Dienst, der vom Benutzer beansprucht wird.

Die Verwendung des IDM bietet den Vorteil, dass eine standardisierte Schnittstelle zur Verfügung steht. Außerdem macht der Autorisierungsmechanismus und die direkte Bezahlung das Abschließen von Verträgen zwischen Dienstanbieter und Mobilfunkbetreiber überflüssig. Und zu letzt garantiert das System die Zustimmung des Benutzers zur Weitergabe seiner Daten und gibt ihm mehr Kontrolle über seine persönlichen Daten.

Danksagung Vielen Dank an Lexi Pimenidis für seine Unterstützung.

Literatur

- [1] T. Kölsch, L. Fritsch, M. Kohlweiss, and D. Kesdogan. Privacy for profitable location based services. volume 3450 of *LNCS*. Springer, 2005.
- [2] Marwa Marbrouk and et. al. OpenGIS Location Services (OpenLS): Core Services. 2004.
- [3] PRIME WP 14.0. Prime Framework V1, 2005. <http://www.prime-project.eu.int>.

Ortsabhängiges Digitales Rechtemanagement

Thomas Mundt
Universität Rostock
18059 Rostock, Deutschland
thm@informatik.uni-rostock.de

Zusammenfassung: In diesem Paper stellen wir in Konzept und eine Architektur für ein ortsabhängiges Systems zur digitalen Rechteverwaltung (DRM) vor. Die Lösung basiert auf einer vertrauenswürdigen Hardware. Diese übernimmt die Positionsbestimmung mit Hilfe der Satellitennavigation, die Bereitstellung des Schlüssels für das DRM System und die Authentisierung der Positionsangabe.

1 Einleitung

Trotz aller ethischen und praktischen Nachteile haben DRM Systeme ihre Berechtigung, wenn es um den Schutz von geistigem Eigentum geht. Normalerweise wird der Schutz realisiert, indem digitales Material an an eine bestimmte Ressource, eine einzelne Person oder einen einzigen Computer gebunden wird. Es gibt einen offensichtlich Bedarf nach einem System, das den Zugriff nur an bestimmten Orten ermöglicht.

Die Idee der authentisierten Positionsbestimmung entstand auf Grund einer Reihe von Bedarfsszenarios. Eine kleine Auswahl:

- Eine Firma möchte sicherstellen, dass vertrauliches Material nur innerhalb des Firmengeländes einsehbar ist.
- Ein Oscar-nomierter Film soll nur bei den Juroren zuhause abspielbar sein.
- TV Shows oder DVDs sollen nur für ein einzelnes Land lizenzierbar sein.
- Ein Geldtransporter soll sich nur vor der Bank öffnen lassen.

Mit Hilfe unserer Idee ist es möglich, den Zugriff auf geistiges Eigentum auf ein bestimmtes Gebiet zu beschränken. Dazu bedienen wir uns eines der Positionsangabe, wie sie mit Hilfe eines Systems zur Satellitennavigation bestimmt wird. Ein solches System ist das europäische Galileo [1]. Die Hauptkomponente unseres Systems ist eine vertrauenswürdige Hardware. Diese übernimmt die Positionsbestimmung und überprüft, ob Anzeichen für eine Fälschung der Positionsinformationen vorliegen. Dazu enthält es eine präzise und vertrauenswürdige Zeitquelle. Dieser vertrauenswürdige Baustein muss zwingend als Hardware realisiert werden. Sie kann dann in einer Reihe von Geräten eingesetzt werden, z.B.:

- TV Settop Boxen oder Satellitenempfänger
- Geräte zur Datenanzeige (Notebooks)
- Geräte zur Datenspeicherung (Festplatten etc.)

Das System benutzt Plausibilitätsprüfungen um zu gewährleisten, dass das empfangene Signal nicht manipuliert wurde, zum Beispiel durch Empfang und Wiederaussendung an einem anderen Ort oder zu einer anderen Zeit.

