

Jörg Roth ▫ Jochen Schiller ▫ Agnes Voisard (editors)

3. GI/ITG KuVS Fachgespräch Ortsbezogene Anwendungen und Dienste



Freie Universität Berlin

Institut für Informatik

Takustr. 9

14195 Berlin

Herausgeber:

Verlag:

ISBN:

Auflage:

Institut für Informatik der Freien Universität Berlin

Freie Universität Berlin, Universitätsbibliothek

3-929619-39-3

60

3. GI/ITG KuVS Fachgespräch

Ortsbezogene Anwendungen und Dienste

**am 7. und 8.9.2006
in Berlin**

Jörg Roth, Jochen Schiller und Agnès Voisard (eds.)

Inhalt

<i>SIGN - Adapting Navigation Instructions to Individual Users</i>	3
Ulrich Meissen, Stefan Pfennigschmidt, Agnès Voisard	
<i>Location Based Services im Tourismus</i>	8
Bernhard Kölmel	
<i>On the Applicability of Rule-Based Programming to Location Inference</i>	12
Katharina Hahn, Kirsten Terfloth, Georg Wittenburg, Jochen Schiller	
<i>Challenges in Information Systems for Disaster Recovery and Response</i>	16
Holger Kirchner, Thomas Risse	
<i>Modelling Geo Data for Location-based Services</i>	20
Jörg Roth	
<i>Implementierungsbericht Mobile Hunters</i>	26
Jörg Lonthoff, Erich Ortner, Michael Wolf	
<i>Ein Framework für kontextbezogene Anwendungen in der Nexus-Plattform</i>	32
Matthias Wieland, Daniela Nicklas	
<i>A Generic Approach to Modeling Context Constraints</i>	36
Rüdiger Gartmann	
<i>Impact of Misbehaviour and its Detection in Ad-hoc Wireless Sensor Networks using AIS</i>	40
Sven Schaust, Martin Drozda, Helena Szczerbicka	
<i>Positionserkennung von Studierenden in Hörsälen mit dem Chi-Quadrat-Anpassungstest</i>	44
Thomas King, Stephan Kopf, Wolfgang Effelsberg	
<i>Zwischenbericht: Ortsabhängige Auto-Adresskonfiguration in mobilen ad hoc Netzen</i>	49
Thomas Mundt	
<i>Self-Organizing Information Systems for Disaster Management</i>	53
Artin Avanes, Timo Mika Gläßer, Markus Scheidgen	
<i>Eine Rechteverwaltungsplattform für heutige LBS-Anwendungen</i>	57
Tobias Kölsch, Marc Wilhelm	
<i>A Middleware for Decentralized Wireless Networks</i>	61
Wolfgang Golubski	
<i>Eine Plattform zur Unterstützung von proaktiven ortsbezogenen Mehrbenutzer-Anwendungen</i>	65
Johannes Martens, Georg Treu, Peter Ruppel, Diana Weiss, Axel Küpper, Claudia Linnhoff-Popien	

SIGN – Adapting Navigation Instructions to Individual Users

Ulrich Meissen, Stefan Pfennigschmidt, and Agnès Voisard

Fraunhofer Institute for Software and Systems Engineering (ISST)
Mollstraße 1, 10178 Berlin, Germany
{ulrich.meissen, stefan.pfennigschmidt, agnes.voisard}
@isst.fraunhofer.de

Abstract. We focus here on the instruction component of navigation systems that take the experience and expectations of a user into account. The SIGN system that we present aims at giving drivers sensible instructions without being intrusive when for instance they already know a route. In other words, it adapts itself to the situations drivers are in and minimizes the set of instructions sent to them.

1 Introduction

Current navigation systems support two modes: either they are *off* or *on*. In the on-mode, instructions are sent to users, such as “take a left now”, “go straight on”, and so on. This can be annoying if users already know the route to be taken. Then the off-mode would be desirable, with the option of switching to an on-mode. However, a constant switch between an off- and on-mode is not user friendly. Our approach circumvents this problem by offering a combination of the two modes based on user’s familiarity with a predefined computed route. In other words, in an on-mode, we aim at defining the optimal minimal set of instructions based on familiarity and expectation. The problems are then (i) to define familiarity and (ii) to cope with unexpected events. For instance, a driver may be used to a 50 km/hour speed limit on a road where she drives every day (for instance, on Ku’dam in Berlin). This segment should not be pointed out to her as she is familiar with it. If, however, the road situation changes because of construction and if the speed limit becomes 30 km/hour, this should be indicated her as an unexpected event.

Another issue to take into account when considering the adaptivity of a navigation instruction unit is the situation users are in at a certain time and check it against previous situations and recognize whether the conditions are the same. For instance, if the user has taken a road segments many times at day time, we can infer that he or she is familiar with it *at day time* but not necessary at night.

This paper describes SIGN, a system that personalizes instruction sets based on users’ current situation and history and that handles exceptions based on current and unusual traffic conditions. Note that the system assumes a pre-computed route and extracts the optimal set of instructions from this route, given a user’s history.

The paper is organized as follows. Section 2 gives basic concepts. This is followed by the SIGN architecture (Section 3). Section 4 draws our conclusion and presents our on-going work.

2 Basic concepts

This section first gives our terminology before describing our approach.

2.1 Terminology

From navigation systems. As in common classical navigation systems, we denote a basic road element a **link**. A link can be a portion of a road between two intersections but also a dead-end street or a ramp onto a highway. A **segment** is an ordered set of links. This notion is introduced to aggregate links. A **route** is an ordered set of links. An **explained route** is a set of pairs (s, Exp) , where s is a segment and Exp an instruction (sometimes called “explications” in the navigation system jargon) taken from the basic set: [take-a-left, take-a-right, go-straight-on, make-a-U-turn-if-possible,...].

From situation-based systems. A **situation** reflects “something that is happening during a time interval”. It can be the activity of a person between 9 am and 10 am. It can also be the fact that a driver is driving on Avenue “Unter den Linden” between 11 am and 11:15 am, on a rainy day. What characterizes a situation is an associated set of characteristics such as {“driving”, “Unter den Linden”, “rainy day”}. Each characteristics belongs to a dimension. Moreover, each dimension is organized along a hierarchy of concepts. For instance, if we take the location of the driver, “Unter den Linden”, location is a dimension, and “Unter den Linden” appears at the street level of the hierarchy street/district/city/region/country. It is important to note that this hierarchy allows one to consider a situation at various levels of granularity.

More formally, a situation is defined as a tuple (t_b, t_e, P) , where t_b identifies the begin time, t_e the end time of the situation, and P represents the set of values of the dimensions describing the features of the situation [1]. Dimensions that we use in our system are: ‘transportation-means’ (with the following taxonomy: ‘any’, ‘car’, ‘bike’, ‘tram’), ‘street’ of type string, and ‘day-light’ of type boolean. The **history** of a user is stored as a situation sequence.

2.2 Personalizing navigation instructions

In order to adapt to drivers, we first need to consider the concepts of familiarity associated as a weight with a segment. It expresses the fact that a user has taken a segment a “certain number of times” under similar conditions. A second concept not described here is the one of “expectedness” [2].

Transportation situations are stored in a database. They are periodically evaluated to derive familiarity. With each segment is then associated a weight for a given user that reflects the level of familiarity. When a route is computed, all the segments that constitutes this road are extracted and assigned a new weight that reflect its similarity (in terms of road conditions, etc.) given the current conditions.

2.3 Approach

We focus here on the instruction problem in the context of routes computed in advance by a routing engine and we want to find out the best set of instructions based on personalization parameters. We do not compute the best route based on familiarity criteria.

Given a route R_{ref} computed by a routing engine, two cases need to be examined (we denote R_{taken} the route that the user actually takes):

1. $R_{\text{ref}} = R_{\text{taken}}$ The user sticks to the route that is given to him or her.
2. $R_{\text{ref}} \neq R_{\text{taken}}$ The route taken deviates from the one that was offered and evolves dynamically.

In the first case, the problem is a problem of combinatorial optimization and it is reduced to finding out the optimal combination of the longest-familiar segments in the history-familiarity list. The trade-off is as follows. Remember that we want to minimize intrusion, hence give the minimal set of instructions while making sure that the user knows the directions when we do not give her any instruction. Then the problem is to pick the segments that satisfy best the two criteria. For instance, on a 10 km route, one may have taken one of its 5 km segments 2 times and one of its 2 km segments 4 times (under similar conditions). The choice between the 5 km-segment and the 2 km-segment is left to the designer of the application.

In the second case, the problem is to find out at time t the longest-familiar segment that intersects with the current one, no matter if the picked segment is significantly shorter than the one that follows shortly after. Note that the trade-off in picking at any time longest-familiar segments is the same as above.

3 Architecture

The architecture of SIGN is given on Figure 1. The routes of the routing engine and additional information about conditions such as weather or daylight are transformed into sequences of atomic situations via feature extraction and interval aggregation as described in [1]. The *situation interpreter* serves as a general component that enables interpretation modules such as the *expectation interpreter* and the *familiarity interpreter* to compare actual situations sequences with historical situation sequences under different conditions. The rules of interpretation—e.g., the familiarity function for certain user groups depending on the number of times a user has taken the road and the similarity of the conditions—can be defined by the system designer via the *training tool*. The interpretation result is transformed by the *navigation explanation* component into adapted instructions for the user on his route. The *observer* component writes the observed user behavior into the database for the historical situation sequences. We have used this architecture in the TRANSIT project [3]. It is not optimized in terms of performance or efficiency but it proved to be a very flexible basis for interpreting a user’s situations in location-based services for several different tasks in this case for adapting navigation instructions.

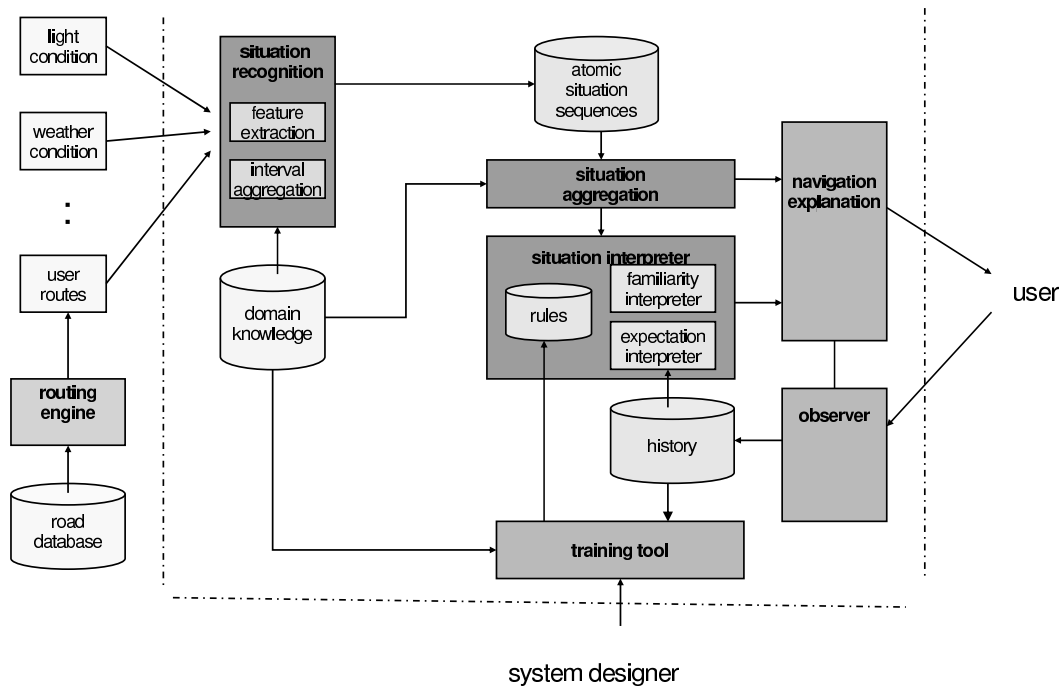


Fig. 1. The SIGN Architecture.

4 Conclusion

This paper briefly presented SIGN, a system that sends drivers a sensible set of navigation instructions based on their personal behavior and a given route computed by a navigation system. This notion is important as one could think of an approach that computes the *optimal familiar* route, which means the best possible route under familiarity criteria. In our architecture, this would imply an arrow between the situation interpreter and the routing engine. Parts of SIGN were tested in our BMBF-funded project TRANSIT [3]. Our current work consists in particular in designing efficient algorithms to recognize situations and in forming clusters of users (based on their profile) who may have the same types of requirements in terms of navigation instructions.

Acknowledgements

This research was partly carried out in the TRANSIT project, supported by the German Ministry of Education and Research (BMBF) under grant No. 19P4038A.

References

1. U. Meissen, S. Pfennigschmidt, A. Voisard, and T. Wahnfried. Context- and Situation Awareness in Information Logistics. *Proceedings of the EDBT Workshop on Pervasive Computing*, Lecture Notes in Computer Science No. , Springer Verlag, Berlin/Heidelberg/New York, 2004.

2. U. Meissen, S. Pfennigschmidt, A. Voisard, and T. Wahnfried. Resolving knowledge discrepancies in situation-aware systems. *International Journal of Pervasive Computing and Communication JPCC*, 1(4):327–336, Dec. 2005.
3. TRANSIT : An Intermodal Transportation Information Management System in the Context of Large-scale Events. www.transit4events.org, as of July 2006.

Location Based Services im Tourismus

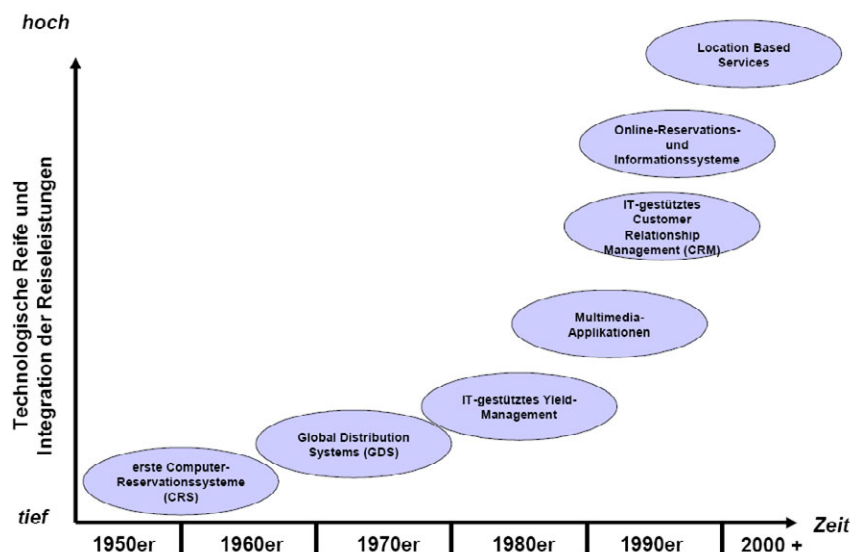
YellowMap AG
Bernhard Kölmel
Wilhelm-Schickard- Str. 12
D-76131 Karlsruhe
eMail: Bernhard.Koelmel@YellowMap.de

Hintergrund

„Location Based Services gelten als die kommende Killerapplikation. Mit der geographischen Position des Benutzers ist man in der Lage ihm Dienste anzubieten, die bisher nicht möglich waren“. Von der viel zitierten „Killerapplikation“ der Jahre 2001 bis 2003 haben wir uns schon vor einiger Zeit verabschiedet. Gerade lassen wir die Phase hinter uns, in der wir allgemeingültige Geschäftsmodelle kreieren wollen, von der die ganze Branche profitiert.

Damit im Vorfeld eine unkontrollierte Entwicklung der Märkte, wie durch den Internet-boom 1995 und dem damit verbunden Zusammenbruch der Dotcom-Blase im Jahr 2000, vermieden wird, sollten spezifische anwendungsnahe Geschäftsmodelle anhand von realistischen betriebswirtschaftlichen und informationstechnischen Aspekten betrachtet werden.

Der Tourismus ist als Branche bekannt, welche die neuesten Entwicklungen (e-Business) in kurzer Zeit aufnimmt und in praktische Lösungen für den Kunden und die beteiligten Geschäftspartner umsetzt. Bereits im Jahr 2002 bezeichnet Zipf „GIS und LBS als Grundlage für moderne Tourismusinformatiossysteme“ [Zip02]. Innovative Entwicklungen im Bereich innovativer Informationssysteme (GIS) finden seit langem Anwendung im touristischen Bereich. So wurden in den 50er Jahren die ersten Reservationssysteme durch die Airlines entwickelt. In den 60ern und 70ern wurden diese Systeme auf Reiseveranstalter und Mietautofirmen, später auch auf Hotelketten ausgedehnt. Diese Integration verschiedener Leistungen in verschiedenen Regionen der Welt unter einem Dach wird über Globale Distributions-Systeme (GDS) sichergestellt. In den 80er Jahren führte man – ebenfalls durch die Airlines angetrieben – die ersten IT-gestützten Yield-Management-Systeme ein, die bei Reiseveranstaltern und Reisebüros in den 90ern in angepasster Form übernommen wurden. Auch in den 80ern haben die ersten Multimedia-Lösungen in der Tourismusbranche ihren Einzug gehalten [Ber05]. Die folgende Abbildung verdeutlicht diese Entwicklung.



Unter dem Thema "Mobiler Tourismus" wurden bereits eine große Zahl verschiedener Projekte aus dem Bereich GIS in touristischen Anwendungen implementiert. Insbesondere werden Dienste propagiert, mittels derer Interessierte vor Ort Zugriff auf aktuelle, positions-bezogene Informationen (LBS: Location Based Services) erhalten können. Durch die rasante Entwicklung mobiler Endgeräte und dem Ausbau neuer, breitbandiger Telekommunikationsnetze wird den LBS-Diensten eine starke Bedeutung in der Zukunft zugemessen.

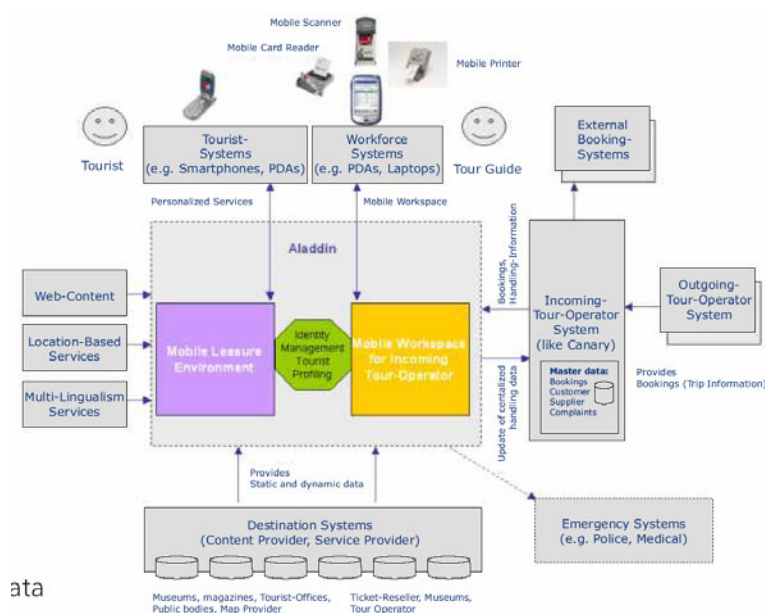
So rasch wie die Technologien von der Branche wahrgenommen und teilweise umgesetzt werden, so langsam schreitet deren Reifungsprozess zu breitenwirksamen integrierten Lösungen voran. Das aktuelle Verhalten der Akteure in der Tourismusbranche lässt sich folgendermaßen charakterisieren [Ber05]:

- Rasche Aufnahme innovativer Ideen und Lösungen
- Punktuelle Erfolge aber keine Kontinuität
- Schwer nachweisbare, teilweise unmögliche Messung der „Erfolgsstories“
- Gewünschte, aber nur punktuell erfolgte Koordination und Integration der Lösungen

Insbesondere traditionelle europäische Destinationen stehen vor der Notwendigkeit, angesichts der globalen Dimensionen des internationalen Wettbewerbs und zur Stärkung der regionalen Wettbewerbsfähigkeit neue Handlungsoptionen zu erarbeiten. Sie müssen dabei berücksichtigen, dass der Markt sich kontinuierlich wandelt und dass Kundenansprüche stetig steigen.