2 Technologische Basis

2.1 Authentisierte Positionierung

Nach unseren Erkenntnissen gibt es bisher kein praktisch funktionierendes System zur authentisierten Positionsbestimmung mit dem existierende GPS [2]. In "An authenticated

Camera” [3] präsentieren Kelsey, Schneier und Hall wie für ein mit einer Kamera aufgenommenes Bild eine authentische Positionsangabe gespeichert werden kann. Sie beschreiben mögliche Angriffe, wie z.B. „rerouting“, wobei GPS Signale weitergeleitet werden. Sie kommen zu der Aussage, dass diese Form eines Angriffs nur schwer zu verhindern ist. Ein Artikel namens ”GPS Spoofing Countermeasures” [4] gibt einen generellen Überblick über Methoden, um GPS sicherer gegen sogenannte Spoofing Angriffe zu machen. Der Erfinder mit dem US Patent 5,922,073 [5] (”System and method for controlling access to subject data using location data associated with the subject data and a requesting device”) behauptet, eine Lösung für das in diesem Paper beschriebene Problem zu haben.

2.2 Digitales Rechtemanagement (DRM)

Um zu verstehen, wie die Position als Eingabe für ein DRM System benutzt werden kann, geben wir an dieser Stelle eine sehr kurze Einführung. Eines der Beispielszenarios verlangte die Beschränkung der Empfangbarkeit einer Fernsehsendung auf ein festgelegtes Gebiet.

DRM erlaubt es dem Rechteebsitzer zu entscheiden, unter welchen Umständen Lizenznehmer auf Dokumente zugreifen dürfen [6]. Der Zugriff auf DRM-geschütztes Material kann auf Operationen wie Lesen, Schreiben oder Ändern beschränkt werden. Eine Vielzahl von Bezahlmodalitäten wie z.B. Pay per View, Pay per Copy oder Pay per Instance sind implementierbar. Eine umfassendere Einführung findet sich in [7]. Das Material ist durch kryptographische Verfahren wie Verschlüsselung, Wasserzeichen oder Signaturen geschützt. Weitere Informationen zur DRM-relevanten Kryptographie finden sich in [11].

2.3 Satellitengestützte Navigation

Die meisten Szenarios für den Einsatz unserer Idee spielen in größeren Gebieten, wie z.B. Ländern oder Städten. Aus diesem Grund haben wir uns entschieden, zuerst ein großflächig arbeitendes Verfahren zur Positionsbestimmung als Location Provider zu wählen. Wir haben uns auf satellitengestützte Navigationssysteme konzentriert. Da die meisten Funktionen vom US-System NavStar GPS und vom europäischen Galileo identisch sind, werden wir Galileo als Beispiel benutzen [1]. Die Satelliten sind mit hochpräzisen Uhren ausgestattet. Ein Zeitsignal und die vorausberechneten Bahndaten der Satellitenflotte (Ephemeride) werden ständig von den Satelliten ausgesendet. Die mobile Komponente empfängt diese Signale von mehreren Satelliten und berechnet die Position aus den Differenzen der Laufzeiten der einzelnen Signale. Einige Dienste von Galileo arbeiten mit authentisierten Signalen. Aus diesem Grunde werden sie digital mit dem privaten Schlüssel signiert. Der öffentliche Schlüssel wird verteilt. Weitere Details finden sich in [13].

2.4 Präzise Uhren

Damit das vorgestellte System vor Rerouting-Angriffen geschützt werden kann während es offline arbeitet - also ohne Möglichkeit zur Rückmeldung - ist es notwendig, eine genau gehende Uhr in das System zu integrieren. Abhängig von der Gangungenauigkeit der Uhr kann die Zeit zwischen zwei Nachjustierungen gewählt werden. Hochstabile Quartz-Uhren, die zur Integration in Heimelektronik geeignet sind, haben eine Gangungenauigkeit von ca. $5 \cdot 10^{-7}$ [15]. Daraus ergibt sich die Möglichkeit, Uhren zu benutzen, die weniger als 15 Sekunden pro Jahr falsch gehen. Wir werden später zeigen, wie oft eine Zeitsynchronisierung mit diesen Uhren notwendig ist, um Rerouting-Attacken zu verhindern.

2.5 Authentisierte Uhrensynchronisation

Um die Uhr des vorgeschlagenen DRM-Systems zu stellen, ist es notwendig auf gesicherte Weise Zeitstempel mit einer für den Anbieter des geschützten Materials vertrauenswürdigen

Uhr auszutauschen. In seiner Dissertationsschrift aus dem Jahre 2004 [16] stellt Mykhailo Lyubich eine Methode zur abgesicherten Synchronisation von Uhren vor. Er diskutiert eine Reihe von Bedrohungen gegen solch ein System, wie z.B. „Impersonation“- und „Replay“-Angriffe. Für unsere Zwecke ist es ausreichend, die interne Uhr des DRM-Systems so mit der externen Uhr zu synchronisieren, dass die Authentizität der externen Uhr gewährleistet ist. Dies kann durch die Nutzung des Challenge-Response-Verfahrens mit zeitlichen Einschränkungen realisiert werden.