Forschungsprojekt ALADDIN

Im Rahmen des von der EU unterstützten F&E Projektes ALADDIN, beteiligen sich zehn Partner entlang der Wertschöpfungskette an einem Vorhaben, welches versucht, die neuesten Entwicklungen der LBS im Rahmen einer integrierten und anwendungsspezifischen Lösung zu testen. Das Projektkonsortium besteht aus zehn Institutionen, die sich die Arbeit für drei Testdestinationen aufteilen: Stockholm (SE), Budapest (HU) und Kufstein (A). Ziel des Projektes ist die Entwicklung einer mobilen Lösung, welche einerseits die Arbeitsapplikationen für Incoming-Reiseveranstalter und andererseits die Dienstleistungen und Informationen einer Destination mit Location Based Services vereinigt. Die Untersuchung unterschiedlicher Szenarien umfasst die Nutzung einer solchen Lösung für Reiseleiter vor Ort, für Tourismusorganisationen, aber auch für den Endkonsumenten (Tourist). Die untenstehende Abbildung liefert einen Überblick des „ALADDIN -Modells“ und stellt die technischen Komponenten übersichtlich dar.



Aus der Sicht des Konsortiums umfasst eine ideale mobile Applikation für Destinationen drei Komponenten:

1. Das mobile Arbeitsumfeld für die lokalen Incoming-Gesellschaften und Reiseveranstalter (mobile workspace), das gleichzeitig als Erfassungsplattform für die Points of Interest (POIs) und ortsbezogenen Veranstaltungen dient.
2. Mobile Betreuung des Touristen im Rahmen der von ihm nachgefragten Informationen und Leistungen (mobile leisure environment). Hier werden kontextabhängige Dienste auf Basis der Ortes, der Zeit und der Interessenssituation der Touristen angeboten
3. Die Integration der beiden Bereiche, welche sich eines 'Identitätsmanagements' des Touristen bedient, sodass der Gast einerseits bequem alle Informationen aus einer Hand erhält und andererseits sein Datenschutz gewährleistet wird.

Auf diese Weise integriert Aladdin eine Reihe von Applikationen und Endgeräte, auf einer gemeinsamen Plattform, welche dem Anbieter und dem Gast zugute kommt. Speziell zu erwähnen ist die Einbindung von Mehrsprachsystemen (multi-lingualism services). Kontextsensitive Dienste bauen auf den ortsabhängigen Services auf und berücksichtigen zusätzlich zum aktuellen Standort auch die sich verändernden Interessen eines Anwenders. Wer mobil ist, schlüpft immer wieder in unterschiedliche "Rollen" und tritt als Geschäftsreisender, als Kunde in einem Kaufhaus oder als Tourist auf.

Szenario

Folgende ALADDIN-Szenarien sind möglich [Sti05]: „*Herr Kruppke hat ein Flugzeug früher nach Faro genommen: Die Nachricht erreicht den Incoming-Agenten Simao im Auto auf seinem Pocket-PC - samt einer aktualisierten Übersicht der weiteren Neuankömmlinge. Kruppke ist Golfan, erinnert den Handheld den Agenten: Also gleich fragen, ob der Gast auch an Portugals Südküste einlochen würde. Wenn er mag, kann er direkt bei Simao buchen. Der mobile Minirechner verfügt über ein Kreditkartenlesegerät*“. Der ALADDIN Dienst geht folgendermaßen vor. Der Dienst basiert auf geocodierten POIs (z.B. Notdiensten in der Umgebung, Veranstaltungen, Informationen über Diskotheken etc.) und ist abgestimmt auf den hinterlegten Kontext des Nutzers. Bereits bei der Buchung wird der Kontext des Kunden erfasst, die Struktur der Kontextdimensionen ist maßgeblich von der Anwendung abhängig und kann nicht generisch abgebildet werden. Aus diesem Grund werden für die Kontextdimensionen einige Kriterien beschrieben, die zur Klassifizierung der Merkmale herangezogen werden können. Dies sind lokale Kriterien, Zeitkriterien, physikalische Kriterien und infrastrukturelle Kriterien. Der Kontext beinhaltet in diesem Anwendungsfall Informationen über die Reisezeit, Reiseort, Hobbys, Interessen, Lebens- und Reisesituation des Touristen. So können mit Zustimmung des Reisenden kontextspezifische Angebote mit dem Schwerpunkt der Reisedestination zugestellt werden. Dies sind beispielsweise Angebote zu regionaltypischen Freizeitveranstaltungen, Einladungen zu Sport- und Freizeitveranstaltungen, Event für Singles oder Familien (falls im Kontext so definiert).

Die Angebote werden direkt

Alle Anwendungsbereiche und Dienstleistungsangebote von ALADDIN zeigen einen hohen Grad an kontextabhängigen Speziallösungen (z.B. Events und Marktveranstaltungen für die Innenstadtbelegung der Touristendestination etc.) mit hochwertigen Inhalten und Angeboten der wichtigen Akteure. Dies sind vor allem:

- Lokale Akteure im öffentlichen Sektor (Bürgermeisterämter, CityManager, Tourismusmanager, Städteplanung und -entwicklung, kommunale Informationsstellen, Infrastrukturinstitutionen wie Verkehrssysteme etc.)
- Lokale Akteure des privaten Sektors (Beherbergungs- und



Gastronomiewesen, Einzelhandel, Reiseveranstalter und Reisebüros, privater Personentransport etc.)

- National und international tätige Institutionen und Unternehmen (Touristikkonzerne, Außenstellen von Tourismusmarketingorganisationen Reiseveranstalter etc).

Dabei werden die Touristendaten, Tourdaten, Profile über die Standardschnittstellen SOAP/XML importiert bzw. synchronisiert und anschließend mit der Geocoding Engine zur späteren Verwendung geocodiert.

Zukunftsperspektiven der „Location Based Services“

„Location Based Services“ bieten zahlreiche Entwicklungsmöglichkeiten. Immer leistungsstärkere Handys und Handheld-Computer erhöhen die Verfügbarkeit solcher Dienstleistungen für eine breite Öffentlichkeit. In diesem Zusammenhang stellen sich eine Reihe von Fragen für eine optimale Nutzung solcher Services: Welche Herausforderungen stellt der Aufbau derartiger Infrastrukturmaßnahmen an Dienste- und Inhaltenanbieter? Wer sind die Benutzer von „location based services“ und welche Ansprüche stellen sie? Und wie wird sich die zunehmende Verbreitung mobiler Endgeräte auf das Wechselspiel zwischen Anbietern und Benutzergruppen sowie auf die Qualität der Angebote auswirken? Welche technischen Anforderungen bzw. Herausforderungen sind im aktuellen Forschungsfokus (Definition und Strukturierung: LBS (Kategorien, was sind die interessantesten), neue Features wie Luftbilder, was verstehen Endnutzer unter der lokalen Suche), was sind sinnvolle Suchstrategien und wie werden die Nutzer unterstützt (Theasuri und Ontologien), wie kann die Content-Qualität sichergestellt werden (Metadaten, Dubletten, Spezialcontent, Web 2.0 Content), wie kann man Multimodalität sinnvoll einsetzen, welche neue Entwicklungen beeinflussen den Markt (social web etc.)?

Literatur

[Ber05]

Beritelli, P. (2005): Aladdin - Der Weg zum virtuellen Reiseveranstalter? IDT Blickpunkte, N°13, December 2005

[Sti05]

Stirm, P. (2005): Schöne neue Welt. FVW - The Magazine for the German Travel and Tourism Industry (26/05)

[Zip02]

Zipf, A. (2002): GIS und LBS als Grundlage für moderne Tourismusinformationssysteme. Sonderband zum GIS Day 2001. Mainzer Geographische Studien. Mainz.

On the Applicability of Rule-Based Programming to Location Inference

Katharina Hahn, Kirsten Terfloth, Georg Wittenburg, and Jochen Schiller

{khahn,terfloth,wittenbu,schiller}@inf.fu-berlin.de

Department of Mathematics and Computer Science

Freie Universität Berlin

Takustr. 9, 14195 Berlin, Germany

Abstract. Location-based services offer a powerful approach to provide highly relevant information and functionality to a mobile user. Addressing the problem of location inference, we expand the design space by proposing to employ rule-based programming techniques. Based on position coordinates as well as data gathered by environmental sensors, either a precise location or an abstract location class can be deduced. Both types of location may then be utilized to trigger services in an event-centric manner.

1 Introduction

Location-based services (LBS) continue to attract the interest of both academia and industry. The fundamental building block to allow for the usage of LBS on the client side is a subsystem in charge of establishing the current position of a portable device, e.g. by performing a simple mapping from a given tuple of coordinates to a semantically enriched *location*. Data of this kind, along with context- and/or situation-dependent information may be utilized to reason about whether to trigger an LBS of a service provider or initiate service discovery.

A key problem to be solved is to provide a method for specifying what exactly the enriched location, context or situation is. It is worth noting that scenarios are conceivable in which precise information about the current location is not required, since information about the abstract location class is sufficient. Based on the inferred location it must be decided which actions to undertake, i.e. which services to offer to a user, in case a location relevant to the user is detected.

A model on how to define context with the help of sensors on a device and derive a situation by adding semantical information has recently been introduced by [1]. Orthogonally, [2] presents an event-based system with the goal of determining which services are to be taken into account as soon as location information becomes available. The model we present in this paper aims to combine these two complementary features: The FACTS middleware framework [3] can be utilized for both specification and reasoning. Its rule-based programming naturally suits this area due to its inherent event-driven semantics and powerful data abstraction facilities. Additionally, FACTS has been designed with the goal of minimizing resource usage both in terms of memory footprint and processing time. Hence, it is not only a good choice for mobile devices but even more so for highly embedded systems such as wireless sensor nodes.

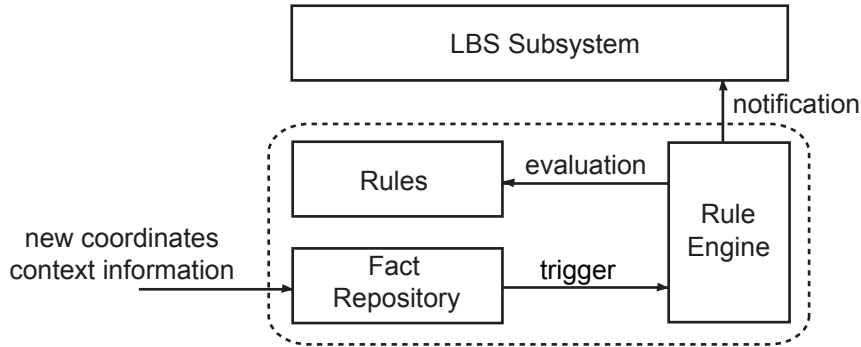


Fig. 1. Interaction between FACTS and the LBS subsystem.

2 A Rule-Based Event-Notification Middleware

The following system model is assumed to allow users equipped with devices that are capable of using wireless communication protocols to employ location-based services: Devices are able to sense their location coordinates, e.g. with a GPS receiver, via GSM cell information, or based on locally deployed RFID tags. Nodes may additionally be equipped with supplementary sensors, e.g. a microphone to detect the noise level or a temperature sensor in order to gather context information of the users current environment.

Using the FACTS middleware, we create a link between LBS, context specification and event notification systems as illustrated in Figure 1. We introduce a hierarchical model of semantically enriched location to aggregate suitable services to sensed context. Those locations may refer to a unique location, e.g. a subway station (*U Hermannplatz, Berlin*) or a specific club (*Goya Club, Berlin*). We also introduce a more abstract notion of location such as a class of location objects, e.g. stations, clubs or universities, that can be used to trigger services or service discovery.

2.1 Precise Location Matching

Initiation of suitable location-based services for a user relies on acquiring knowledge about the current location of the target device. In the following, we assume a location-tuple (longitude, latitude) to be provided by the sensor on the device in question.

The fact repository of the FACTS middleware stores location information concerning *precise location* objects. This precise location object is the result of matching pure geographical data to relevant places, as specified in Listing 1.1. Relevance is defined according to the location-based services which are offered in the environment. Assuming services bound to clubs, the precise location *Goya Club* can be derived from a location information (N52°29.91', E13°21.15') entering the system.

Listing 1.1. Location inference ruleset - Precise location.

```
1 ruleset LocationBasedClubServices
2
3 fact location [long = N52.29.91, lat = E13.21.15, place = "Goya□Club"]
4 fact location [long = N52.30.62, lat = E13.24.97, place = "Sage□Club"]
5
6 rule updateLocation 100
7 <- exists {coordinate}
8 -> retract {currentLocation}
9 -> define currentLocation [place = {location place
10   <- eval ({this long} == newCoordinateLong)
11   <- eval ({this lat} == newCoordinateLat)
12 }]
13 -> retract {coordinate}
```

Listing 1.2. Location inference ruleset - Abstract location.

```
1 fact context [place = "Goya□Club", type = "night□club"]
2
3 rule updateContext 90
4 <- exists {currentLocation}
5 -> retract {currentContext}
6 -> define currentContext [type = {context type
7   <- eval ({this place} == {currentLocation place})
8 }]
```

2.2 Abstract Location Matching

Either by precise location information or by means of gathering data from additional environmental sensors, so called *abstract locations* can be inferred. These describe the generalization of precise location objects and aggregate features of all derived precise location classes. A rule-based implementation of the process of generalizing an abstract location from a precise location is given in Listing 1.2.

The abstraction class of the example denoted above, is the class *night club*. Services bound to this abstraction class include for example a peer-to-peer dating service which finds matching people within the surrounding according to a pre-defined profile. The microphone of a mobile node can be used to affirm the context “within a club”, and thus the abstract class *night club* if no precise location object is available.

2.3 Inheriting Rules of Super Location Classes

The distinction between precise and abstract location classes is made in order to simplify the coupling of services to locations. All precise location classes inherit the services of their abstract classes. Therefore, it is not required to define rules for several instances which are all sub-nodes of the same abstract class. A sample inheritance hierarchy is depicted in Figure 2.

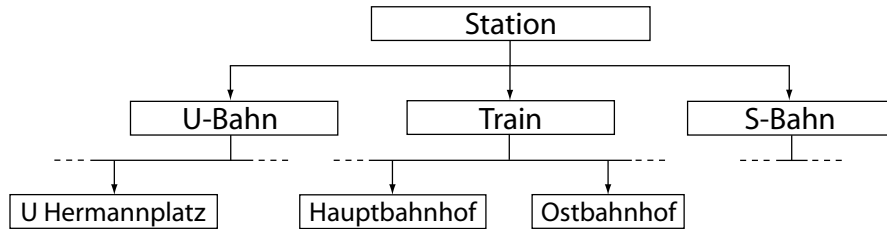


Fig. 2. Location object hierarchy.

When being at a train station in Berlin and therefore within the abstract class *Station*, rules may for example be defined that trigger services to find a ride or to find people who are willing to share their ticket. Within the precise location class *Berlin Hauptbahnhof* additional rules to activate services which display time tables of trains or a map of the station are triggered.

2.4 Service Trigger

By means of the steps defined in the previous subsections, we are able to easily trigger service usage according to the whereabouts of a mobile user. Services are activated through previously defined rules within the ruleset. An application designer can simply add a rule to the ruleset in order to define new triggers for any situation.

3 Conclusion

We propose leveraging the benefits of rule-based programming to ease the task of location inference. To this end, we differentiate between precise and abstract locations depending on available input data, which includes both position coordinates and data gathered by environmental sensors. Upon detecting a location, the middleware may trigger a service in an event-centric way. Adaptations to service notification or expansion of their domain is simply done by adding new rules to the ruleset.

References

1. Meissen, U., Pfennigschmidt, S., Voisard, A., Wahnfried, T.: Context- and Situation-Awareness in Information Logistics. In: Proc. of the International Conference on Extending Database Technology (EDBT'04) Workshops, Heraklion, Crete, Greece (2004)
2. Hinze, A., Buchanan, G.: The Challenge of Creating Cooperating Mobile Services: Experiences and Lessons Learned. In: Proc. of 29th Australasian Computer Science Conference (ACSC 2006), Hobart, Australia (2006)
3. Terfloth, K., Wittenburg, G., Schiller, J.: Rule-Oriented Programming for Wireless Sensor Networks. In: Proc. of the International Conference on Distributed Computing in Sensor Systems (DCOSS'06)/ EAWMS Workshop, San Francisco, USA (2006)

Challenges in Information Systems for Disaster Recovery and Response

Holger Kirchner¹, Thomas Risse¹
{kirchner@ipsi.fhg.de

¹Fraunhofer IPSI - Integrated Publication and Information Systems Institute, Darmstadt, Germany

Abstract. Disaster recovery and response require a timely coordination of the emergency services. IT-technology provides a tremendous potential to increase efficiency and effectiveness in this area by propagating information efficiently to all the right locations. In this paper we sketch requirements and describe challenges for an integrated disaster management communication and information system, addressing in particular networking, configuration, scheduling and data management issues.

1 Introduction

Dealing with disasters such as earthquakes, floods, to name just a few, poses an ever-present challenge to public emergency services. Their success and efficiency depends on a few aspects: up-to-date information being propagated up and downstream efficiently, effective resource management, and a well-organized cooperation and coordination between the different services.

Up-to-date information needs to be available in real-time for the decisions makers. An integrated communication and information system for easy plug-in into other systems will establish a disaster platform for the reliable and secure exchange and processing of information.

In this paper, we sketch a system that allows for horizontal and vertical information flow from the staff on the scene up to central operations by way of a multi-level wireless voice and data communication infrastructure, as well as integrated applications. By combining network solutions such as trunked radio or satellite technology, wireless LAN ad-hoc networks, and personal or body area networks we will provide at all levels for recording and analysis of the current situation, semi-automatic data aggregation and de-aggregation, resource scheduling, as well as access to services and information sources and sinks.

Usually information systems for disaster response are divided into three phases: the pre-phase addressing the preparations before, the post-phase analyzing what happened during the disaster (lessons learnt e.g. for training) and the phase in between, i.e. the situation during the emergency, our main focus.

The structure of this paper is as follows: Following this introduction, we provide a brief user requirements' analysis and, in section 3, an architecture outline. Section 4 addresses applications and information flow. section 5 discusses (auto-)configuration. Sections 6 address data

management issues, respectively. Section 9 concludes this paper.