3 Konzept, Architektur und Bedrohungen

In diesem Abschnitt stellen wir kurz die Basisidee vor und motivieren die gewählte Architektur. Die Architektur hängt stark von der Analyse der möglichen Bedrohungen ab. Der Zusammenhang von Bedrohung und Design wird herausgearbeitet. Unser Ansatz übertrifft existierende Ansätze in folgenden Punkten:

- Wir benutzen präzise Uhren als Zeitreferenz, um Zeitdifferenzen zu erkennen, die sich aus Rerouting- oder Replay-Attacken ergeben.
- Wir führen eine vertrauenswürdige Hardware ein, um Manipulationen der Signale zwischen den einzelnen Komponenten zu vermeiden.
- Wir benutzen eine authentifizierte Methode zur regelmäßigen Synchronisation der internen Uhr.

3.1 Überblick über die vorgeschlagene Architektur

Um die Positionsinformation zur Steuerung des DRM-Moduls benutzen zu können, ist es notwendig, eine Möglichkeit zur Authentisierung der von den Satelliten empfangenen Signale zu finden. Das DRM-Modul benötigt darüber hinaus Informationen über die Fläche, innerhalb der der Zugriff (bzw. die jeweilige Operation) möglich sein soll. Diese Daten müssen ebenfalls gegen Manipulation geschützt werden. Es spielt dabei keine Rolle, ob das geschützte Material gesendet oder auf einem Hardwaremedium, wie z.B. einer DVD, geliefert wird. Abbildung 1 gibt einen Überblick über die allgemeine Funktionalität des DRM-Moduls.

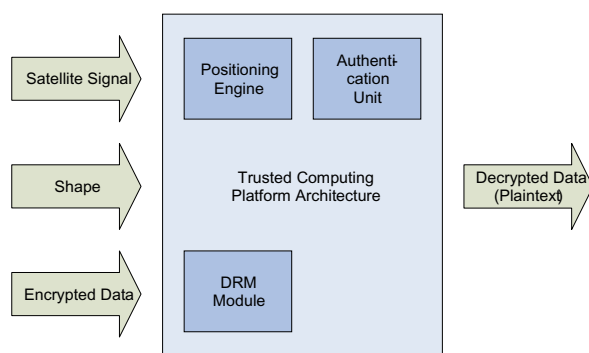


Abbildung 1: Arbeitsweise.

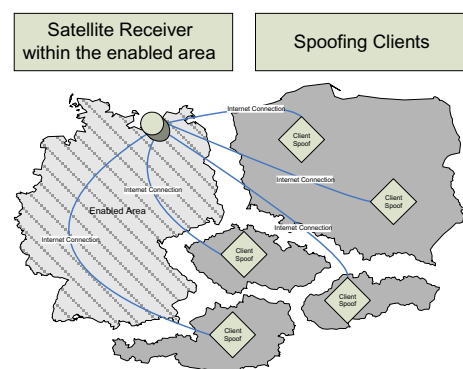


Abbildung 2: Rerouting-Angriffe.

3.2 Bedrohungen und Gegenmaßnahmen

Relevant für diese Idee sind Situationen in denen weder ein permanenter Rückkanal zur Verfügung steht noch eine externe Möglichkeit besteht, ein Gerät zu orten. Ein Rückkanal könnte zur authentifizierten Zeitsynchronisation oder zur Messung von Verzögerungen genutzt werden. Durch das Fehlen eines Rückkanals muss das Gerät in der Lage sein, autonom zu entscheiden, ob es sich innerhalb des gewünschten Gebiets befindet. Zwei Angriffe sind augenscheinlich geeignet, das Gerät zu täuschen.

Hochwahrscheinlich ist ein **Rerouting**-Angriff. Dabei wird ein Empfänger innerhalb des freigegebenen Gebiets aufgestellt. Dieser empfängt das Satellitensignal und verteilt es an die illegalen Empfänger außerhalb des Gebiets, wie in Abbildung 2 gezeigt. Die sich daraus ergebende Zeitdifferenz liegt im Bereich von Millisekunden beim Rerouting. Längere Verzögerungen entstehen beim **Replay**. In diesem Fall wird das Signal nicht online übermittelt, sondern zuerst zwischengespeichert und dann deutlich später wiedergegeben.