2 User Requirements Analysis

The Fraunhofer Gesellschaft conducted a major study on disaster and emergency management systems [N]. Apart from communication and information management, the following areas were addressed: optimization and simulation, decision support, visualization, geographical information systems, and simulation and training. One of the findings was that maintaining communications is the "primary challenge" during a disaster and that the following major requirements were not yet met in a satisfactory way:

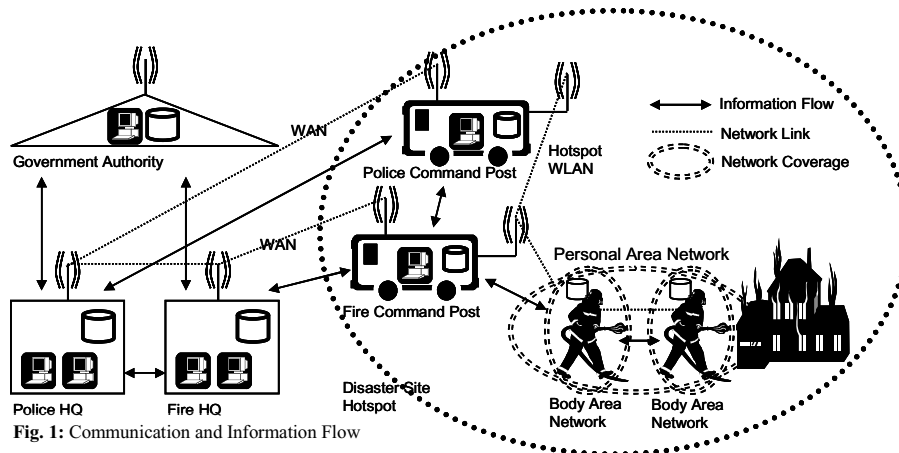
- Integration and linking of information
- Availability of communication, redundancy of links
- Fast data access
- Timeliness and updating of information
- Standardization of information

We refer to the MESA project [M] for requirements in the area of mobile communication standardization.

3 System Architecture

A high-level view of the proposed system's communication architecture is provided in Figure 1. The different emergency services' headquarter buildings are connected to each other and government authorities, e.g. the state governor or FEMA, by terrestrial and/or satellite networks. Likewise, when disaster site command posts are established, they are connected by terrestrial wireless or satellite links to the respective HQs. For "hot spot" on-site communications, a wireless LAN (infrastructure, ad hoc, or both) is set up. Firefighters and other emergency personnel may be equipped with personal or body area networks, providing connectivity for sensors and terminal displays, thus acting as both data sources and sinks.

The information flow of applications can be both horizontal, i.e. between peer entities, and vertical, i.e. along an organization's hierarchy and beyond; both push and pull information propagation are supported.



4 Applications and Information Flow

In this section, we take a bottom-up approach and first describe how frontline personnel like firefighters or other rescue workers operating in difficult terrain may benefit from the proposed system. Firefighters' equipment often includes sensors and detectors, e.g. for radiation or explosive gases. The readings are traditionally transmitted by voice communications to squad leaders. More immediate and reliable data transmission can be accomplished using smart sensors linked, via networks, to a computer in the squad leader's vehicle, where they are immediately analyzed and put into context. In some cases, data may be coupled to the location of the measurement using portable GPS receivers, so, for example, danger zones can be determined more precisely. Moreover, vital parameters of the emergency personnel can be transmitted continuously for monitoring purposes [M]. Thus firefighters are data sources in our system, but they are also data sinks: Messages, hazmat warnings, maps, even data on missing persons may be transmitted to their robust mobile devices. Of course, offline operation capability is a must in a difficult communication environment.

The hierarchy levels of all organizations involved in a rescue effort need to interact closely. They correspond to aggregation levels at which gathered data is analyzed, put into context, and transformed into reports (upstream) and instructions (downstream). At the same time, decisions often have to be taken in an ad hoc manner, which requires efficient access to current information, or, in some cases, an online "help desk". Our proposed applications allow the data to be correlated with other information, to be aggregated or de-aggregated, and to be exchanged with others according to an information flow model deemed appropriate for the situation. A workflow system using templates and taking the involved organizations and the number of hierarchy levels into account, easily adapts to changing organizational structures and facilitates collaborative work within and across services.

Headquarter staff is often responsible for scheduling and coordination and acts as an interface to other agencies and the public. They are, due to their physical distance to the disaster site, particularly dependent on the up-to-date information inflow. On the other hand, HQs usually have vast amounts of stored data, e.g. on hazardous materials, which may need to be accessed by on-site personnel. This calls for integrated applications building on wide area data links between the HQs and site command posts. If a disaster spreads, even HQs may need to be relocated, or operations directors may decide to move closer to the scene, so it is vital to provide a "portable information environment" ready for relocation. This places additional requirements on databases and cooperative environments provided for HQs.

5 Configuration

We have so far argued that the proposed system has to be able to manage vast amounts of data at all levels. Exchanging data in real time between the right entities is a key challenge. The information flow must be controlled in such a way that during the disaster the system is robust and ready to be extended or (re)configured easily. This section shows that these requirements call for auto- and self-configuration of the services in the network.

5.1 Motivation for Auto-Configuration

Without the proper configuration of hosts in networks, they will not be able to find or communicate with each other. Device configuration is therefore of utmost importance. This can be done either statically or dynamically. Devices connected permanently to an administered network are usually assigned static network configuration parameters [HK]. Not permanently connected devices can use dynamic network configurations. The devices themselves must be configured, i.e. all the necessary parameters must be assigned to the host (device) by a network configuration service. The network configuration service itself also requires configuration. However, in a communication and information system aimed at

disaster recovery, manual administration of network hosts is impractical or impossible. Hence, automatic configuration of the hosts is desirable.

5.2 Communication Spheres

As previously suggested, there are three kinds of actors in the system with regard to their degree of mobility:

- Stationary actors: Police, fire, etc. HQs, government authorities, and even foreign authorities or organizations (in case of disasters affecting several countries)
- Semi-mobile actors: mobile command posts.
- Mobile actors: frontline personnel, e.g. firefighters.

The communication can be seen as a “network in network” hierarchy as well. The mobile actors may have several sensors communicating with a handheld device. In essence, these gadgets form a small wearable, or body area, network. The communication to the nearest mobile actors can occur in a personal area network.

As discussed in section 4, the disaster site is covered by a WLAN, operated by semi-mobile actors, who, in turn, can communicate with stationary actors through a WAN.

5.3 Configuration of Devices

One aspect of the configuration of mobile devices is addressing parameters. Data sinks as well as data sources must know with whom to communicate, i.e. the device interface must be configured with a unique address. Duplicate address assignments must be detected, and message collisions must be managed.

Typically, a mobile actor will operate (either manually or automatically) sensor devices, which gather different types of data. Some data, like the amount of explosive gases in the air, is relevant both for him personally as well as for the command post. Other data, such as positioning information is not crucial for the mobile actor, but rather for his superiors at the command post.

Similarly, the mobile actor may want to access a mapping catalogue at his own and/or another command post. Or, he may want to obtain readouts of the vital parameters of a fellow mobile actor. This scenario repeats itself throughout the whole hierarchy depicted above. In essence, there will be a large number of services available offering different kinds of information or connection possibilities.

5.4 Discovery of Services

The discovery of these services can be managed in various ways. At the top hierarchy layers (stationary and semi-mobile actors), the service information management can be regulated using a stationary service trader. Actors offering services could register them in the service trader, as well as poll it for available services.

For the mobile actors, a dynamic process for starting and operating a (mobile) service trader is an option. To

cope with the extremely dynamic situation, services listed in this mobile service trader have a significantly lower registration lifetime, in order to correctly depict the network state. Alternatively, if the number of mobile actors currently in contact is beneath a certain threshold value, no service trader will be assigned at all. In this case, service information can be managed and exchanged through the use of multicast.

As sensor devices are turned on and off and different actors come in and out of reach of each other, the list of available services for each individual actor will most probably be changing constantly. This fact, in conjunction with the amount of diverse information services involved, suggests a system that *dynamically* and *automatically* configures itself. I.e. the devices must on a stand-alone basis find out which services are available, which other actor to send its readings to, present and describe its own services to other devices in the vicinity, and so on. As much as possible of the configuration and managing of services must be automated to facilitate the tasks of the various actors. The goal is to present the right data to the right actor at the right time with as little human intervention as possible.

Moreover, an autoconfiguration-enabled system can also manage the load on the network, e.g. through directing traffic to alternative command posts. In this manner, communication “bottlenecks” can be avoided, which otherwise could prove fatal if occurring at the wrong time. We will come back to this issue in sections 6 and 7.

6 Data Management

Distributed applications for disaster management have to deal with unreliable communication environments, low data transmission rates, and different processing and storage capabilities of the devices used. Hence good communication quality cannot be guaranteed. On the other hand, decisions in the command posts are based on information received from people working in the critical area. Vice versa, people in such areas act on instructions given by the headquarters. For both sides it is hence important to get complete information, as incomplete information delivery can result in wrong decisions or actions. Furthermore, decisions have to be made quickly. This means that information and instructions have to be delivered fast. Thus, as outlined in the following subsection, the main challenges for data management in mobile and unreliable environments, especially in disaster situations, are reliability and performance.

6.1 Challenges

Reliability means that the user always receives complete information of the highest possible timeliness. Incomplete information has to be detected and, if possible, must be requested again. Otherwise the application or

user has to be notified about the transmission failure. In less critical situations it might be possible to reuse information from previous transmissions.

The second important factor is the *performance* of the system. Beside the application, the response time of the system depends on the bandwidth of the communication channel. Hence low transmission rates make it difficult to deliver e.g. complex maps within short time. The data management has to guarantee the efficient usage of the communication channels. Moreover, the response time of the system should be largely independent of the number of communication partners in the system.

Furthermore, the data structures used for data exchange must be flexible in different ways. Sensors can provide their data in proprietary formats. Hence they have to be transformed at some point to the standard data structure. This can be done at the receiving device or, if the device is not able to do this, the data has to be encapsulated into a standard message and transformed at some other point. In addition, the compatibility of different versions of data schemas has to be handled. Data schemas can change over time if new versions of an application are developed, but it is important that devices with different application versions still be able to communicate.

6.2 Suggested Approach

To overcome the previously identified problem, XML [X] should be used as a standard data interchange format. XML documents can contain all required information – from simple messages to complex maps. Furthermore it is flexible in the handling of evolving data structures. A disadvantage of XML might be long tag names and white space, increasing the document size. But the intelligent selection of tag names in the design phase and compression will reduce the document size significantly.

The efficient usage of communication channels is based on a continuous and balanced transmission of data to avoid communication peaks. Hence intelligent caching, pre-fetching and selection of XML documents are the core technology for the implementation of mobile data management. Caching allows for effective usage of communication bandwidth by avoiding retransmission of mostly static information. Intelligent pre-fetching and selection strategies are used for timely delivery of complex location aware information, e.g. maps of buildings.

The reliability of the system is increased by redundant storage of XML documents on different devices (peers) [G]. Hence information has to be replicated in a peer-to-peer manner among nearby devices. This decreases the probability of information loss in the case of a communication failure because other devices can be used as an "information router". Vice versa, in this way it is possible to avoid the loss of important sensor information.

For the proposed technologies data integrity and timeliness of information are important. A distributed transaction management, which is adapted to the special needs in disaster management, ensures that instructions or technical descriptions are completely transmitted to the receiver. Notifications of incomplete transmissions are necessary. The caching strategy has to take the timeliness level of information into account, e.g. static maps have to be updated less frequently than instructions.

7 Conclusion

We have sketched an integrated communication and information system for disaster recovery and response, addressing in particular networking, configuration, data management, and resource scheduling.

In order to make our vision become reality, several IT research disciplines need to work together to realize an effective, yet easy-to-use system that helps emergency services cope with disasters. To name only a few, networking needs to provide robust communications at WAN, LAN, PAN, and BAN levels, integrating heterogeneous networks to allow the rescue effort to proceed smoothly even in the most difficult communication environments. Data management must provide static and dynamic data where and when it is needed. Security is of foremost concern, necessitating solutions for authentication, encryption, data integrity, and non-repudiation. Devices and user interfaces must be tailored to hostile environments and to users who are often not computer literate. Application and information flow designers need to consider fast-changing working environments and resource management is a challenge given the chaotic nature of disasters.

We believe that the IT research community is obliged to respond to the need for technology to support a more informed and coordinated disaster response, and we are convinced that a system such as the one sketched in this position paper would be quite beneficial in this respect.

References

- [H] Auf der Heide, E. Disaster Response: Principles of Preparation and Coordination, Online Book: <http://216.202.128.19/dr>
- [X] eXtensible Markup Language, www.w3.org/XML.
- [G] Gribble, S., Halvey, A., Ives, Z., Rodrig, M., Suiciu, D. (2001). What can Databases do for Peer-to-Peer?, Proc. of the WebDB (Databases on the Web), June 2001.
- [HK] Guttman, E. Autoconfiguration for IP Networking: Enabling Local Communication, IEEE Internet Computing, May/June 2001.
- [M] Mobile Broadband for Emergency and Safety Applications: MESA Project, www.projectmesa.org.
- [N] Nickel, S., et al. (2002). Marktanalyse Katastrophen- und Notfallmanagementsysteme (in German), Fraunhofer Gesellschaft (eds.). www.fhg.de.

Modelling Geo Data for Location-based Services

Jörg Roth
Univ. of Applied Sciences Nuremberg, Germany
Joerg.Roth@FH-Nuernberg.de

1 Introduction

Geo data form the natural resource for location-based services. Geo objects describe the world in terms of natural, artificial and virtual entities that cover the earth's surface. Important functions such as displaying maps, maintaining points of interests (POIs) and navigation strongly rely on the quality of geo data. Spatial data can also be used to improve positioning results (using, e.g., *map matching* in car navigation systems).

The traditional field that deals with geo data is land surveying. As nowadays land survey data is stored digitally, it can in principle be used for location-based services. However, the goals of land surveying often do not meet the requirements of location-based services. As a result, the corresponding data often cannot directly be used.

In this paper, we present a geo data model optimized for location-based services. We identified a number of requirements that the geo model should meet, extended the traditional properties of geo data, which are *thematic*, *geometric* and *topological* properties and introduced *structural*, *organizational* and *meta data* properties.

2 Traditional Geo Models – the ATKIS Case

To discuss geo models, we first have to put them into concrete terms:

- We first have to distinguish *geo models* from *location (sensor) models*. Geo models represent entities in the world, whereas location models represent the position (of, e.g., a mobile user) as a point in space, an area or a probability function. We carefully have to distinguish a static model representation from dynamic position measurement properties, even though mobile objects (such as trains or ships) may be described using both types of attributes. As a third kind, so called *semantic location models* introduce semantics into location information and generate human-readable symbolic locations.
- Second, we carefully have to distinguish geo model *contents* from their *representation*. Often formats such as KML [4], GML [7], G-XML [3] or LandXML [5] are also considered as geo models, but actually they are descriptive languages. Even though the decision for a specific language highly influences the interoperability between communicating components, it does not be a conceptual decision in the first place (as long as the language is expressive enough).

According to Breunig [2], geo data can be divided into a number of geo-objects, each of them representing real geographic entities such as rivers or abstract entities such as city borders. A geo object can be described by an identifier, topological information, its geometry and thematic content. This overall structure is reflected by data from land survey offices as presented in the next section. We analyzed two data sources and exemplarily imported records (some ten thousands) from land survey offices: TIGER/Line files [17] and ATKIS geo data [1]. We describe the AKTIS case in more detail.

ATKIS (*Amtliches Topographisches Kartographisches Informationssystem*, in English: official topographic cartographic information system) provides geospatial data for governmental purposes in Germany. The ATKIS system unites the distributed data of different land survey offices of the German states. Usually, offices using these data have direct access to ATKIS databases, but private persons or companies can purchase parts of the data exported from the database and exported as EDBS data [6].

As a first observation, we could not use the data directly. Even though the data contain expressive geometric information, it has been optimized for printing maps and not for efficient lookup of geo objects as required by location-based services. Objects, e.g., are spread among different records and are cut at borders of the map grid (every km² tile). Due to this and more problems presented below, we created an import and postprocessing workflow (fig. 1):

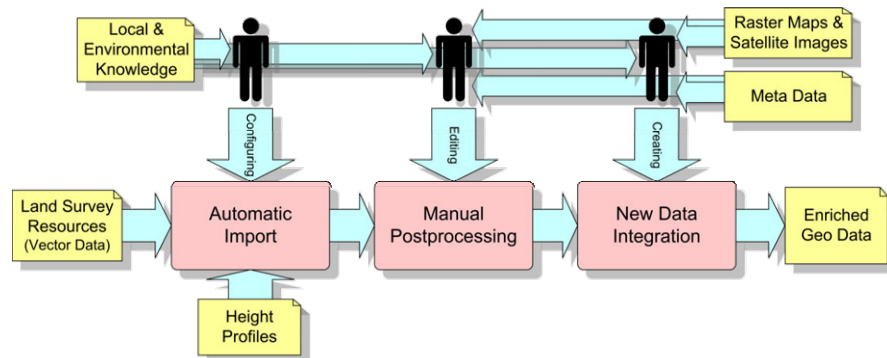


Fig. 1: The import and postprocessing workflow

- An automatic import component parsed and interpreted the original EDBS data format. Related EDBS records have to be merged into single geo objects. Finally, different geometric representations were converted into a single geometric format.
- In a second step (manual postprocessing), we manually merged separated objects that are actually single ones. Typical examples are road objects that are split up at every crossing. This happens because ATKIS prohibits overlapping objects. In addition, we assigned unique names (many objects in ATKIS are unnamed), integrated meta data and classified objects more precisely. The original sources only contain a coarse-grained classification, e.g. universities, schools and kindergartens have the same, meaningless type.
- Third, we integrated new data not contained in the geo data source. E.g., important geo objects such as touristic sights are missing and objects on the same piece of land are often not represented separately. In addition, we integrated *virtual* geo objects such as ZIP zones and GSM cells.

The first step requires manual configuration. Once the configuration is completed, it can be used for similar imports without any new effort. The following steps are more cost intensive as each geo object has to be checked and edited individually. To help an author to perform the second and third step he may use raster data (bitmap maps and satellite images), meta data from different sources (e.g. tourist offices) and should incorporate local and environmental knowledge (which may require visiting geo objects physically).

The experiences of importing geo data from land survey resources can be summarized as follows:

- The import was cost-intensive in three ways. First, geo data is not available for free (in the ATKIS case). Second, development costs arise from the creation of appropriate import filters. Third and most important: cost-intensive manual editing and data integration was required.
- Important thematic information is missing: the classification is not fine-grained enough for locations-based services and some important types of geo-objects are simply missing.

This is because land survey data is mainly collected for property management: In contrast to a location-based service that looks up e.g., a hotel, property management is only interested in any commercial usage of a certain property without further details.

- Geometric information often is inappropriate. Source objects are often artificially cut at the map grid or intersecting objects are divided into different parts. In addition, certain objects are only represented as lines or points without any widths, even though real objects have three dimensions. To perform a test whether a mobile user resides at or in an object, we need at least a two-dimensional ground projection with height levels.
- No or too few meta data is available in the source data. E.g. names are often missing or meaningless and images or links to further resources are missing.
- No or too few topological data is available in the source data. E.g., ATKIS does not express relations between objects and thus topological connections have to be integrated manually.
- Location-based services have additional demands on geo data not fulfilled by the source data. E.g., services may require *distributed* storage and quick retrieval of geo objects. Also improvement of positioning with the help of geo data is currently not supported.

Looking at these drawbacks we introduced a new geo data model that is suitable for innovative services that rely on geo data.

3 Geo Data for Location-based Services – the Hybris Project

Based on the observations above, we want to extend traditional geo data models to be appropriate for location-based services. The *Hybris* framework provides the resulting geo data model with sufficient expressiveness and an authoring environment containing tools for import and editing geo data.

Commercial systems in the area of location-based services often import geo data from hundreds of geo spatial databases [10]. One of our goals was to represent all required geo information within a single data format. To get the required expressiveness, we first extended the list of traditional properties of geo objects (i.e. geometric, thematic and topological properties). The resulting list of properties (including the traditional ones) is:

- *Thematic properties* mainly provide a classification of geo objects. Useful classifications follow an object-oriented approach that identifies classes and subclasses of objects.
- *Meta data* are names, images, sounds, Web links etc.
- *Geometric properties* describe the geometric shape and are used for rendering maps, distance measurements and testing, whether a given location is at or inside a certain object.
- *Topological properties* are used for street navigation and more complex distance measurements (e.g., distance to get to an object using public transportation).
- *Structural properties* are used to store and access geo data in technical infrastructures. Inside the *Nimbus* project [11-15] we use structural properties to perform a quick geometric lookup of geo objects that are stored in a distributed federation of servers. In contrast to topological properties, structural properties describe relations in terms of storage and access, which are not directly perceived by end-users.
- *Organizational properties* are information about the generation process of geo data. They answer questions such as: What is the data source, who modified an entry and how long is the entry expected to be valid? Such data is important to achieve and preserve the quality of geo data. In contrast to structural properties, organizational properties describe non-technical characteristics.

Even though structural and organizational properties are also meta data in a general meaning, we summarize only those properties that describe content perceivable by an end-user as meta data. Inside the Hybris project, we use an XML language to represent geo objects, whereas each geo object is stored in an individual XML file. That simplifies the distribution of geo objects to different servers. Some details:

Thematic properties: Two fields inside a record provide the object classification: First the ATKIS class imported from the original source provides a rough classification. Second, we introduced a more fine-grained and object-oriented classification with currently 305 classes, that can easily be extended.

Geometric properties: We use a 2.5D representation, i.e. objects are projected on the earth's surface and projected points have an altitude property. We distinguish arbitrary areas presented as *multi-polygons with holes* [16] (fig. 2a), *multi-lines* (fig. 2b) and *point-like areas* (fig. 2c). We always store a multi-polygon representation, also for lines and points, using the so-called *buffer* operation, which computes all points within a certain distance to the geometric object. This simplifies the *location in area test* but still preserves point or line information. We, e.g., can simply check if a car is on a specific road (using the multi-polygon), but we also know the course of the road (using the line information).

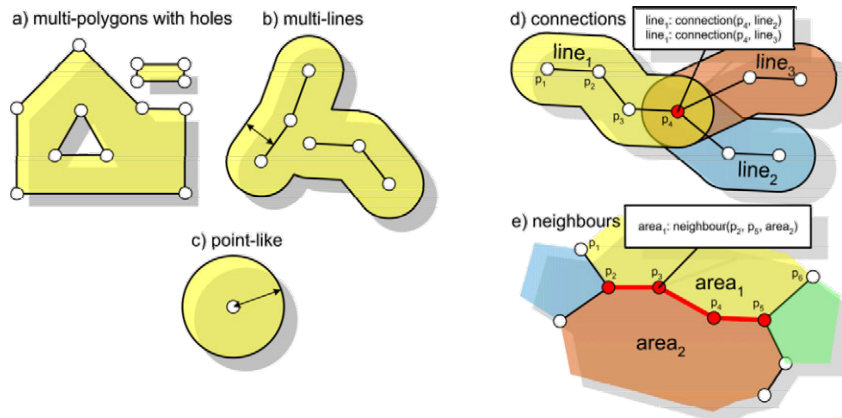


Fig. 2: Geometric and topological properties

Topological properties: Two objects can be topologically connected in one or two dimensions. One-dimensional connections link two objects with the help of a single shared point (fig. 2d). Examples are roads that join in a crossing. Such connections may be unidirectional, to, e.g., express that a road can only be entered from a specific direction. Two-dimensional connections represent common borders, e.g. between neighbouring pieces of land (fig. 2e).

Structural properties: The most important structural property is the *domain name* – a worldwide unique identifier. In contrast to a human readable name as stored inside the meta data, the domain name can be used as a pointer to reference other geo objects (as required to build topological connections). The name in the original sources are often ambiguous, thus we use a hierarchical naming scheme described in [16]. For a fine-grained tuning of the storage mechanism, geo objects additionally have an *area type* (currently one of *stationary*, *mobile*, *temporary* and *abstract*). Mobile areas, e.g., have certain demands on the storage mechanism (e.g. quick update times), thus a server has to know the corresponding characteristics.

Table 1 summarizes the properties and fields in the Hybris geo data model.

Fig. 3 shows the main authoring tool to edit geo objects. The different properties are reflected by different views on the data. The tool allows searching and editing entire geo object trees. It is planned to implement a plug-in interface to build future extensions (e.g. refactoring and rule-checking functions) without affecting the tool core.

Table 1: Hybris geo data properties

Attribute	Description	Values/Example
Structural:		
domain	worldwide unique ID	hierarchical name string / FH_Nuernberg.Woehrd.nuernberg.de
areatype	structural area type	{stationary, mobile, temporary, abstract}
association/ relation	Nimbus association and relation [16]	other domain name / Pegnitz.rivers.geo
Geometric:		
geometry	geometry type	{pointarea, linearea, polyarea}
pointarea	point-like areas	single WGS84 point / C(4927.1782,N,01105.5404,E,300.0m/3)
linearea	lines	multiple line strings / L(4927.1782,N,01105.5404,E,300.0m...)
polyarea	arbitrary 2D areas	multiple polygons with holes / P(4927.1782,N,01105.5404,E...)
lowerbound/ upperbound	2.5D layer	altitude value[m] / 300
captionpoint/ captionangle	point/angle to display the object's name	single WGS84 point 4927.1782,N,01105.5404,E,300.0m angle[°] / 45
Topological:		
connection	lines are connected	tuple (point, other line ID)
neighbour	define a shared border	tuple (startpoint, endpoint, other area ID)
Thematic:		
lsiclass/ atkisclass	object classification	LSI/ATKIS class ID / 22120, 2114
Meta Data:		
name	real object name	UTF-8 string / FH Nürnberg
image	Image file	URL / http://www.fh-nuernberg.de/image.jpg
link	Web link	URL / http://www.fh-nuernberg.de
Organizational:		
expiration	expiration time	duration[min] / 1440
author/modified/ approved	who created/modified /approved the entry	URL or email address / Joerg.Roth@wireless-earth.de

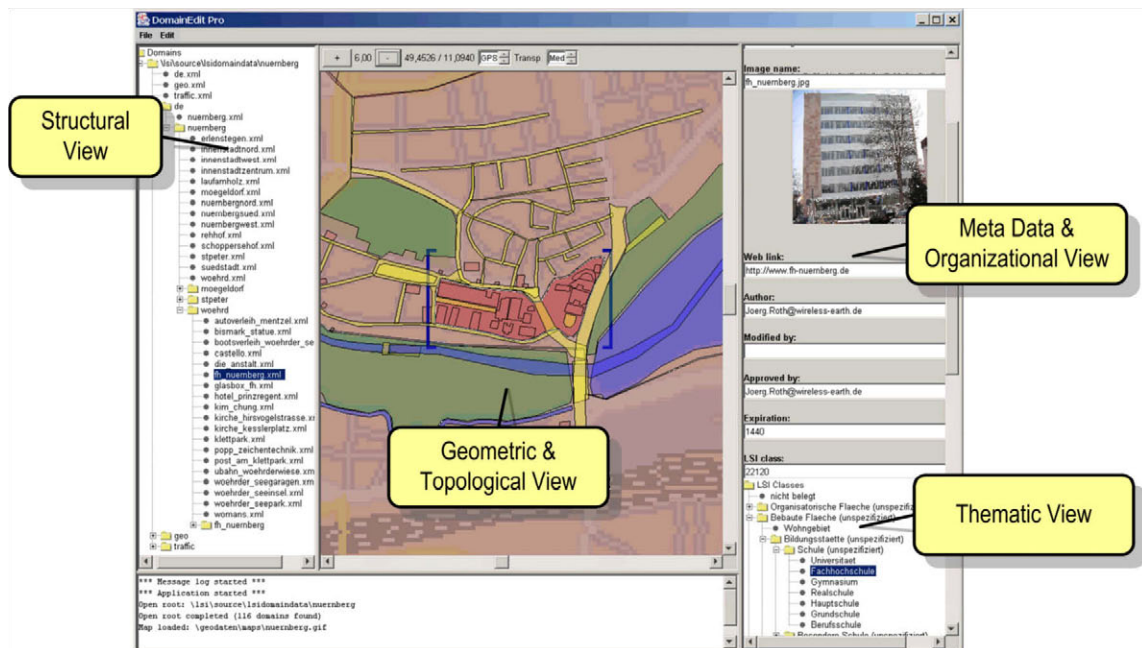


Fig. 3: The Hybris authoring tool

4 Conclusion and Future Work

This paper presents an optimized geo data model for location-based services. Experiences of the import of land survey data have been considered. As an imported observation, geo data of traditional sources have certain drawbacks and cannot directly be used in location-based service scenarios. Import and adaptation is cost-intensive. In particular, structural, organizational and meta data properties have to be added manually.

In the future we want to introduce certain extensions. First, we want to model unsharp borders as necessary to, e.g., model WLAN cells or certain parts of a city (e.g. its centre) which have a fuzzy border rather than a certain geometric limit. The corresponding geometric property has to have a compact representation, but at the same time should allow the execution of efficient geometric lookup operations.

As a second direction we want to generate huge sets of geo data without using land survey sources. In environments such as Wikipedia or Google Earth users make contributions without a forceful central editor. To follow this idea, we want to establish a Web-based workflow that allows arbitrary users to contribute geo data knowledge.

References

1. Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV): ATKIS Homepage, <http://www.atkis.de/>
2. Breunig, M.: *Integration of Spatial Information for Geo-Information Systems*, Lecture Notes in Earth Sciences 61, Springer, 1996
3. Database Promotion Center: *G-XML Project Homepage*, <http://www.dpc.jipdec.jp/gxml/contents-e/>
4. Google Earth, *KML Specification*, <http://earth.google.com>
5. Land Development and Transportation Professionals: *LandXML Project Homepage*, <http://landxml.org/spec.htm>
6. NÖV – Nachrichten aus dem öffentlichen Vermessungsdienst Nordrhein-Westfalen, Vol. 2-3, 1987
7. Open GIS Consortium: *OpenGIS Geography Markup Language (GML) Implementation Specification*, www.opengis.org, 2004
8. Roth, J.: *Mobile Computing*, dpunkt-Verlag, Sept. 2005
9. Roth, J.: *Data Collection*, in Jochen Schiller, Agnès Voisard (eds), *Location-Based Services*, Morgan Kaufmann Publishers, May 2004
10. Roth, J. (ed): 1. GI/ITG KuVS Fachgespräch "Ortsbezogene Anwendungen und Dienste", June 24-25, 2004, Hagen, Informatik Bericht 317, University of Hagen, June 2004
11. Roth, J.: *Semantic Geocast Using a Self-organizing Infrastructure*, Innovative Internet Community Systems (I2CS), Leipzig (Germany), June 19-21, 2003, LNCS 2877, Springer-Verlag, 216-228
12. Roth, J.: *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
13. Roth, J.: *The Role of Semantic Locations for Mobile Information Access*, Sept. 22, 2005, Bonn (Germany), Proceedings of the 35th Annual GI Conference, Vol. 2, 538-542
14. Roth, J.: *The Distributed Location Resolution Problem and its Efficient Solution*, IADIS International Conference Applied Computing 2006, San Sebastian (Spain), Feb. 25-28, 2006, 3-10
15. Roth, J.: *Detecting Identifiable Areas in Mobile Environments*, The 21st Annual ACM Symposium on Applied Computing, Apr. 23-27, 2006, Dijon, (France), ACM Press, 986-991
16. Roth, J.: *A Decentralized Location Service Providing Semantic Locations*, Habilitation Thesis, University of Hagen, Computer Science Report 323, Jan. 2005
17. U.S. Census Bureau: 108th Congressional Districts Census, TIGER/Line Files, <http://www.census.gov/geo/www/tiger/index.html>, 2006

Implementierungsbericht Mobile Hunters

Jörg Lonthoff, Erich Ortner, Michael Wolf
Fachgebiet Wirtschaftsinformatik I – Entwicklung von Anwendungssystemen
Technische Universität Darmstadt
Hochschulstraße 1, 64289 Darmstadt
{Lonthoff, Ortner}@winf.tu-darmstadt.de
Michael.Wolf@briefkasten.us

Abstract

Mobile Hunters ist ein mobiles ortsbezogenes Abenteuerspiel, bei dem mehrere Jäger versuchen, einen Flüchtenden zu fangen. Der Flüchtende kann auf seiner Flucht Unschuldsbeweise finden, die ihm auch eine Gewinnchance ermöglichen. Die reale physische Ortsveränderung der realen mit Mobiltelefonen ausgestatteten Mitspieler veranlassen dabei die virtuellen Bewegungen der virtuellen Spielfiguren auf den Displays der Mobiltelefone. Als Lokalisationsmethode wird Cell of Origin eingesetzt, was zu Implementierungsproblemen führte, da das Auslesen der Cell-ID aus der J2ME-Umgebung nicht direkt möglich ist. Dieser Beitrag zeigt eine mögliche Lösung dieses Problems und zeigt mögliche Code-Optimierungen bei der Entwicklung von Anwendungen auf mobilen Endgeräten.

1. Mobile Hunters – Ein Mobile Location-Based Game

Mobile Hunters [LoOr06] ist ein mobiles ortsbezogenes Abenteuerspiel, bei dem mehrere Jäger versuchen einen Flüchtenden zu fangen. Es müssen mindestens zwei Personen teilnehmen, um eine entsprechende Spielesession eröffnen zu können. Im Spiel existieren zwei Rollen: ein oder mehrere Jäger und ein Flüchtender. Die Jäger haben das Ziel, den Flüchtenden vor Ablauf einer festgelegten Spieldauer (Standard 30 Minuten) zu verhaften. Der Flüchtende muss jeweils den Jägern entkommen oder seine Unschuld durch Aufsammeln einer bestimmten Anzahl von Unschuldsbeweisen belegen.

Eine Person kann ein Spiel anlegen, indem sie sich mit Benutzername und Passwort beim Spieleserver anmeldet. Diesem Spiel können weitere Personen als Mitspieler beitreten. Dabei wird eine Abstandsabfrage durchgeführt, die ein zu großes Spielfeld verhindert. Die Person, die das Spiel angelegt hat, entscheidet, wann sich ausreichend viele Mitspieler gefunden haben und startet das Spiel. Die Maximalanzahl (Standard 10) der Mitspieler kann eingestellt werden.

Ist das Spiel gestartet, übernimmt der Spieleserver zunächst eine zufällige Rollenverteilung und teilt den Mitspielern mit, ob sie Jäger oder Flüchtender sind. Weiterhin verteilt der Spieleserver bei Spielstart alle Gegenstände für Jäger und Flüchtenden nach einem Zufallsprinzip auf dem Spielfeld. Nach diesem Initialisierungsprozess beginnt das Spiel synchron auf allen beteiligten Clients, und der Countdown der Spieldauer läuft. In festgelegten Intervallen (Standard 3 Minuten) erscheint die aktuelle Position des Flüchtenden eingebettet in eine Straßenkarte auf den Displays der Jäger. Auf dem Spielfeld sind verschlossene (virtuelle) Kisten verteilt, die von den Spielern geöffnet werden können, wenn sie auf der gleichen Position (Geokoordinate mit festgelegtem Radius) sind, auf der sich die Kiste befindet. Es gibt Kisten, die nur für die Jäger sichtbar sind. Diese Kisten enthalten Angriffswaffen. Für den Flüchtenden sind nur Kisten sichtbar, in denen sich Abwehrgegenstände und Unschuldsbeweise für ihn befinden.

Ein Angriff kann dann erfolgen, wenn sich die Mitspieler auf der gleichen Position befinden und der Angreifer eine Waffe besitzt. Greift ein Jäger einen anderen Jäger an, wird der Angegriffene für eine bestimmte Zeit (Standard 30 Sekunden) handlungsunfähig, indem das Menü seines Clients ausgeblendet wird. Bei diesem Angriff verliert der Angreifer seine Waffe. Greift ein Jäger den Flüchtenden an, kann sich der Flüchtende mit einem entsprechenden Abwehrgegenstand wehren. In diesem Fall ist der Angreifer für eine bestimmte Zeit (Standard 30 Sekunden) handlungsunfähig. Konnte sich der Flüchtende nicht verteidigen, dann hat der angreifende Jäger gewonnen.

Der Flüchtende kann das Spiel gewinnen, indem er entweder eine bestimmte Anzahl (Standard 3) an Unschuldsbeweisen in den Kisten findet oder innerhalb der Spieldauer nicht erfolgreich angegriffen wird.

Am Ende jedes Spieles werden jedem Mitspieler Punkte nach einem vorgegebenen Punkteschema vergeben, die auf dem Spieleserver aufsummiert und gespeichert werden. Dabei erhält jeder Mitspieler eine Mindestpunktzahl, um die Spielmotivation zu erhöhen. Durch einen solchen Aufbau einer Spieler-Community wird ansatzweise der Massive-Multiplayer-Game Gedanke realisiert.

Mobile Hunters reiht sich damit ein in die Kategorie der Mobile Location-Based Games (MLBG). Diese in [LeLo05] vorgestellte Kategorie spezialisiert Location-Based Games als Aggregation von Location Aware Games und Spatially Aware Games [NPM01, S. 61-62] (s. Abb. 1). In Abgrenzung zu „Mobile Games“, die als Ortsbezug nur einen weiteren Mitspieler benötigen, der sich in der näheren Umgebung befindet, haben MLBGs einen Bezug zum Ort des Spielers. Location Aware Games sind Spiele, bei denen die Ortsinformation des Spielers mit in den Spielablauf einbezogen wird. Ein typischer Vertreter ist z. B. eine Schatzsuche, bei der ein Mitspieler einen bestimmten Ort erreichen muss. Spatially Aware Games sind Spiele, bei denen die Umgebung der realen Welt auf das Spiel übertragen wird. Hierbei entsteht eine Verbindung zwischen der realen und der virtuellen Welt.

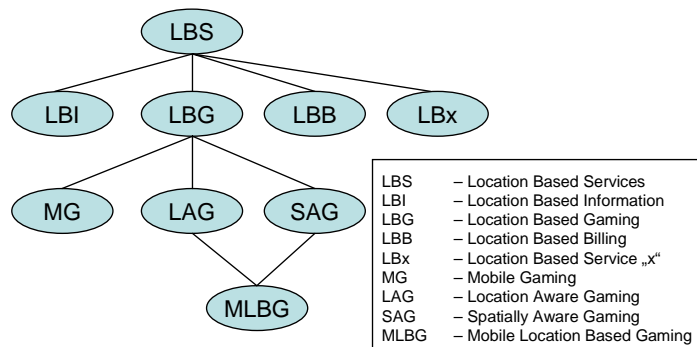


Abbildung 1: Taxonomie Location-Based Services

2. Lokalisationsmethoden

Allen Location-Based Services [SV04] liegt ein Mobile Positioning (Lokalisation) zugrunde. Dabei fasst Mobile Positioning die Technologien zur Lokationsbestimmung von mobilen Endgeräten zusammen. Für die Art der Positionsbestimmung gibt es zunächst zwei Varianten:

- netzwerkbasierte Technologie – das Netzwerk stellt die Lokationsinformation bereit (COO, AOA, TDOA, TOA, RadioCamera, SA [vgl. Ro05, Rö02])
- terminalbasierte Technologie – das Endgerät ermittelt seine Position selbst (COO, E-OTD, GPS, A-GPS [vgl. Ro05, Rö02])

Cell of Origin (COO) ist die einfachste Ortsbestimmung in Mobilfunknetzwerken. Hier wird die Funkzelle (Cell) bzw. die Basisstation, in der das Mobiltelefon eingebucht ist, identifiziert. Hierbei wird eine Positionsgenauigkeit in Abhängigkeit von der Größe einer Funkzelle erreicht, also im Falle eines GSM-Netzwerks ca. 20-100 m in Ballungsräumen und bis zu 50 km in ländlichen Gebieten [Sc03]. COO kann als netzwerkbasierte Technologie aufgefasst werden, da der Netzbetreiber über sein Home Location Register (HLR) jederzeit über die eingebuchte Cell-ID des jeweiligen Mobilfunkteilnehmers verfügt. COO ist aber auch als ein terminalbasiertes Verfahren anzusehen, da die jeweilige Cell-ID direkt aus dem Endgerät (Terminal) ausgelesen werden kann. Hierzu ist allerdings eine Referenzdatenbank notwendig, welche die zu den jeweiligen Cell-IDs gespeicherten Geokoordinaten enthält.