Eine mögliche Gegenmaßnahme gegen diese Angriffe besteht in der Integration einer hochpräzisen Uhr in das DRM-Modul. Die Uhr erlaubt es, Zeitdifferenzen durch Rerouting oder Replaying zu erkennen. Wie in Abschnitt 2 dargelegt, haben typische Uhren Gangungenauigkeiten von weniger als 15s pro Jahr. Angenommen, eine Internet-Verbindung wird für das Weiterleiten der Satellitensignale verwendet, so entstehen Verzögerungen von mindestens 5ms, verursacht durch aktive Netzwerkkomponenten. Eine ISDN-Verbindung würde die gleiche Verzögerung verursachen. Verbindungen mit geringerer Latenzzeit sind weit teurer und praktisch für Angreifer nicht verfügbar. Sie erfordern zumindest ein dedizierte Mietleitung. Einen Uhrenfehler von 15s pro Jahr unterstellend, sammelt die Uhr einen Fehler von 5ms innerhalb von drei Stunden auf. Daraus folgt, dass die Uhr alle drei Stunden neu mit einer externen, vertrauenswürdigen Zeitquelle synchronisiert werden muss, um Rerouting-Angriffe zuverlässig erkennen zu können. Diese Zeit kann weiter vergrößert werden, wenn ein konstanter Uhrenfehler angenommen wird.

4 Erste Realisierung

Wir haben das vorgestellte System prototypisch in Software auf einem Linux-basierten Satellitenempfänger [17] realisiert und erfolgreiche Tests mit Hilfe von GPS durchgeführt. Künstlich eingeführte Verzögerungen sollten Rerouting-Angriffe vortäuschen. Das System arbeitet einwandfrei.

Wir sind augenblicklich dabei, anderen Location Provider für die authentifizierte Positionsbestimmung nutzbar zu machen.

5 Literatur

- [1] O. Onidi, "Galileo is launched," 2002, http://europa.eu.int/comm/dgs/energy_transport/galileo/doc/galileo_is_launched.pdf.
- [2] E. D. Kaplan, *Understanding GPS: Principles and Applications*. Norwood: Artech House Publishers, 1996.
- [3] J. Kelsey, B. Schneier, and C. Hall, "An authenticated camera," in Proc. of the 12th Annual Computer Society Applications Conference (ACSAC). ACM Press, December 1996.
- [4] J. S. Warner and R. G. Johnston, "GPS Spoofing Countermeasures," http://www.homelandsecurity.org/bulletin/DualBenefit/warner_gps_spoofing.html, 2003.
- [5] K. Shimada, "Patent US 5,922,073: System and method for controlling access to subject data using location data associated with the subject data and a requesting device," 1999.
- [6] J. Feigenbaum, *Security and Privacy in Digital Rights Management*. Heidelberg: Springer, 2003.
- [7] W. Rosenblatt et.al., *Digital Rights Management: Business and Technology*. New York: John Wiley and Sons, 2001.
- [8] "RC4 License," A general introduction is available at <http://en.wikipedia.org/wiki/RC4>, The RC4 Algorithm is interlectual property of RSA Data Security, Inc., 100 Marine Parkway, Redwood City, CA 94065-1031, USA.
- [9] J. Daemen and V. Rijmen, *The Design of Rijndael. The Wide Trail Strategy (Information Security and Cryptography)*. Heidelberg: Springer, 2001.
- [10] N. Ferguson and B. Schneier, *Practical Cryptography*. New York: John Wiley and Sons, 2003.
- [11] T. Barr, *Invitation to Cryptology*. Upper Saddle River (NJ): Prentice Hall, 2002.
- [12] U. Reimers, *Digital Video Broadcasting (DVB). The International Standard for Digital Television*. Heidelberg: Springer, 2000.
- [13] The European Commission, "Mission high level definition," 2002, http://europa.eu.int/comm/dgs/energy_transport/galileo/doc/galileo_hld_v3_23_09_02.pdf.
- [14] G. W. Hein, J. Godet, J.-L. Issler, J.-C. Martin, P. Erhard, R. Lucas-Rodriguez, and T. Pratt, "Status of galileo frequency and signal design," 2002, http://europa.eu.int/comm/dgs/energy_transport/galileo/doc/galileo_stf_ion2002.pdf.
- [15] J. R. Vig, "Introduction to quartz frequency standards," <http://www.ieee-uffc.org/freqcontrol/quartz/vig/vigtoc.htm>, Army Research Laboratory, Tech. Rep., Oct. 2002.
- [16] M. Ljubich, "Methods for Enhancing the Security of Java Card based Payment Protocols," Ph.D. dissertation, University of Rostock, Rostock, Germany, 2004.
- [17] "Dream multimedia," <http://www.dreammultimedia-tv.de/>, 2004.
- [18] National Marine Electronics Association, "Nmea 0183 interface standard," 2002.
- [19] D. Eastlake, 3rd, "RFC 3174: US Secure Hash Algorithm 1 (SHA1)," 2001.