Eine Mobilfunkstation hat einen Abstrahlungsbereich mit dem Radius (r) um die Position (P) eines Mobilfunkmasten. Dabei wird die geografische Position (P) im World Geodetic System 1984 (WGS84)-Format angegeben. Ein Abstrahlungsbereich kann mehrere Zellen (Cell-IDs) beinhalten, hat dabei aber immer die gleiche geografische Position (P). In Abbildung 2 ist ein solcher Abstrahlungsbereich eines Funkmasten in Darmstadt zu sehen.

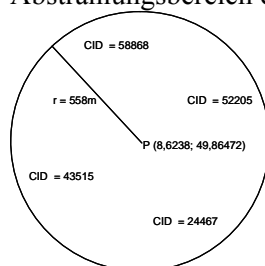


Abbildung 2: Abstrahlungsbereich eines T-Mobile GSM Funkmasten in Darmstadt

3. Systemanforderungen

Um die Akzeptanz des Spiels zu erleichtern, wurden nur solche Technologien in die Entwicklung von „Mobile Hunters“ einbezogen, die bereits akzeptiert und weit verbreitet sind, oder deren Verbreitung relativ wahrscheinlich ist.

Um ein möglichst breites Einsatzfeld zu gewährleisten, wurde auf das GSM-Mobilfunknetz zurückgegriffen, welches derzeit die weiteste Verbreitung findet [NPM01, S. 61] und per GPRS eine geeignete dauerhafte IP-Verbindung ermöglicht. Bei der Wahl des Endgerätes wurde auf den Einsatz zusätzlicher Hardware (z.B. GPS Empfänger) verzichtet und ein Java-fähiges Handy (Nokia 6680) als „Spielgerät“ ausgewählt. Aus den Erfahrungen der ersten Version des Spiels auf einem Siemens SX 1 konnte eine sinnvolle untere Schranke für Prozessorleistung (mind. 200 MHz) und Speicherkapazität (Hauptspeicher mind. 1 MB) ermittelt werden. Als Betriebssystem wird Symbian OS 7.0 in Verbindung mit Mobile Information Device Profile (MIDP) 2.0 und Connected Limited Device Configuration (CLDC) 1.1 vorausgesetzt. Die Mindestauflösung des Displays sollte 126 x 208 Pixel nicht unterschreiten.

Für die Wahl des Lokalisationsverfahrens war es essentiell, dass nur auf die vorhandene Infrastruktur zurückgegriffen wird, ohne daran Modifikationen vorzunehmen. Daher wurde das Cell of Origin (COO)-Verfahren gewählt.

Die Benutzerinteraktion erfolgt Menügesteuert über das Mobiltelefon, um Gegenstände aufzunehmen, anzusehen, abzulegen und einzusetzen.

4. Systemarchitektur und Implementierungshürden

Über ein in Symbian C implementiertes Programm (CellID.exe) wird die Cell-ID aus dem Mobiltelefon ausgelesen und dem Spieleserver übermittelt. Der in Java programmierte Spieleserver läuft auf einem Windows 2003 Server, welcher parallel dazu noch über eine SQL-Datenbank verfügt, auf der die Daten der Spieler und Spiele gespeichert und abgefragt werden können. Dieser Spieleserver stellt das zentrale Spiel-Management dar und ist das Verbindungsglied zwischen den einzelnen Mobiltelefonen. In diese Architektur konnte das Public and Privacy Gateway (PPGW) der T-Systems eingebettet werden, welches die Positionsdaten (Transformation Cell-ID in Geokoordinate) und die Bilder der für das Spiel benötigten Straßenkarten über eine Web Service-Schnittstelle zur Verfügung stellt (s. Abb. 3).

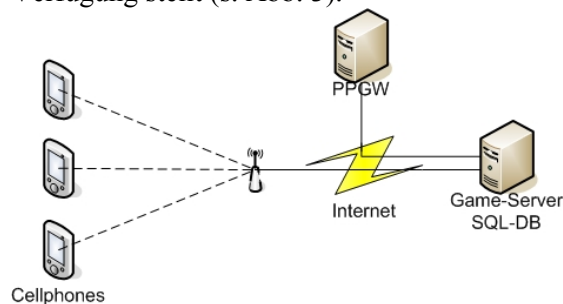


Abbildung 3: Systemarchitektur Mobile Hunters

Alle Anfragen der Mobiltelefone werden an den Spieleserver weitergeleitet, der sich die benötigten Daten von der Datenbank oder vom PPGW holt und diese an die betroffenen Mobiltelefone weiterleitet. Die Nachrichten, welche zwischen Client und Server ausgetauscht werden, betreffen dabei eine Vielzahl von Aktionen, die voneinander unterschieden werden müssen. Aus diesem Grund wird jeder Nachricht ein eindeutiger Opcode (Befehlscode) hinzugefügt, der die Art der Nachricht identifiziert und dementsprechend von Server und Client richtig eingeordnet und verarbeitet werden kann.

Auslesen der Cell-ID unter J2ME

Die größten Probleme bei der Implementierung ergeben sich mit dem Auslesen der Cell-ID. Ein Auslesen über J2ME ist nicht möglich, da in der API kein Zugriff auf Systemdaten vorgesehen ist. Deshalb muss die Cell-ID über ein C++ Programm ausgelesen werden, das direkt auf die Symbian Befehle zugreifen kann. Daraus folgend ergibt sich das Problem der Datenübergabe. Ein J2ME-Programm kann nicht direkt auf ein C++ Programm zugreifen und es benutzen. Ein Lösungsversuch,

bei dem die Cell-ID mit dem C++ Programm in eine Textdatei gespeichert wird, um diese periodisch auszulesen, scheiterte an der Rechtevergabe auf dem Mobiltelefon. Ein J2ME-Programm läuft in einer Sandbox und hat deshalb keinerlei Zugriffsrechte auf das Dateisystem des Mobiltelefons. Als Workaround wurde auf dem Mobiltelefon ein Webserver installiert. Über einen Stream liest das J2ME-Programm die Cell-ID aus und schickt sie an den Spieleserver, der daraufhin eine Umrechnung in Geokoordinaten vornimmt.

In der nächsten Evolutionsstufe des Spiels wurde das C++ Programm so verändert, dass es nach dem Einlesen die Daten selbst über einen Stream an den Spieleserver schickt. Somit konnte die Prozessorlast auf dem mobilen Endgeräte gesenkt werden, was zuvor zu Performanceproblemen geführt hatte.

Auf dem Nokia 6680 ergab sich zudem ein neues Problem: Das Symbian Programm ließ sich zwar installieren, es konnte dann aber nicht über das Menu des Mobiltelefons gestartet werden. Es musste ein zusätzliches Programm installiert werden, das auch .exe-Dateien im Filesystem sichtbar und ausführbar machte.

Übereinstimmung von Emulatoren und realem Endgerät

In Verbindung mit dem SX1 von Siemens ergaben sich große Probleme mit den Emulatoren. Ein für die IDE Eclipse zur Verfügung gestelltes Plugin des Emulators verarbeitete im Gegensatz zum mobilen Endgerät Gleitkommazahlen, sodass es zu Beginn zu unerklärlichen Abstürzen kam. Desweiteren konnten keine Geschwindigkeitsbegrenzungen vorgenommen werden, sodass das Spiel auf dem Emulator sehr viel schneller lief als auf dem mobilen Endgerät. Außerdem war der Emulator nicht in der Lage, einen Zugriff auf das Internet zu simulieren, weshalb zusätzliche Programme auf dem PC installiert werden mussten. Der Nokia-Emulator zeigte diese Schwächen nicht, sodass mit diesem die Implementierung realitätsnäher getestet werden konnte. Dennoch hatte auch dieser einen Fehler. So war es im Emulator möglich, ein .png-Bild aus einer Internetquelle direkt zu laden. Das mobile Endgerät quittierte diesen Versuch allerdings nur mit einer Fehlermeldung, sodass das Bild doch direkt vom Spieleserver geladen werden musste.

Bildübertragung

Bei der Übertragung der Bilder vom Spieleserver zum Mobiltelefon kam es öfter zu Fehlermeldungen auf dem mobilen Endgerät. Ursache hierfür ist eine Störung der Bildübertragung durch eine andere Datenübertragung (zum Beispiel das Einloggen eines anderen Spielers). Die Verbindung muss deshalb reserviert und nach der Übertragung wieder frei gegeben werden.

Abhängigkeiten von Mobilfunk Providern

Im ersten Projekt war es noch möglich, die Mobiltelefone direkt über eine IP anzusprechen, denn jedes Endgerät bekam eine andere IP. Dies war im zweiten Projekt bei einem anderen Mobilfunkprovider nicht mehr der Fall. Dort wurde ein Portal zwischengeschaltet, weshalb auf die IP der Endgeräte nicht direkt zugegriffen werden konnte, da alle angemeldeten Mobiltelefone die gleiche IP zugewiesen bekamen. Deshalb wurden die Endgerät über den Benutzernamen identifiziert und die Verbindung selbst über eine Hashtabelle gespeichert.

5. Code-Optimierungsmöglichkeiten für mobile Endgeräte (Lessons Learned)

Bei mobilen Endgeräten und gerade bei Mobiltelefonen ist der sparsame Umgang mit Systemressourcen bei der Entwicklung mobiler Anwendungen essentiell. Die integrierten Prozessoren arbeiten meist mit einer Taktfrequenz im Bereich von 166 MHz - 300 MHz. Dies ist im Vergleich zu PC Prozessoren sehr wenig.

Optimierungspotenzial Schleifen

Schon bei der Modellierung sollten Schleifen so „klein“ wie möglich gehalten werden, insbesondere dann, wenn viele Iterationen zu erwarten sind. Zusätzlich lässt sich die Laufzeit optimieren, indem Methodenaufrufe grundsätzlich, vor allem aber in Schleifen, so weit wie möglich vermieden werden. Wie dies unter anderem realisiert werden kann, zeigt ein einfaches kleines Code-Beispiel.

```
for (int i=0; i<vector.size(); i++) {  
    [... auszuführender Code...]  
}
```

Code-Beispiel 1: Schleife in einer normalen Java-Umgebung

Hierbei wird nun die Methode `vector.size()` in der Anzahl der Iterationsstufen aufgerufen, was sich angesichts der Prozessorleistung eines Mobiltelefons durchaus bemerkbar machen kann. Vorausgesetzt, die Größe des Vektors ändert sich durch die Ausführung der Schleife nicht, wäre es in diesem Fall besser, `vector.size()` zuvor in einer Variablen zu speichern. Damit muss die Methode nur einmal aufgerufen werden, unerheblich wie viele Iterationen die Schleife durchläuft.

```
int size = vector.size();
for (int i=0; i<size; i++) {
[... auszuführender Code...]
}
```

Code-Beispiel 2: Optimierte Schleife in einer ressourcenbeschränkten J2ME-Umgebung

Dieses vereinfachte Beispiel zeigt anschaulich, dass schon kleine Änderungen im Code die Laufzeit verbessern können.

Caching

Auf mobilen Endgeräten, bei denen die CPU-Zeit sehr begrenzt ist, sollten Methodenaufrufe insgesamt vermieden werden. Dies ist besonders dort einfach zu handhaben, wo solche Methoden Werte zurückliefern, die sich danach nicht mehr ändern. Das Beispiel mit der Länge eines Vektors ist nur eine Möglichkeit in diesem Bereich. Vor allem bei der GUI gibt es einige Möglichkeiten in J2ME z. B. die Breite und Höhe des Displays mit den Methoden `getWidth()` und `getHeight()` auszulesen. Diese Werte werden je nach Spiel recht häufig gebraucht. Es wäre daher nicht sinnvoll, jedes Mal die entsprechende Methode aufzurufen.

```
private void centerDisplay(Koordinate koord) {
x = koord.getX() - (display.getWidth() / 2) + 15;
y = koord.getY() - (display.getHeight() / 2) + 20;
[... auszuführender Code...]
if (x > getMap().getWidth() - display.getWidth()) {
x = getMap().getWidth() - display.getWidth();
}
[... auszuführender Code...]
if (y > getMap().getHeight() - display.getHeight()) {
y = getMap().getHeight() - display.getHeight();
}
[... auszuführender Code...]
}
```

Code-Beispiel 3: Viele unnötige Methodenaufrufe

Sinnvoller ist es, hier alle Werte, die wiederverwendet werden, in Variablen zu speichern und nur diese weiterzuverwenden. Folgendes Code-Beispiel verdeutlicht diese Optimierung:

```
private void centerDisplay(Koordinate koord) {
int displayWidth = display.getWidth();
int displayHeight = display.getHeight();
Canvas map = getMap();
int mapWidth = map.getWidth();
int mapHeight = map.getHeight();
x = koord.getX() - (displayWidth / 2) + 15;
y = koord.getY() - (displayHeight / 2) + 20;
[... auszuführender Code...]
if (x > mapWidth - displayWidth) {
x = mapWidth - displayWidth;
}
[... auszuführender Code...]
if (y > mapHeight - displayHeight) {
y = mapHeight - displayHeight;
}
[... auszuführender Code...]
}
```

Code-Beispiel 4: Optimierte Verwendung von Methodenaufrufen

Obwohl diese zweite Variante einige Zeilen mehr Code enthält, kommt sie mit sechs Methodenaufrufen aus, während bei der ursprünglichen Variante zwölf Aufrufe benötigt werden. Dies ist vor allem umso bemerkenswerter, weil die Methode `centerDisplay()` im Vergleich zu anderen relativ kurz ist.

Bedenkt man, dass zum Beispiel Integers in Java nur vier Byte vom Speicher belegen, sie aber sinnvoll gecached eingesetzt eine große Anzahl von Methodenaufrufen und damit Prozessorzeit sparen, so ist offensichtlich, dass der leicht erhöhte Speicherverbrauch durch den Performancezugewinn mehr als ausgeglichen wird.

Optimierungspotenzial von Arrays

Der Datentyp Array zum Abspeichern und Wiedergeben von Werten ist im Vergleich mit anderen Möglichkeiten, wie zum Beispiel einem Vektor, der schnellste. Daher sollte dieser Datentyp so oft wie möglich anstelle anderer Varianten verwendet werden.

Profiler

Im Zusammenhang mit der Optimierung sollte noch ein weiterer Aspekt betrachtet werden. In vielen Fällen betreffen 90% des ausgeführten Programms während der Laufzeit nur 10% des programmierten Codes. Die Optimierung des Codes sollte daher unbedingt diese 10% des Codes abdecken. Für das Auffinden dieses Teils im Code bietet J2ME Wireless Toolkit eine Anwendung, die sich „Profiler“ nennt. Eine Beschreibung für dessen Nutzung findet sich in [Sh03]. Der Profiler bietet dabei einen Überblick über die komplette Codeausführung, wie zum Beispiel die Anzahl von Iterationen einer Schleife oder eines Methodenaufwurfes.

Acknowledgment

Die Grundlagen zu „Mobile Hunters“ entstanden in einer Kooperation zwischen dem Fachgebiet Wirtschaftsinformatik I der TU Darmstadt und der T-Systems International GmbH. Im Rahmen des Wirtschaftsinformatikpraktikums im Wintersemester 2004/05 haben acht Studenten die Spielidee konzipiert und das Spiel „Mobile Hunters“ implementiert.

Ein besonderer Dank gilt den Studenten Tilman Laschinger, Florian Seeanner und Michael Wolf, die im Rahmen einer kooperativen Studienarbeit eine komplette Neuimplementierung und Funktionserweiterung erfolgreich entwickelt haben und damit die Basis für diesen Beitrag gelegt haben.

Literatur

- [LeLo05] Lonthoff, J.; Leiber, T.: Mobile Location Based Gaming als Wegbereiter für Location Based Services. In: von Knop, J.; Haverkamp, W. und Jessen, E. (Hrsg.): „Heute schon das Morgen sehen“, LNI-Proceedings Volume P-73 zur 19. DFN-Arbeitstagung über Kommunikationsnetze, Düsseldorf, 18.-20.05.2005, S. 343-360.
- [LoOr06] Lonthoff, J.; Ortner, E.: Mobile Location-Based Gaming as Enabler for Location-Based Services (LBS). In: Isaías, P.; McPherson, M.; Bannister, F. (Edts.): Proceedings of IADIS International Conference e-Society 2006, Volume I, 13.-16.07.2006, Dublin, Ireland, pp. 485-492.
- [NPM01] Nicklas, Daniela; Pfisterer, Christoph; Mitschang, Bernhard (2001): Towards Location-based Games. In: Proceedings of the International Conference on Applications and Development of Computer Games in the 21st Century: ADCOG 21, Hongkong Special Administrative Region, China, 2001, S. 61-67.
- [Ro05] Roth, J.: Mobile Computing: Grundlagen, Technik, Konzepte, 2. akt. Auflage, dpunkt.verlag, Heidelberg, 2005.
- [Rö02] Röttger-Gerigk, S.: Lokalisierungsmethoden. In: Gora, W.; Röttger-Gerigk, S. (Hrsg.): Handbuch Mobile-Commerce. Springer: Berlin, 2002, S.419-426.
- [Sc03] Schiller, J.: Mobilkommunikation, 2. Auflage, Pearson Studium, München, 2003.
- [Sh03] Shivas, M.: J2ME Game Optimization Secrets.
http://www.developer.com/ws/j2me/article.php/10945_2234631_1, Abgerufen am 01.08.06, 2003.
- [SV04] Schiller, Jochen H.; Voisard, Agnès: Location-based services. Morgan Kaufmann, San Francisco, CA, 2004.

Ein Framework für kontextbezogene Anwendungen in der Nexus-Plattform

Matthias Wieland

Daniela Nicklas

Institut für Architektur von Anwendungssystemen
Universität Stuttgart
Universitätsstraße 38, 70569 Stuttgart
wielanms@informatik.uni-stuttgart.de

Institut für Parallele und Verteilte Systeme
Universität Stuttgart
Universitätsstraße 38, 70569 Stuttgart
danickla@informatik.uni-stuttgart.de

Abstract: Im Rahmen des Nexus-Projekts wurden zahlreiche kontextbezogene Anwendungen entwickelt. Aus diesen Erfahrungen heraus wurde ein Framework erarbeitet, das die Entwicklung orts- und kontextbezogener Anwendungen durch geeignete Abstraktionen und Laufzeitbibliotheken unterstützt. Dieser Beitrag beschreibt den Aufbau und die Vorteile dieses Frameworks. Zum Schluss geht der Beitrag noch auf die zukünftige Entwicklung des Nexus-Projekts im Bereich Anwendungsunterstützung ein.