Datenschutzmechanismen für Ortsinformationen aus der Sicht zukünftiger Anwendungen

Georg Treu und Axel Küpper
LMU München, Lehrstuhl für Mobile und Verteilte Systeme

[georg.treu|axel.kuepper]@ifi.lmu.de

Zusammenfassung

Die Anonymisierung bzw. Verschleierung von Ortsinformationen stellt bei herkömmlichen reaktiven und selbstbezogenen Diensten, wie beispielsweise einem Restaurantfinder, bereits ein großes und nicht zufriedenstellend gelöstes Problem dar. Für proaktive Dienste mit hoher Genauigkeit oder gemeinschaftsbezogenen Diensten, wie sie in Zukunft abzusehen sind, scheint sich die Schwierigkeit der Findung geeigneter Verfahren noch zu verstärken. Dieser Beitrag zeigt für diese Arten von Diensten einige der kritischen Herausforderungen hinsichtlich Anonymisierung auf, gibt einen dahingehend aufbereiteten Stand der Technik wider und skizziert kurz einen eigenen, derzeit in Entwicklung stehenden Ansatz zur Anonymisierung proaktiver, ortsbezogener Community-Dienste.

1 Einführung

Obwohl die Entwicklung geeigneter Datenschutzmechanismen für Kontext- bzw. Ortsinformationen wohl als eines der Hauptkriterien für die Nutzerakzeptanz darauf aufsetzender Dienste zu sehen ist (vgl. [Lang01]), wurden in diesem Forschungsbereich leider noch keine zufriedenstellende Ergebnisse erzielt.

Techniken zum Schutz vor Mißbrauch von Ortsinformationen kann man grundsätzlich unterteilen in Mechanismen, die auf Privacy-Policies basieren, sowie solchen zur Anonymisierung bzw. Verschleierung von Ortsinformationen. Während Letztere zum Ziel haben, dem Nutzer technisch verifizierbaren Schutz vor Mißbrauch seiner Daten zu bieten, wird bei Privacy-Policies ein gewisses

Maß an Vertrauen zwischen Dienstanbieter und Nutzer vorausgesetzt, da die Einhaltung der Policies prinzipiell nur schwer kontrolliert werden kann.

Dieser Beitrag beschäftigt sich ausschließlich mit Anonymisierungs- und Verschleierungstechniken, da diese, anders als Policies, dem Nutzer gewisse Garantien bieten können. Der Schwerpunkt der Betrachtung liegt hierbei darauf, inwiefern bestehende Techniken den Herausforderungen zukünftiger ortsbezogener Dienste, insbesondere denen proaktiver sowie gemeinschaftsbezogener (Community-) Diensten gewachsen sind. Anschließend wird ein eigener Ansatz skizziert, der die Anonymisierung von Ortsinformationen bei proaktiven, ortsbezogenen Community-Diensten zum Ziel hat. Der Beitrag endet mit einem Fazit und Ausblick.

2 Herausforderungen zukünftiger Anwendungen

Herkömmliche ortsbezogene Dienste waren reaktiv und auf einen einzelnen Nutzer beschränkt, der auch mit dem zu ortenden Zielobjekt identisch war. Für Ihre Ausführung war ferner eine relativ grob-granulare Ortung, z.B. auf Cell-Id Basis, hinreichend. Ein typisches Beispiel für einen solchen Dienst ist beispielsweise ein *Restaurant-Finder*, der dem Nutzer Restaurants in dessen Umgebung anzeigt, eventuell unter Einbeziehung von Präferenzen des Nutzers, die dieser als Parameter mitliefern kann. Wie in [KuTr05] beschrieben, ist zu erwarten, dass sich in der Zukunft für ortsbezogene Dienste komplexere Szenarien ergeben.