1 Einleitung

Die Entwicklung von kontextbezogenen Anwendungen ist sehr teuer, fehleranfällig und kompliziert, wenn keine Techniken eingesetzt werden, um die Wiederverwendung von Kontextdaten und Software-Komponenten zu ermöglichen. Die Nexus-Plattform [SFB627] ist eine Infrastruktur, welche eine Entwicklung solcher Anwendungen durch ein umfassendes Umgebungsmodell unterstützt [Ni05], durch welches Angebote verschiedener Datenanbieter dynamisch gefördert werden. Dadurch wird die Wiederverwendung der Umgebungsmodelldaten ermöglicht. Zudem werden von der Nexus-Plattform zahlreiche Dienste angeboten, wie z. B. der Ereignisdienst, Geonachrichtendienst oder Kartendienst.

Um die Funktion der Plattform zu demonstrieren, wurden zahlreiche Nexus-Anwendungen entwickelt [NGS03, NHM+04]. Diese Anwendungen wurden bisher immer von Grund auf neu entwickelt. Um diesen Entwicklungsprozess zu verbessern, wurde ein Framework für die Entwicklung kontextbezogener Anwendungen geschaffen [Wi05]. Dies soll zur Kostenreduzierung, Vereinfachung und Standardisierung bei der Nexus-Anwendungsentwicklung führen. Erreicht wurden diese Ziele durch die Anwendung der softwaretechnischen Grundkonzepte eines Frameworks, -Modularität, Wiederverwendbarkeit, Erweiterbarkeit und Umkehr des Kontrollflusses.

Es gibt diverse Frameworks, die zur Entwicklung kontextbezogener Anwendungen eingesetzt werden können [HeIn04, DeAb00, DSA01]. Allerdings sind diese nicht auf den Einsatz im Zusammenspiel mit der Nexus-Plattform ausgelegt.

Im Folgenden wird der Aufbau und die Grundkonzepte des entwickelten Frameworks beschrieben. Danach wird ein Ausblick auf die zukünftigen Vorhaben im Rahmen der Anwendungsunterstützung im Nexus-Projekt gegeben.

2 Aufbau des Frameworks

Das Framework besteht aus drei unterschiedlichen Teil-Frameworks [Wi05]. Diese drei Frameworks können auch getrennt voneinander eingesetzt werden. Das **Entwicklungs-Framework** besteht aus den Wrappern zum Zugriff auf die Nexus-Bibliotheken. Es wird über feste Schnittstellen vom Anwendungsprogrammierer genutzt und wird in den Sourcecode einer Nexus-Anwendung eingebunden. Alle Funktionen dieses Frameworks sind bereits zur

Compile-Zeit bekannt und fest verknüpft. Die Bibliotheken bieten allgemeine Funktionen zur Erhebung von Kontextinformationen und für die Abfrage von Kontextinformationen aus dem Umgebungsmodell.

Der zweite Baustein des Frameworks ist das **Laufzeit-Framework**. Es hat die Aufgabe, die Endbenutzer der Anwendungen zu unterstützen. Das Laufzeit-Framework ist eine eigenständige Anwendung, die auf dem Endgerät abläuft. Sie wird vom Benutzer des Endgerätes verwendet, um z. B. Nexus-Anwendungen zu finden und diese einfach zu installieren, zu konfigurieren und schließlich auszuführen. Es bietet jedoch auch Nexus-Anwendungen Unterstützung. Zum Beispiel kann das Laufzeit-Framework die Anwendungen über ihren aktuellen Ausführungskontext informieren. Außerdem verwaltet es beschränkte Ressourcen, wie z. B. Sensoren des Endgerätes, die sonst immer nur von einer Anwendung gleichzeitig genutzt werden könnten. Diese Funktionen können nur zur Laufzeit angeboten werden und sind darum nicht im Entwicklungs-Framework sondern im Laufzeit-Framework enthalten.

Das **Projekt-Framework** unterstützt den Entwicklungsprozess und somit den Entwickler mit Werkzeugen. Dieser Baustein wurde noch nicht implementiert. Einige Ideen dieses Framework-Teils werden in der zukünftigen Forschung näher untersucht. Eine kleine Auswahl an Themen wird in Kapitel 5 vorgestellt.

3 Aufgabenbereiche einer kontextbasierten Anwendung

In Abbildung 1 werden die konzeptionellen Schnittstellen einer kontextbasierten Nexus-Anwendung dargestellt. Die Anwendung kommuniziert über das Nexus-Interface mit der Nexus-Plattform, um Informationen über das Umgebungsmodell zu erhalten. Über die grafische Benutzerschnittstelle (User Interface) interagiert der Benutzer mit der Anwendung. Des Weiteren greift sie auf lokal am Endgerät angeschlossene Sensoren zu, um ihren Ausführungskontext zu ermitteln (Local System Interface). Außerdem nutzt sie entfernte Dienste, die Änderungen in der realen Umgebung bewirken (Remote Services Interface). Durch Framework-Unterstützung in diesen Hauptaufgabenbereichen kann die Nexus-Anwendungsentwicklung vereinfacht werden. Das entwickelte Framework bietet jedoch nur Unterstützung im Bereich des Nexus Interface und des Local System Interface. Die anderen Schnittstellen sind Gegenstand der Forschung und werden zukünftig an Bedeutung gewinnen.

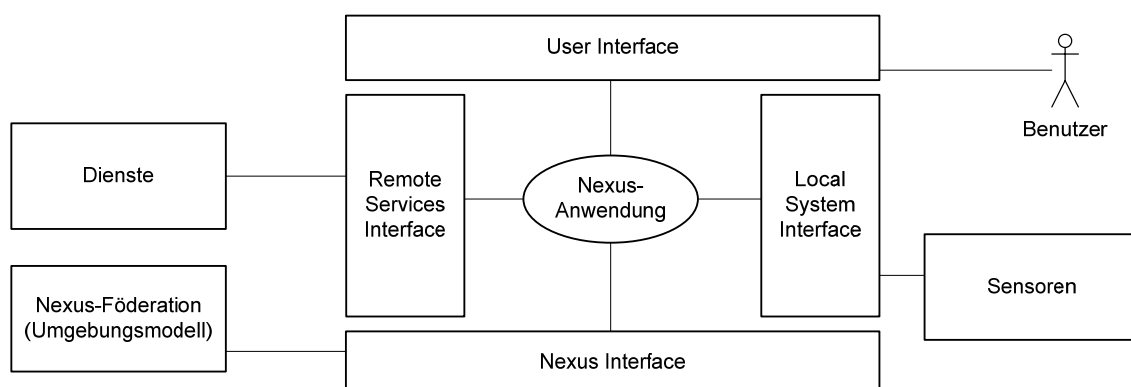


Abbildung 1: Die Schnittstellen einer Nexus-Anwendung

4 Konzepte des Frameworks

In diesem Kapitel werden die Konzepte vorgestellt, die dem Entwurf und der Implementierung des Frameworks zugrunde liegen.

Komponentenbasiertes System: Das Framework ist aus Komponenten, die festgelegte

Schnittstellen besitzen, aufgebaut. Diese Komponenten können flexibel miteinander verbunden werden. Das bedeutet aus Sicht der Anwendungen, dass die Komponenten entweder extern oder intern genutzt werden können. Werden die Komponenten extern genutzt, werden sie als eigenständige Prozesse gestartet und kommunizieren mittels Interprozesskommunikation. Intern genutzt bedeutet, sie werden fest in die Anwendungen eingebunden und sind somit Teil der Anwendung. Dadurch können sie direkt aufgerufen werden und laufen im Prozess der Anwendungen ab.

Robuste lokale Kontexterfassung: Die Erfassung der Kontextinformationen ist für Nexus-Anwendungen eine zentrale Aufgabe, darum soll sie zu einem möglichst großen Teil vom Framework übernommen werden. Das Kontexterfassungssystem des Frameworks ist: Robust gegen Ausfälle, unabhängig von der Sensor-Hardware, die genutzt wird, bietet Datenfusion bei mehrfach verfügbaren Sensordaten.

Ereignisbasiertes System: Um eine Umkehr des Kontrollflusses zu erreichen, wird an vielen Stellen im Framework ein ereignisbezogenes Benachrichtigungssystem eingesetzt. Dieses System besteht aus den folgenden Bestandteilen:

Verteiltes System: Um eine hohe Flexibilität des Frameworks zu erreichen, sollen dessen Komponenten als verteiltes System aufgebaut werden. Da die Subsysteme in eigenen Prozessen ablaufen, muss eine Technik für Interprozesskommunikation eingesetzt werden, um einen Informationsaustausch zwischen den Subsystemen zu ermöglichen. Für das Framework wurde Remote Method Invocation (RMI) gewählt.

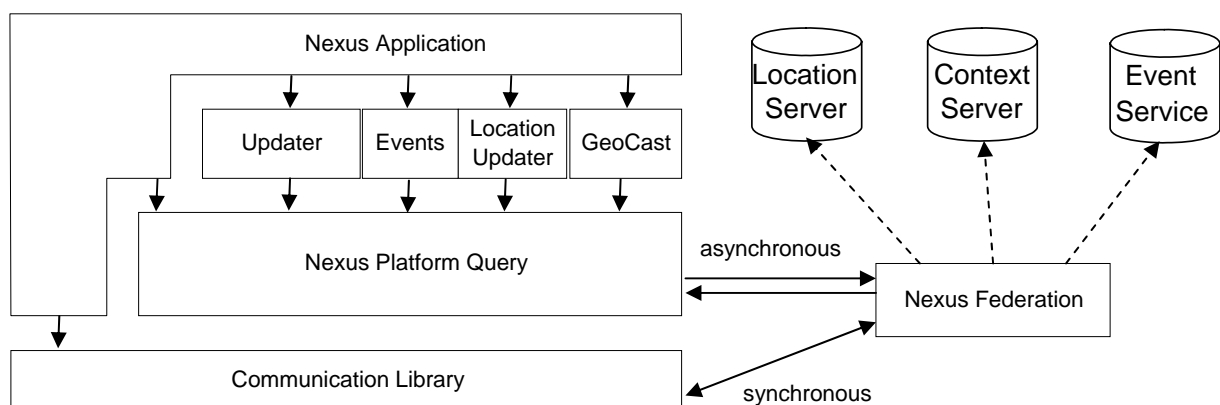


Abbildung 2: Die modulare Kommunikationsarchitektur des Frameworks

Modulare Kommunikationsarchitektur: Nexus-Anwendungen kommunizieren auf viele verschiedene Arten mit ihrer Umgebung. Damit sie immer auf dem gerade benötigten Abstraktionslevel Unterstützung finden, wird von dem Framework eine modulare Kommunikationsarchitektur, die aus drei Schichten besteht, bereitgestellt (siehe Abbildung 2). Die Anwendungen können je nach Wunsch mittels jeder dieser Schichten kommunizieren. Die Anwendungen werden dadurch in die Lage versetzt, mit der Nexus-Plattform und externen Diensten zu kommunizieren. Die unterste Schicht kapselt die zur Kommunikation eingesetzte Methode (z. B. RMI oder SOAP). Diese Schicht arbeitet synchron, d. h., die Anwendung wird bei der Nutzung blockiert, bis die Antwort des Servers verfügbar ist oder ein Timeout stattfindet. Die mittlere Schicht realisiert asynchrone, ereignisbasierte Kommunikationsfunktionen. Für jeden Nexus-Dienst existieren Stellvertreterobjekte, die mit diesen Kommunikationsfunktionen arbeiten. Schließlich stellt die oberste Schicht erweiterte Funktionen zur Verfügung, die auf den Basisfunktionen aufsetzen. Dies sind Funktionen, die im Anwendungsbereich oft benötigt werden und dafür optimierte Schnittstellen anbieten. Eine Funktion "Location Updater" übernimmt z. B. das Aktualisieren der eigenen Position im Location Server, ohne dass dafür extra eine Kontextanfrage erzeugt werden muss.

5 Zusammenfassung und Ausblick

Die Funktionalität des Frameworks wurde durch Neuentwicklung zweier ortsbezogener Beispielanwendungen überprüft. Dies hat gezeigt, dass die Anwendungsentwicklung durch das Framework erleichtert wird, und die Kosten reduziert werden. Bei bestehenden Anwendungen wurde nachgewiesen, dass durch den Einsatz des Frameworks die Qualität der Software verbessert werden kann.

Insgesamt zeigte sich durch die Erfahrungen im Nexus-Projekt, dass für die Entwicklung mobiler, orts- oder kontextbezogener Anwendungen neue oder angepasste Methoden, Werkzeuge und Entwurfsmuster notwendig sind, die durch den Stand der Technik nur unzulänglich unterstützt werden [BDK+06]. Deswegen wird dieses Forschungsgebiet im Projekt Nexus in Zukunft verstärkt aus verschiedenen Blickwinkeln bearbeitet: Für große, prozessorientierte Anwendungen, wie sie im Umfeld der Smart Factory [BJR+03] auftreten, wird erforscht, wie der Kontrollfluss der Anwendungen mit Hilfe von kontextbasierten Workflows dargestellt werden kann. Dies wird sich vor allem für Anwendungen in prozessorientierten Umgebungen auszahlen [WLJ+06]. Des Weiteren wird für die Entwicklung verteilter kontextbasierter Anwendungen Werkzeugunterstützung entwickelt, ebenso für das Testen und Debuggen, da sich dies wesentlich vom Testen normaler Anwendungen unterscheidet. Im Bereich der Kontextverarbeitung wird eine weitere Abstraktionsebene eingeführt. Aus Basiskontext wird höherwertiger Kontext, so genannte Situationen, abgeleitet. Diese Situationen stehen allen Anwendungen zur Verfügung, werden aber von der Plattform erkannt und überwacht.

Literaturverzeichnis

- [NGS03] D. Nicklas; M. Grossmann; T. Schwarz: NexusScout: An Advanced Location-Based Application On A Distributed, Open Mediation Platform, Proceedings of the 29th VLDB Conference, Berlin, Germany, 2003
- [NHM+04] D. Nicklas; N. Hönle; M. Moltenbrey; B. Mitschang: Design and Implementation Issues for Explorative Location-based Applications: the NexusRallye, Proceedings for the VI Brazilian Symposium on GeoInformatics: GeoInfo 2004
- [HeIn04] K. Henriksen; J. Indulska: A Software Engineering Framework for Context-Aware Pervasive Computing, Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04), Orlando, Florida, March 14 - 17, 2004, pp. 77-86
- [DeAb00] A. Dey; G. Abowd: The Context Toolkit: Aiding the Development of Context-Aware Applications, Workshop on Software Engineering for Wearable and Pervasive Computing, Limerick, Ireland, June 6, 2000
- [DSA01] A. Dey; D. Salber; G. Abowd: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, Human-Computer Interaction, 16. 2001
- [SFB627] Sonderforschungsbereich 627: Nexus, <http://www.nexus.uni-stuttgart.de/>
- [Wi05] M. Wieland: Entwicklung eines Frameworks für Nexus-Anwendungen (Diplomarbeit), 2005
- [BJR+03] M. Bauer; L. Jendoubi; K. Rothermel; E. Westkämper: Grundlagen ubiquitärer Systeme und deren Anwendung in der "Smart Factory". In: Gronau, Norbert (ed.); Krallmann, Hermann; Scholz-Reiter, Bernd (ed.): Industrie Management - Zeitschrift für industrielle Geschäftsprozesse. Bd. 19(6), 2003
- [WLJ+06] M. Wieland; F. Leymann; L. Jendoubi; D. Nicklas; F. Dürr: „Task-orientierte Anwendungen in einer Smart Factory“, 1. Konferenz Mobilität und Mobile Informationssysteme (MMS 2006), Lecture Notes in Informatics (LNI), P-76, 2006
- [Ni05] D. Nicklas: „Ein umfassendes Umgebungsmodell als Integrationsstrategie für ortsbezogene Daten und Dienste“, Dissertation, Universität Stuttgart, 2005
- [BDK+06] C. Becker; F. Dürr; M. Knoll; D. Nicklas; T. Weis: „Entwicklung ortsbezogener Anwendungen.“ Praxis der Informationsverarbeitung und Kommunikation. (angenommen), 2006

3. Fachgespräch »Ortsbezogene Anwendungen und Dienste« **A Generic Approach to Modeling Context Constraints¹**

Rüdiger Gartmann
Fraunhofer ISST, Dortmund
9. August 2006

1 Introduction

A couple of years ago, context awareness were basically limited to location sensitivity. Today, context is interpreted in a much broader sense. Dey's definition [Dey00] is widely accepted. In the last years much work has been done on context modeling. An overview is given by Strang and Linnhoff-Popien [StLi04]. Common to these approaches is that context is modeled as a distinct state in order to describe the situation of a certain entity at a certain point of time. To make use of this context information it has to be matched against a set of criteria defining how to react on the given context values. These criteria can be regarded as context constraints. Whenever an actual context fulfills these constraints the according action has to be executed. Still missing is a model to describe these criteria, which is subject of this paper.

2 Context Scope Model

Context can be modeled as an n-tuple of context dimensions such as location, time, air temperature, business hours, etc. Mapped to a Cartesian coordinate system, each context can be regarded as a point in an n-dimensional hyperspace, spanned by the n context dimensions. In contrast to this, context constraints relate to one or more geometric fields in that hyperspace, called solids. They define the scope of valid contexts. Thus, context constraints are fulfilled whenever the point in the context hyperspace representing an actual context is element of a solid representing the context scope.

To cope with the different scale units of context dimensions, all context values have to be mapped onto a Cartesian coordinate system with all values being elements of the real numbers. The transformation can be expressed as an injective mapping of the initial context values to the context representation.

¹ This work is partially funded by the German Ministry of Education and Research within the COMPASS 2008 project (Grant No. 01 IMD02B)

A typical approach to define multi-dimensional geometries (here context scopes) is to express them as systems of linear inequations. A boundary representation as usually used in Solid Modeling [Mänt88] would be also applicable but is disregarded due to the fast growing complexity for larger numbers of dimensions. Conceptually, the approach is limited to linear geometries, i.e. polytopes, but in practical applications it seems reasonable to cope with that limitation because it simplifies the definition of scopes as well as the computation of the scope-context-matching. As in other domains such as the geospatial domain, non-linear objects can be approximated by linear ones. Thus, the required preciseness of such a model is achieved by the according granularity of the approximation.

To describe a context scope as a system of linear inequations, these inequations can be defined either using the context representation model or using the initial context dimensions. In the latter case, the scope inequations have to be transformed into the same representation as discussed above for the representation of context dimensions. Transforming inequations which are only defined on a single context dimension transformation is fairly easy. In this case the transformation function for that context dimension can be applied to the given context restrictions (for b being a constant, c_i the initial context value, d_i the representation value and i the context dimension):

$$c_i \leq b \rightarrow d_i \leq f(b)$$

Inequations (or context restrictions) being defined on more than one context dimension are more complex. First of all, such restrictions are only feasible if both context dimensions share a common range of values and can semantically be related to each other. This is probably not often the case. But it is conceivable that for instance a restriction is applied to a quotient of a person's weight and height. Although this is a trivial example, (weight and height are expressed in real numbers and therefore the transformation probably is $d_i=f(c_i)=c_i$) we can state in general that for each context element in the term the transformation function has to be applied. Thus, any context restriction defined on the initial context scales of the form

$$a_1c_1 + a_2c_2 + \dots + a_nc_n \leq b$$

(defining a straight line with coefficients a_i and a constant b) can be transformed into a context restriction in the context representation model as follows (with f being the transformation function):

$$a_1f^{-1}(c_1) + a_2f^{-1}(c_2) + \dots + a_nf^{-1}(c_n) \leq b$$

Typically, systems of linear inequations are represented in a matrix notation which allows to apply standard calculation methods such as the simplex algorithm. Thus, the inequations have to be transformed into a standard notation in a form like $a_1x_1+a_2x_2+\dots+a_nx_n+b\leq 0$. Using a matrix notation for m different inequations of R_n leads to the following matrix:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{pmatrix}$$

A non-convex service scope would be represented by more than one matrix. Within a matrix, all inequations are 'AND'-associated, different matrices are 'OR'-associated.

3 Example

Figure 1 shows a context scope defined on the context dimensions latitude, longitude and time. The point C is an actual user context.

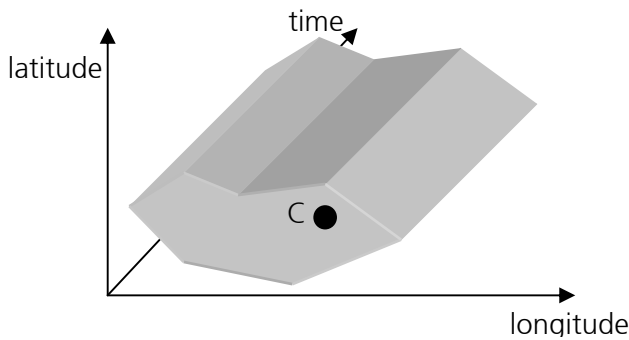


Figure 1: Context Scope

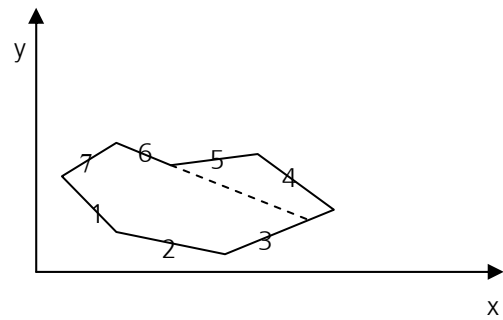


Figure 2: Geographic Scope

As visualized in Figure 2, the geographic scope is represented by a non-convex polygon, bounded by seven straight lines. This non-convex polygon now has to be split into two convex polygons; the original polygon can for instance be split along the straight line 6. The result is two polygons, one bounded by the straight lines 1, 2, 3, 6, 7, and the second by the straight lines 3, 4, 5, 6. Represented as systems of linear inequations, they are described as follows:

Polygon 1:

$$\begin{aligned} y &\geq -x + 1,4 && \Leftrightarrow -x - y + 1,4 \leq 0 \\ y &\geq -0,125x + 0,6 && \Leftrightarrow -0,125x - y + 0,6 \leq 0 \\ y &\geq 0,4x - 0,9 && \Leftrightarrow 0,4x - y - 0,9 \leq 0 \\ y &\leq -0,4x + 2 && \Leftrightarrow 0,4x + y - 2 \leq 0 \\ y &\leq 0,59x + 1 && \Leftrightarrow -0,59x + y - 1 \leq 0 \end{aligned}$$

Polygon 2:

$$\begin{aligned} y &\geq 0,4x - 0,9 && \Leftrightarrow 0,4x - y - 0,9 \leq 0 \\ y &\leq -0,75x + 3,6 && \Leftrightarrow 0,75x + y - 3,6 \leq 0 \\ y &\leq 0,16x + 1 && \Leftrightarrow -0,16x + y - 1 \leq 0 \\ y &\geq -0,4x + 2 && \Leftrightarrow -0,4x - y + 2 \leq 0 \end{aligned}$$

If we assume a restriction on the time dimension z such as 'between 8 and 21 hours' then using the transformation function $z = f(\text{time}) = \text{hours}(\text{time}) * 60 + \text{minutes}(\text{time})$ this would lead to the following additional inequations:

$$z \geq 480 \quad \Leftrightarrow -z + 480 \leq 0; \quad z \leq 1260 \quad \Leftrightarrow z - 1260 \leq 0$$

Using the matrix notation, this would lead to the following 'OR'-associated two matrices:

$$\begin{pmatrix} -1 & -1 & 0 & 1,4 \\ -0,125 & -1 & 0 & 0,6 \\ 0,4 & -1 & 0 & -0,9 \\ 0,4 & 1 & 0 & -2 \\ -0,59 & 1 & 0 & -1 \\ 0 & 0 & -1 & 480 \\ 0 & 0 & 1 & -1260 \end{pmatrix} \vee \begin{pmatrix} 0,4 & -1 & 0 & -0,9 \\ 0,75 & 1 & 0 & -3,6 \\ -0,16 & 1 & 0 & -1 \\ -0,4 & -1 & 0 & 2 \\ 0 & 0 & -1 & 480 \\ 0 & 0 & 1 & -1260 \end{pmatrix}$$

Assumed the context C is represented by the coordinates (2,9; 1,2; 555), this would be a solution of the second matrix, which means context C is an element of the context scope.

4 Summary

Since different applications in different domains may use completely different context dimensions it is necessary to define a model of maximum genericity. This is achieved by mapping the context dimensions to dimensions in a Cartesian coordinate system. Thus, the specifics of each context dimension is handled by the transformation functions, resulting in a completely generic representation. Using the proposed model, contexts are represented as points in the context hyperspace, whereas the context scope is represented by one or more polytopes. A certain context lies within a scope if the context coordinate lies within (one of) the scope polytope(s), or, in other words, of the context tuple is element of the solution set of a matrix representing that polytope.

5 References

- [Dey00] Anind K. Dey: Providing architectural support for building context-aware applications. PhD Thesis, College of Computing, Georgia Institute of Technology, 2000.
- [Mänt88] Martti Mäntylä: An Introduction to Solid Modeling. Computer Science Press, Rockville, Maryland, 1988.
- [KoGr04] Thomas H. Kolbe, Gerhard Gröger: Unified Representation of 3D City Models. In: Geoinformation Science Journal, Vol. 4, No. 1, 2004.
- [StLi04] Thomas Strang, Claudia Linnhoff-Popien: A Context Modeling Survey. In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004, Nottingham/England, 2004.

Impact of Misbehaviour and its Detection in Ad-hoc Wireless Sensor Networks using AIS

Sven Schaust, Martin Drozda, Helena Szczerbicka
FG Simulation und Modellierung
Institute of Systems Engineering
University of Hannover
{svs,drozda,hsz}@sim.uni-hannover.de

1 Motivation

Characteristic for ad-hoc wireless sensor networks is the lack of a centralized control. Instead each node serves as a routing device, which is able to forward packets to its neighbors¹ and receive packets from them. Node movement is allowed, but rare for sensor nodes. Each node is able to be switched on and off at any time depending on its power saving configuration. Such networks are extremely vulnerable to user misbehavior. Since nodes within an ad-hoc sensor network are expected to have limited computational power and be battery powered, a system that is going to protect them has to be *lightweight*. It has to be adaptive too, as ad-hoc sensor networks are expected to operate autonomously with only spare maintenance. Classical intrusion detection approaches which are based on intrusion signatures therefore do not apply for such a task.

Artificial immune systems (AIS) on the other hand seem capable of handling such demands. AIS are inspired by the *Human immune system* (HIS) using selected features of this defense system. The basic feature of an HIS is the ability to discriminate between *self* and *non-self*. In case of sensor networks non-self is any kind of user behavior that impacts the network in a negative way. The purpose of our simulation based experiments is to show that AIS are a suitable approach for detecting misbehavior in ad hoc sensor networks.

2 Artificial Immune System

Artificial immune systems (see [2], [9]) are inspired by the Human immune system. The latter one is a complex mechanism which is able to protect humans against a variety of virological and bacteriological attacks. The system's capabilities to differentiate between *self* and *non-self* is remarkable.

¹Nodes within radio range of the sender.

Due to evolutionary development the human body has gained an innate immune system and a system which is able to adapt and learn to detect and neutralize new kinds of attacks.

2.1 Adaptive Immune System

The adaptive immune system is the source of inspiration when it comes to artificial immune systems. The cells involved in this system are called T- and B-cells. The first ones are produced in the thymus, while the latter ones are produced in the bone marrow. These cells are able to bind to specific structures (or surfaces) of attackers (*antigens*), thus building up an effective defense line. Whenever a T-cell binds to an antigen a response is triggered which involves B-cells to eliminate the antigen. Due to a sophisticated learning process the immune system is able to memorize attackers and to respond faster the next time.

2.1.1 Learning Phase

T-cells are covered by receptors that are able to bind antigens. These cells are produced by a random process in the thymus. After surviving a negative selection process² the T-cells are injected into the body. A reaction is forced when T-cells bind with a possible non-self antigen. If the non-self is classified as harmful an immune reaction is triggered by T- and B-cells. The T-cells start to divide and mature over time, also producing memory T-cells. These are highly specialized long living cells and if activated by antigens cause a steady and fast immune reaction. Mature B-cells are able to produce antibodies. These match only specific antigens of known attackers. Vaccination is an artificial process to stimulate such a response and production. See [4] for more details on the human immune system.

²Cells must not recognize self.

2.2 AIS Algorithm

An Artificial immune system normally reflects typical behavior of the Human immune system by collapsing different sorts of T- and B-cells into a single entity called detector.

AIS Algorithm:

1. Create self-set
2. Create detector with pseudo-random process
3. Check detector against self-set. If it does match, throw detector away. If not add detector to detector-set.
4. Compare suspected non-self strings with detectors. If it matches trigger response. If the response is triggered quite often, mark detector as memory (T-cell) detector.
5. Clone and randomly change the detectors binding capabilities to produce new detectors.

3 Misbehavior detection with AIS

In accordance with the literature, see e.g. [3], we represent self, non-self and detectors as bit-strings. The matching rule employed is the *r-contiguous bits matching rule*. Two bit-strings of equal length match under this rule if there exists a substring of length r at position p in each of them and these substrings are identical. Candidate detectors are produced using negative selection, i.e. they are created randomly and tested against the set of self strings. If they do not match any self string they become detectors. Similar to [2] we have collapsed different sorts of T- and B- cells (different types of detectors) into a single entity.

3.1 Experimental Setup

The goal of our experiments was to show what kind of impact misbehavior has on ad-hoc sensor networks and to what extent AIS are feasible for detection of such misbehavior. Therefore we used an ad-hoc scenario to simulate traffic within such a network.

Node distribution: Random way point snapshot of 1718 nodes in a square of $3km^2$.

Number of Connections: 10 connections with a constant bit rate of 1 packet per second.

MAC protocol: IEEE 802.11b.

Routing protocol: DSR.

Simulated time: 4 hours.

Hardware: Linux (SuSE 10.0) PENTIUM 4, 3GHz PC with 2 GB RAM.

Misbehavior: Dropping of packets which should be forwarded.

We produced two traffic trace files, one with normal behavior and one with misbehavior. Misbehavior was produced synthetically at 236 nodes. These nodes were programmed to drop 10% of all packets to be forwarded. From each trace file we produced 28 time windows each covering 500 seconds. We created antigens defined by genes for every node in a time window. Genes were created from the MAC and routing layer. We believe that combining several layers of the OSI model is necessary in order to generate invariant genes. We observed several values for every single time window: number of complete handshakes (RTS,CTS,DATA,ACK), number of RTS packets sent, number of data to be forwarded, number of data packets actually forwarded, average delay for each forward, number of RERR packets to be forwarded, number of RERR packets actually forwarded and its average delay. The last three values were taken from the routing layer, while the rest was taken from the MAC layer. The produced genes are the following:

$$\text{Gene 1: } \frac{\text{completeHandshakes}}{\text{numberOfRTSSent}} \times 100$$

$$\text{Gene 2: } \frac{\text{numberForwardedData}}{\text{numberDataToBeForwarded}} \times 100$$

Gene 3: Average delay between forwarding a data packet and forwarding at the next hop.

$$\text{Gene 4: } \frac{\text{numberRERRForwarded}}{\text{numberRERRToBeForwarded}} \times 100$$

Gene 5: Average delay between forwarding a RERR packet and forwarding at the next hop.

Each gene is a bit-string of length 10, where each bit represents an interval. Antigens were produced by concatenating the 5 genes. The undefined value of a gene was set to the maximum interval in case of genes 1, 2, 4 and to the minimum interval for genes 3 and 5.

We used an AIS with a *negative selection algorithm* and the *r-contiguous bits matching rule* for string matching (see also [3]) to detect misbehaviour for each node in all time windows.

4 Results

The experiments show that an AIS is able to detect misbehavior in an ad-hoc network using MAC and routing related genes as antigens. However the shifting of traffic³ from a misbehaving area to an area where no traffic was present before, induced a mis-prediction anomaly. It is likely that this kind of mis-prediction would also occur on real nodes which encounter traffic for the first time. We produced two plots (see figure 1) showing the origin of misbehavior in the Ad-hoc scenario and the detection results by our AIS.

Picture at the top:

To produce a better to read plot only nodes with traffic ≥ 1000 packets are shown. Data was interpolated using the *pm3d* function of gnuplot. The plot shows the misbehavior origin, i.e. the region of nodes which actually drop 10% of packets which should be forwarded.

Picture at the bottom:

The second picture shows the AIS detection results. Again the plots data was interpolated using the *pm3d* function. All nodes which were detected by the AIS are used in the plot. The mentioned anomaly is the gray region on the left.

It seems that relying on the MAC and routing layer alone does not prevent mis-prediction at the moment. More genes combining the information of several OSI layers may be necessary.

We plan to extend our AIS in order to handle such anomalies. Furthermore we plan to test the effects of higher packet loss rates and other misbehavior scenarios (see [8], [10], [7]) with different numbers of nodes affected. We have therefore produced traffic traces for scenarios with 20%, 30% and 50% misbehavior.

In order to verify the simulation runs an implementation of a distributed Artificial Immune System on a sensor network⁴ is mandatory.

References

- [1] Zhou Ji, Dipankar Dasgupta. Real-Valued Negative Selection Algorithm with Variable-Sized Detectors. *Proc. Genetic and Evolutionary Computation Conference 2004 (GECCO-2004)*, 2004: 287-298.
- [2] S. Hofmeyr and S. Forrest. Immunity by Design: An Artificial Immune System. *Proc. Genetic and Evolutionary Computation Conference (GECCO)*, Morgan-Kaufmann, San Francisco, CA, pp. 1289-1296 (1999).
- [3] Slaviša Sarafijanović and Jean-Yves Le Boudec. An Artificial Immune System for Misbehavior Detection in Mobile Ad-Hoc Networks with Virtual Thymus, Clustering, Danger Signal and Memory Detectors. *Proc. ICARIS (Third international conference on artificial immune systems)*, 2004.
- [4] Charles A. Janeway Jr. How the immune system works to protect the host from infection: a personal view. *Proc. Natl. Acad. Sci. U S A.*, 2001 Jun 19;98(13):7461-8.
- [5] U. Aickelin, P. Bentley, S. Cayzer, J. Kim, and J. McLeod. Danger theory: The link between ais and ids. *Proc. International Conference on Artificial Immune Systems (ICARIS'03)*, pages 156–167, Edinburgh, UK, 2003.
- [6] Uwe Aickelin, Julie Greensmith and Jamie Twycross. Immune System Approaches to Intrusion Detection - A Review. *Proc. the 3rd International Conference on Artificial Immune Systems (ICARIS 2004)*, Catania, Italy, 2004
- [7] C. L. Barrett, M. Drozda, D. C. Engelhart, V. S. Anil Kumar, M. V. Marathe, M. M. Morin, S. S. Ravi, and J. P. Smith. Understanding Protocol Performance and Robustness of Ad Hoc Networks Through Structural Analysis. *Proc. IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2005)*.
- [8] Alvaro A. Cardenas, Svetlana Radosavac, John S. Baras. Detection and prevention of MAC layer misbehavior in ad hoc networks. *Proc. 2nd ACM workshop on Security of ad hoc and sensor networks*, 2004.
- [9] D'haeseleer, P., Forrest, S., and Helman, P. An immunological approach to change detection: Algorithms, analysis and implications. *Proc. IEEE Symposium on Research in Security and Privacy*, 1996.
- [10] Yih-Chun Hu, Adrian Perrig, David B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. *Proc. ACM workshop on Wireless security*, 2003.

³DSR tries to find new routes between source and destination if a route becomes invalid.

⁴We will use Mica2 motes running *TinyOS*.

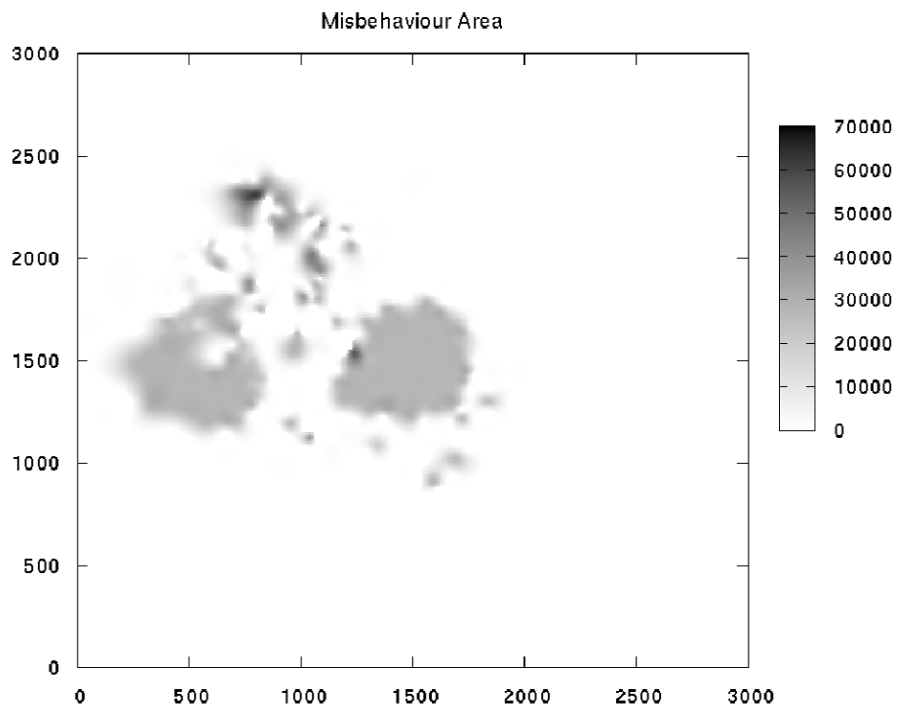
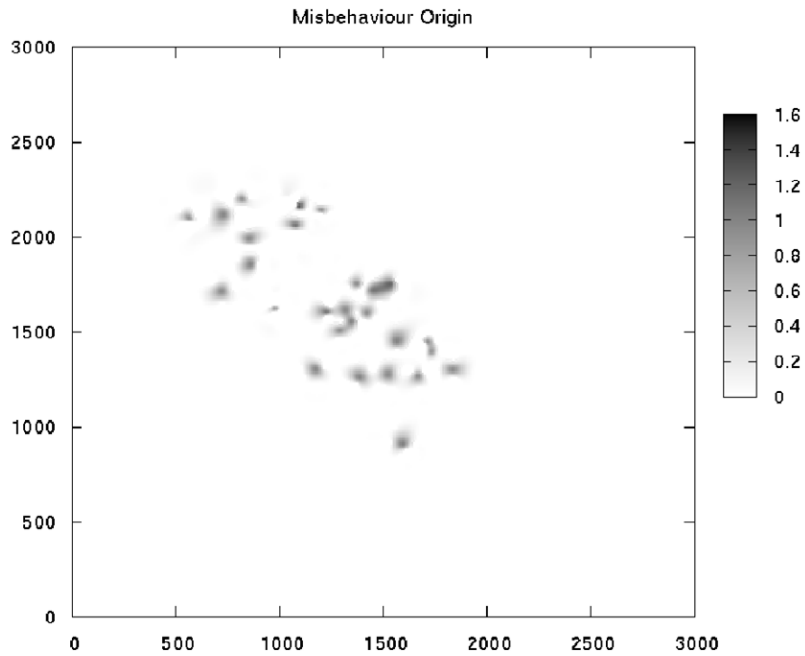


Figure 1: Misbehavior origin and the detection results from the AIS. Top picture only shows misbehaving nodes which have traffic ≥ 1000 packets. Bottom picture shows the detection results of the AIS. Both plots use interpolated data.