Im Gegensatz zu reaktiven Diensten, bei denen der Nutzer den Dienst bewußt aufruft und sich dabei unter Umständen durch eine unbequeme Benutzerführung hangeln muss, arbeiten proaktive Dienste vorausschauend für den Nutzer, und informieren ihn über die (örtlichen) Ereignisse, die ihn interessieren, beispielsweise beim Betreten einer bestimmten Zone. Ein wichtiger Gesichtspunkt bei der Implementierung solcher Dienste ist, dass Nutzer vom System *verfolgt* werden. Dafür ist natürlich insbesondere notwendig, dass der Dienst über einen konsistenten Identifikator für den Nutzer verfügt (egal ob Pseudonym oder wahre Identität). Auch ist es wohl für solche Dienste eher wahrscheinlich als bei den reaktiven, dass sie Profilinginformationen über den Nutzer vorhalten (im Beispiel entspräche dies der Information, über welche Zonen der Nutzer informiert werden soll).

Man kann ferner unterscheiden zwischen Diensten, die sich auf einen einzigen Nutzer beziehen bzw. bei denen der Nutzer dem Zielobjekt entspricht, sowie Diensten, die es Nutzern erlauben, andere Nutzer zu orten bzw. ihren eigenen Aufenthaltsort mit dem anderer in Bezug zu bringen (vgl. Community-Dienste, unten). Für diese Dienste sind sowohl proaktive als auch reaktive Vari-

anten denkbar (vgl. [AEM+04] [TrKu05]). Festzuhalten ist jedoch erstens, dass für diese Dienstarten ein erhöhter Bedarf besteht, die gesammelten Ortsinformationen zu kontrollieren (wer aus der Community wann auf Ortsinformationen zugreifen ist u.U. ein sehr heikles Thema), sowie zweitens, dass hier konsistente Nutzeridentifikatoren umso wichtiger werden, da nun andere Nutzer in der Lage sein müssen, ein Subjekt zu (re-)identifizieren.

Eine andere, relativ offensichtliche und für Datenschutzmechanismen höchst relevante Unterscheidung ist die zwischen Diensten, welche nur eine relativ niedrige Genauigkeit der Ortsinformation benötigen, wie oben erwähnter Restaurant-Finder, sowie denen mit höheren Genauigkeitsansprüchen, wie z.B. ein Dienst, der den Nutzer genau dann benachrichtigt, wenn er vor dem Denkmal steht, an dem er interessiert ist.

Eine ausführlichere Behandlung verschiedener Anwendungstypen findet sich in [Kuep05].

3 Anonymisierungs- und Verschleierungstechniken

Für Mechanismen, die zum Ziel haben, die wahre Identität eines Nutzers bzw. Zielobjekts vor dem ortsbezogenen Dienst oder Dritten geheim zu halten (Anonymisierung), lassen sich v.a. zwei grundlegende Ansätze nennen.

[BeSt03] sieht zwei in Kombination verwendete Mechanismen zur Anonymisierung vor. Erstens soll die Nutzung eines Dienstes dahingehend eingeschränkt werden, dass Ortsinformationen nur dann übertragen werden, wenn sich ein Nutzer *nicht* in einer für ihn charakteristischen Zone, wie Wohnhaus oder Arbeitsstätte, befindet. Diese Beschränkung auf sogenannte *Application Zones* dient der Abwendung statistischer Angriffe, die sich mit Hilfe von Hintergrundinformationen über eine gesuchte Person durchführen lassen und mit Hilfe derer sich von wahrer Identität auf Pseudonym schließen lässt. Der andere, dazu als komplementär zu sehende Mechanismus erlaubt das dynamische Wechseln eines Pseudonyms in sogenannten *Mix-Zones*. So wird sichergestellt, dass zwei Pseudonyme eines Nutzers nicht miteinander verknüpft werden können.

Obwohl dieser Ansatz für einige herkömmliche ortsbezogener Dienste gut geeignet sein mag, fallen bei der Betrachtung proaktiver oder gemeinschaftsbezogener Dienste schnell Schwachpunkte auf: insbesondere für gemeinschaftsbezogene Dienste sind Pseudonymwechsel, welche ja dem Wechsel eines Nutzeridentifikators gleichkommen, nicht tragbar. Ferner scheint der generelle Ausschluss von Gebieten, in denen sich Nutzer per Definition die meiste Zeit aufhalten, insbesondere bei Diensten, die den Nutzer vorrauschaudend im täglichen Leben unterstützen sollen, als zu krass.

Ein anderer Ansatz, [GrGr03], stellt die Anonymität des Nutzers dadurch

sicher, dass dessen Ortsinformation stets so (räumlich oder zeitlich) verfälscht wird, dass mindestens $k-1$ Subjekte zum entsprechenden Zeitpunkt dieselbe Ortsinformation haben. Der Mechanismus beruht auf dem Prinzip der sogenannten k -Anonymität ([Swee02]). Abgesehen von dem Umstand, dass diese Technik wohl für viele zukünftige Dienste die Ortsinformationen in einer nicht akzeptablen Weise verschlechtert, hat der Ansatz noch zwei weitere Nachteile. Erstens ist zur Bestimmung des (sich dynamisch änderenden) Gebietes, welches k -Anonymität gewährleistet, eine zusätzliche Komponente notwendig, die eine globale Sicht auf die einzelnen Zielobjekte hat, und somit ein Sicherheitsrisiko darstellt. Zweitens liefert der Ansatz k -Anonymität jeweils nur für einen gespeicherten *Position Fix* eines Nutzers. Wird bei proaktiven Diensten ein Nutzer beispielsweise mit demselben Pseudonym zwischen Wohnort und Arbeitsplatz verfolgt, so hilft es wenig, dass die Information an beiden Orten k -anonym war. Durch die Betrachtung der Historie des verfolgten Subjekts lässt sich wesentlich sicherer auf dessen wahre Identität schließen, als mit Wahrscheinlichkeit $\frac{1}{k}$.

Andere Mechanismen zielen nicht darauf ab, die Identität des Nutzers geheim zu halten, sondern lediglich, dessen genauen Aufenthaltsort zu verschleiern. Als Beispiel wären hier [DuKu05] zu nennen, die ein formales Modell liefern, mit dessen Hilfe ortsbezogene Anwendungen weiterhin korrekte höherwertigen Informationen generieren können, wie z.B. die Bestimmung des dem Nutzer nächstgelegenen Restaurants, trotz zugrundeliegender unscharfer Ortsinformationen. Obwohl dieser Ansatz natürlich die Privatheit der Informationen verbessert, ist darauf hinzuweisen, dass dies wohl in vielen Fällen nicht genug ist. Ein provokantes Beispiel wäre hierbei der Chef, welcher wissen möchte ob sein Mitarbeiter in der Arbeit ist. Ob die Genauigkeit der erhaltenen Ortsinformationen nun entweder der von Tirol, oder der einer speziellen Skipiste entspricht, ist relativ egal. Der Chef kann trotzdem sicher sein, dass sein Mitarbeiter nicht am Arbeitsplatz ist.

4 Eigener Ansatz

Hier soll nun kurz ein eigener Ansatz vorgestellt werden, der in [TKR05] erstmalig andiskutiert wurde und derzeit am Lehrstuhl weiter bearbeitet wird. Er soll der Anonymisierung proaktiver, ortsbezogener Community-Dienste dienen. Darunter werden Dienste verstanden, welche die Aufenthaltsinformationen von Mitgliedern einer Community miteinander in Bezug bringen. Insbesondere werden hier also nur relative Distanzen zwischen Zielobjekten betrachtet. Globale Referenzpunkte, wie PoIs, werden vorerst nicht betrachtet. Beispiele für solche Dienste sind ein *Buddy Tracker* [AEM+04] [TrKu05], welcher Nutzer darüber informiert, wenn ein anderes Mitglied der Community eine vorgegebene Distanz

zu ihm unterschreitet, oder auch ein *SchoolClassObserver*, mit Hilfe dessen z.B. eine Gruppe (Grund-)Schüler dahingehend überwacht werden kann, dass der Lehrer benachrichtigt wird, sobald sich ein Kind aus der Gruppe entfernt, dessen absolute Aufenthaltsinformation allerdings nicht von Belang ist. Diese Gruppe von Diensten schließt z.B. auch *ortsbezogene "dating" Dienste* mit ein. Der grundlegende Mechanismus für solche Dienste ist also das Erkennen von Entfernen bzw. dem näher Kommen mehrerer mobiler Objekte. Die Idee zur Anonymisierung beruht darauf, die erwähnte Einschränkung auszunutzen, dass nämlich statt global gültiger Positionen der Teilnehmer nur deren relative Distanzen für die Funktionalität wichtig sind. Mitglieder einer Community sollen deshalb mit Hilfe eines gemeinsamen Schlüssels Ihre Ortsinformationen durch Distanz-erhaltenen Koordinatentransformationen verändern, bevor die Informationen den Dienst erreichen. Das Ziel ist hierbei, so statistische Angriffe auf Pseudonyme abzuwehren, ohne diese ständig wechseln zu müssen. Der Ansatz kann hier leider aus Platzgründen nicht näher dargestellt werden.

5 Fazit und Ausblick

Als Fazit ist zu sagen, dass bereits die Anonymisierung bzw. das Verschleiern von Ortsinformationen bei herkömmlichen ortsbezogenen Diensten kaum zufriedenstellend gelöst ist. Für proaktive Dienste mit hoher Genauigkeit oder gemeinschaftsbezogenen Diensten scheint sich die Schwierigkeit der Findung geeigneter Verfahren wohl noch zu verstärken. Insbesondere das Vorhandensein konsistenter Nutzeridentifikatoren bzw. die Verwaltung von Profilinformatioen scheint für solche Dienste eine Grundvoraussetzung zu sein, die allerdings bestehende Mechanismen nicht leisten können. Der etwas nüchterne Ausblick, den wir hier geben können, und der sich schon in dem skizzierten Ansatz für Community-Dienste widerspiegelt, ist, dass wohl Anonymisierungstechniken so gestalten werden sollten, dass sie Besonderheiten bestimmter Klassen von Diensten im Sinne der Privatsphäre des Nutzers ausnutzen. Dafür ist es wohl grundsätzlich notwendig, ortsbezogene Dienste erst einmal zu kategorisieren. Ein genereller Ansatz zur Anonymisierung von Ortsinformationen für alle Arten von Diensten scheint angesichts der dargestellten Problematik sehr schwer.

Literatur

- [Lang01] Langheinrich, M. Privacy by Design - Principles of Privacy-Aware Ubiquitous Systems, *Proceedings of Ubicomp 2001*, Springer, Lecture Notes in Computer Science, Vol. 2201, 2001, 273–291

- [Swee02] Sweeney, L. k-anonymity: a model for protecting privacy, *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, World Scientific Publishing Co., Inc., 10 (5), 2002, 557–570
- [BeSt03] Beresford, A., F. Stajano. Location Privacy in Pervasive Computing, *IEEE Pervasive Computing*, IEEE Educational Activities Department, 2 (1), 2003, 46–55
- [GrGr03] Gruteser M., D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking, *Proceedings of the First International Conference on Mobile Systems, Applications, and Services*, USA, San Francisco, May 2003
- [AEM+04] Amir, A., A. Efrat, J. Myllymaki, L. Palaniappan, K. Wampler. Buddy Tracking — Efficient Proximity Detection among Mobile Friends. *Proceedings of IEEE Infocom 2004*, Hongkong, March 2004.
- [KuTr05] Küpper, A., G. Treu. From Location to Position Management: User Tracking for Location-based Services. *Proceedings of the 14. Fachtagung Kommunikation in Verteilten Systemen KIVS05*, Kaiserslautern Germany, February 2005, Springer-Verlag.
- [TrKu05] Treu, G., A. Küpper. Efficient Proximity Detection for Location Based Services. *Proceedings of the Joint 2nd Workshop on Positioning, Navigation and Communication 2005 (WPNC'05) & 1st Ultra-Wideband Expert Talk (UET'05)*, Hannover, Germany, March 2005, SHAKER-Publishing.
- [TKR05] Treu, G., A. Küpper, P. Ruppel. Anonymization in Proactive Location Based Community Services. *Adjunct Proceedings of 3rd International Conference on Pervasive Computing. Pervasive 2005*. Munich, Germany, May 8-13, 2005.
- [Kuep05] Küpper, A. *LBS — Fundamentals and Operation*. To be published. John Wiley & Sons, 2005.
- [DuKu05] Duckham, M, L. Kulik. A Formal Model of Obfuscation and Negotiation for Location Privacy. *Proceedings of Pervasive 2005*, LNCS, pp. 152-170, Munich, Germany, May 8-13, 2005