Positionserkennung von Studierenden in Hörsälen mit dem Chi-Quadrat-Anpassungstest

Thomas King*, Stephan Kopf*, Wolfgang Effelsberg*

Zusammenfassung

Im Folgenden wird ein Positionierungsalgorithmus vorgestellt, der die Position von Studierenden innerhalb eines Hörsaals ermittelt. An der Universität Mannheim werden Vorlesungen schon seit mehreren Jahren durch interaktive Dienste unterstützt, welche die Studenten mittels PDAs oder Laptops über Wireless-LAN nutzen. Durch die Analyse der Signalstärken der umgebenden Access Points und einen Vergleich von bekannten Messwerten mittels des Chi-Quadrat-Anpassungstests ist eine genaue Bestimmung der Position des mobilen Gerätes und somit auch des einzelnen Studierenden möglich. Dies erlaubt in einem weiteren Schritt die automatische Aufzeichnung von interaktiven Vorlesungen, bei denen die Fragen von Studierenden durch Hineinzoomen mit der Hörsaalkamera besser aufgezeichnet werden kann.

1 Einleitung

Die Übertragung einer Vorlesung an eine andere Universität ist heute ohne größeren technischen Aufwand möglich, führt jedoch zu deutlichen Akzeptanzproblemen bei den Studierenden. Motivationsprobleme treten auf, da der Dozent nur schwer auf Fragen oder Kommentare der entfernten Hörer eingehen kann. An unserer Universität haben wir im Rahmen des WIL/MA-Projektes (Wireless Interactive Learning in Mannheim) [4] ein Softwaresystem entwickelt, das interaktive Dienste speziell für Vorlesungen zur Verfügung stellt. Studierende nutzen diese Dienste seit mehreren Jahren auf mobilen, per Wireless-LAN angebundenen Geräten (z.B. PDAs oder Laptops) und erhalten so die Möglichkeit der bidirektionalen synchronen Kommunikation mit dem Dozenten.

Zur Kommunikation übermittelt ein Studierender im entfernten Vorlesungssaal seinen Fragewunsch über sein Endgerät und stellt dann seine Frage nach

*{king,kopf,effelsberg}@informatik.uni-mannheim.de, Lehrstuhl für Praktische Informatik IV, Universität Mannheim

Aufforderung durch den Dozenten per Mikrofon. Obwohl das Video des entfernten Hörsaal übertragen wird, können einzelne Studenten wegen der geringen Größe häufig nicht erkannt werden. Zur Verbesserung der Kommunikation bietet es sich an, die Kamera im entfernten Hörsaal selbstständig auf den fragenden Studenten auszurichten. Das zentrale Problem besteht nun darin, die Position des Studierenden automatisch zu bestimmen.

2 Verwandte Projekte

In den vergangenen Jahren sind neben dem weltweit verfügbaren Global Positioning System (GPS) eine Vielzahl von Positionierungssystemen für den Einsatz innerhalb von Gebäuden entwickelt worden. Dies ist im Wesentlichen auf die Beschränkungen von GPS zurückzuführen, das innerhalb von Gebäuden nicht zuverlässig funktioniert, da die Radiosignale nicht stark genug sind, um Decken und Wände zu durchdringen. Als Positionierungssysteme für den Einsatz innerhalb von Gebäuden haben sich Wireless-LAN-basierte Verfahren hervorgetan, da diese auf die oftmals vorhandene Hardware-Infrastruktur zurückgreifen und deshalb kostengünstig aufgebaut werden können [1] [2]. Zudem bieten diese Verfahren auch innerhalb von Gebäuden eine durchschnittliche Positionierungsgenauigkeit von bis zu 1,5 Metern [3].

Für eine zuverlässige Positionsbestimmung sind insbesondere Fingerabdruck-basierte Verfahren geeignet. Bei diesen zweistufigen Verfahren wird im ersten Schritt – der sogenannte *Trainingsphase* – die Signalstärken aller empfangenen Access Point an zuvor definierten Messpunkten gesammelt und in einer Datenbank mit den tatsächlichen Positionen (physikalischen Koordinaten) gespeichert. Die Messpunkte sollten dabei den Hörsaal vollständig abdecken. Die elektromagnetischen Eigenschaften an jedem Messpunkt werden als Radio-Fingerabdruck bezeichnet, da sie diesen Messpunkt meist eindeutig identifizieren. In der *Positionierungsphase*, also während der Vorlesung, führt jedes einzelne mobile Gerät der Studierenden regelmäßig Messungen durch und berechnet seine Position. Diese werden dann verwendet, um die Kamera des Hörsals entsprechend auszurichten.

3 Positionierungsalgorithmus

Der in diesem Abschnitt vorgestellte Positionierungsalgorithmus basiert auf dem Chi-Quadrat-Anpassungstest. Der Chi-Quadrat-Anpassungstest überprüft die Hypothese, ob eine Zufallsvariable einer gegebenen Wahrscheinlichkeitsverteilung entspricht. In unserem Fall fassen wir die n Messungen während der Positionierungsphase als Zufallsvariable X auf. Für jeden Messpunkt j (mögliche Sitzposition eines Studenten im Hörsaal) aus der Trainingsphase bestimmen wir

die entsprechende Verteilung F_j . Der Chi-Quadrat-Wert χ_j^2 ist definiert als:

$$\chi_j^2 = \sum_{k=0}^K \begin{cases} \frac{(n_k - \tilde{n}_{kj})^2}{\tilde{n}_{kj}} & \text{wenn } \tilde{n}_{kj} \geq 1, \\ \tau & \text{sonst.} \end{cases},$$

wobei

$$\tilde{n}_{kj} = n * F_j(k)$$

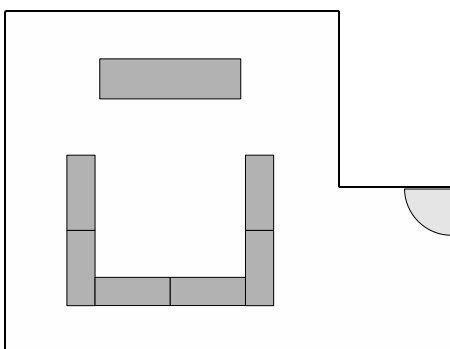
n_k beschreibt die absolute beobachtete Häufigkeit des Merkmals k . Die Position des Messpunktes mit dem kleinsten Chi-Quadrat-Wert wird als geschätzte Position des Studierenden angenommen.

Die Zählvariable k läuft in der oben präsentierten Formel von 0 bis $K = -102$, da dies die zulässigen Werte sind, die bei einer Wireless-LAN-Messung vorkommen können. Sollte \tilde{n}_{kj} einen Wert kleiner 1 annehmen, so geht ein Fehlerwert τ in die Berechnung von χ_j^2 ein. Bei unseren Messungen hat sich ein Wert von $10 * n_k$ für τ als zweckmässig erwiesen.

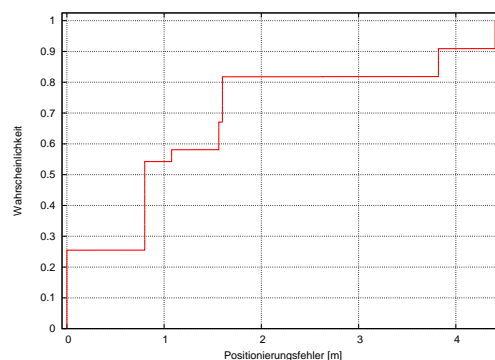
4 Experimentelle Messungen

Dieser Abschnitt beschreibt zunächst die Testumgebung, in der die experimentellen Messungen durchgeführt wurden. Anschliessend wird die Art und Weise der Datensammlung beschrieben. Darauf folgend wird der Positionierungsfehler des Positionierungssystems untersucht und die Ergebnisse diskutiert.

4.1 Testumgebung



(a) Skizze des Seminarraumes ($9,50 \times 8,25$ Meter) mit dem Rednertisch und Tischen für die Studierenden



(b) Kumulierte Verteilungsfunktion des Positionierungsfehlers

Abbildung 1: Skizze des Seminarraumes und Messung des Positionierungsfehlers

Als Testumgebung haben wir den Seminarraum *C112* im *A5, 6* Gebäude der Universität Mannheim ausgewählt. Dieser Seminarraum bietet 12 Studenten Platz und ist ungefähr 9,50 Meter lang und ungefähr 8,25 Meter breit. Die Anordnung der Tische mit jeweils zwei Sitzplätzen für Studierende und der Rednertisch sind in Abbildung 1(a) skizziert.

Der Seminarraum wurde mit vier Cisco Linksys WRT54G Access Points ausgestattet, die aber nur zur Positionierung genutzt wurden. Zur Datenermittlung wurde ein Laptop des Herstellers IBM (Thinkpad R51), eine Lucent Orinco Silver PCMCIA Netzwerkkarte und als Betriebssystem Linux mit Kernelversion 2.6.16 und den Wireless Tools 28per13 verwendet.

4.2 Datensammlung

An jedem der 12 Sitzplätze der Studierenden wurden sowohl für die Trainingsphase als auch für die Positionierungsphase 110 Messungen durchgeführt. Dabei wurde der Laptop auf den Tisch vor den Sitzplatz gestellt, und zwar so, dass der Laptop vom sitzenden Bediener bequem genutzt werden konnte. Nach Abschluss der Trainingsphase wurden die Daten für die eigentliche Positionierung gesammelt, wobei sich die Positionen und Ausrichtungen des Laptops im Vergleich zur Trainingsphase unterschieden.

4.3 Positionierungsfehler

Aus den Trainings- und Positionierungsdatensätzen wurden zufällig 50 Messungen ausgewählt, und anhand dieser Messungen wurde unter Zuhilfenahme des Chi-Quadrat-Positionierungsalgorithmus der Positionierungsfehler bestimmt. Um statistisch stabile Ergebnisse zu erzielen, wurde dieses Experiment 1000 mal wiederholt.

Die kumulierte Verteilungsfunktion des Positionierungsfehlers ist in Abbildung 1(b) dargestellt. Wie man aus der Abbildung entnehmen kann, wurde die Position zu etwa 25 Prozent richtig erkannt. Mit einer Wahrscheinlichkeit von ungefähr 60 Prozent weicht die ermittelte Position um maximal einen Sitzplatz von der realen Position ab. In etwa 20 Prozent der Fälle treten Positionierungsfehler grösser 1,5 Meter auf, d.h. die ermittelte Position ist nicht mehr in direkter Nachbarschaft zur tatsächlichen Position.

Der hier vorgestellte Positionierungsalgorithmus erreicht eine durchschnittliche Positionierungsgenauigkeit von etwa 1,39 Metern bei einer Standardabweichung von 1,40 Metern.

4.4 Diskussion

Vergleicht man die Positionierungsperformance des hier vorgestellten Algorithmus beispielsweise mit dem aus der Literatur bekannten System von Haeberlen et al. [2], so stellt man fest, dass die Ergebnisse unseres Verfahrens um etwa

25 Prozent besser sind. Der auf Gauss-Verteilungen basierende Algorithmus von Haeberlen erreicht bei einer Messung in der Positionierungsphase in dem hier vorgestellten Szenario nur eine durchschnittliche Positionierungsgenauigkeit von etwa 1,85 Meter. Dies ist darauf zurückzuführen, dass die meisten bestehenden Positionierungsverfahren versuchen, mit einer minimalen Anzahl von Messungen in der Positionierungsphase auszukommen. Das Anwendungsszenario, das wir hier betrachten, erlaubt es uns aber, über eine längere Zeit hinweg Messungen an einem Ort durchzuführen.

5 Zusammenfassung und Ausblick

Mit dem hier vorgestellten Positionierungsalgorithmus ist es möglich die Position von Studierenden während einer Vorlesung zu bestimmen. Das Verfahren soll in einem nächsten Schritt für eine automatische Kamerasteuerung verwendet werden, so dass ein automatisches Hineinzoomen bei Fragen aus dem Auditorium möglich wird.

Zusätzlich untersuchen wir momentan die Earth-Mover's-Distance, um die Beschränkungen, die durch die Einführung von τ gegeben sind, zu umgehen. Hiervon versprechen wir uns eine weitere Verbesserung der Positionierungsgenauigkeit.

Darüber hinaus untersuchen wir momentan, wie und auf welche Art weitere Sensoren wie beispielsweise Bluetooth zur Genauigkeitssteigerung verwendet werden können. Auch arbeiten wir an einem Verfahren, welches unterschiedliche Signalstärkeinstellungen bei den Access Points verwendet, um so die Position genauer bestimmen zu können.

Literatur

- [1] BAHL, P. ; PADMANABHAN, V. N.: RADAR: An In-Building RF-Based User Location and Tracking System. In: *Proceedings of the 19th International Conference on Computer Communications (Infocom)*. Tel Aviv, März 2000, 775-784
- [2] HAEBERLEN, A. ; FLANNERY, E. ; LADD, A. M. ; RUDYS, A. ; WALLACH, D. S. ; KAVRAKI, L. E.: Practical Robust Localization over Large-Scale 802.11 Wireless Networks. In: *Proceedings of the Tenth ACM International Conference on Mobile Computing and Networking (MobiCom)*. New York, September 2004, 70-84
- [3] KING, T. ; HAENSELMANN, T. ; KOPF, S. ; EFFELSBERG, W. : Positionierung mit Wireless-LAN und Bluetooth. In: *Praxis der Informationsverarbeitung und Kommunikation* 29 (2006), March, Nr. 1, S. 9-17
- [4] SCHEELE, N. : *The Interactive Lecture: A new Teaching Paradigm based on Pervasive Computing*, Fakultät für Mathematik und Informatik, Universität Mannheim, Diss., Februar 2006

Zwischenbericht: Ortsabhängige Auto- Adresskonfiguration in mobilen ad hoc Netzen

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

Zusammenfassung: Dieses Paper beschreibt einen Zwischenstand aus einem laufenden Vorhaben. Ziel ist die Entwicklung eines Verfahrens zur ortsabhängigen Adressvergabe bei der Konfiguration von mobilen Knoten in einem ad hoc Netzwerk. Die automatische Adressvergabe in einem solchen Netzwerk stellt ein schwieriges Problem dar. Eine zentrale Kontrollinstanz dafür zu verwenden, scheidet auf Grund der Natur eines ad hoc Netzwerks aus. Alle bekannten dezentralen Algorithmen laufen in zwei Phasen ab. In der ersten Phase wird eine Adresse ausgewählt, in der zweiten Phase wird eine Duplikatsprüfung durchgeführt. Die Duplikatsprüfung skaliert sehr schlecht mit der Anzahl der Knoten. Problematisch sind insbesondere Fälle, in denen das ad hoc Netzwerk zeitweilig in mehrere Teilnetze zerfallen ist. Eine Möglichkeit zur Verbesserung der Komplexität stellt die Adressvergabe in einem örtlich begrenzten Kontext dar. Ein solches Verfahren stellen wir hier ansatzweise vor.

1 Einleitung

Ad hoc Netze zeichnen sich durch den Wegfall einer zentralen Infrastruktur aus. Dadurch lassen sich solche Netze schnell und kostengünstig aufbauen. Im Gegensatz zu traditionellen Infrastrukturnetzen leiten in einem solchen Netzwerk je nach verwendetem Routing-Algorithmus einige oder alle Knoten nicht für sie selbst bestimmte Datenpakete weiter. Zur Kommunikation mit den übrigen Knoten muss jeder Knoten mit einer Adresse konfiguriert werden. Das kann manuell geschehen. Wesentlich weniger Administrationsaufwand ist hingegen notwendig, wenn neu hinzukommende Knoten automatisch konfiguriert werden.

In drahtgebundenen und drahtlosen Infrastrukturnetzen gibt es eine Reihe von Möglichkeiten, diese Konfiguration vorzunehmen. Dabei müssen doppelte Adressen unbedingt vermieden werden. Häufig wird ein zentraler Dienst, wie zum Beispiel DHCP, zur eindeutigen Vergabe der Adressen eingesetzt. Kann oder soll ein zentraler Dienst nicht eingesetzt werden, besteht die Möglichkeit, eine Adresse frei zu wählen und auf Duplikate innerhalb des Netzwerkes zu überprüfen. Dabei wird überprüft, ob die zufällig gewählte Adresse bereits vergeben ist. Falls das der Fall ist, wird eine neue Adresse gewählt und der Vorgang wiederholt sich. Solange sich alle Knoten in einer gemeinsamen Broadcast-Domain befinden und ein ausreichend großer Adressraum vorhanden ist, arbeitet auch dieses Verfahren ausreichend effizient und zuverlässig.

In ad hoc Netzen sind einige Besonderheiten zu berücksichtigen. So kann es wegen der Mobilität der Knoten jederzeit vorkommen, dass das Netzwerk in zwei nicht miteinander verbundene Teilnetze zerfällt. In einem solchen Fall kann natürlich nicht überprüft werden, ob eine Adresse bereits vergeben wurde. Bei einer anschließenden Wiedervereinigung der Teilnetze kann es dann zu einem Adresskonflikt kommen.

Ein weiterer Nachteil entsteht durch die begrenzte Reichweite der Funkverbindungen. Dadurch können Broadcasts nicht alle Knoten direkt erreichen, sondern müssen ebenfalls über mehrere Hops weitergeleitet werden. Eine Überprüfung auf Adressduplikate muss trotzdem

das gesamte Netzwerk erfassen. Der Aufwand für ein solches Verfahren skaliert schlecht mit der zunehmenden Anzahl von Knoten.

In diesem Papier stellen wir ein Verfahren in Ansätzen vor, das die genannten Nachteile auszugleichen versucht. Kernprinzip ist dabei, dass Adressen in Abhängigkeit von der Position eines Knotens vergeben werden. Dadurch reduziert sich der Aufwand für die Duplikatskontrolle auf einen engeren, geografisch bestimmaren Raum.

Dieser Kurzbeitrag ist wie folgt gegliedert: Kapitel 2 gibt einen sehr kurz gehaltenen Überblick über den Stand der Technik in den Bereichen ad hoc Netze, Adress-Autokonfiguration und Positionsbestimmung in ad hoc Netzen, Kapitel 3 erläutert unser Konzept zur ortabhängigen Adressvergabe und Kapitel 4 diskutiert die bereits vorhandenen Ergebnisse.

2 Technologische Basis

In diesem Abschnitt geben wir einen kurzen Abriss über die vorhandenen Technologien, die für unser Vorhaben von Bedeutung sind.

2.1 Ad hoc Netze und verwandte Technologien

Ad hoc Netze [9] stellen eine einfache Möglichkeit dar, vor allem mobile Knoten ohne fest installierte Infrastruktur miteinander zu vernetzen. Bei WLAN-Netzen nach IEEE 802.11 ist der ad hoc Modus im Standard vorgesehen. Für die den ad hoc Netzen verwandten Mesh-Netzwerke [10] wurden zusätzlich eine Reihe von Routing-Verfahren entwickelt. Da auch beim Routing eine schlechte Skalierbarkeit mit der Anzahl der Knoten zu verzeichnen ist, wurden Auswege gesucht. Ein Ausweg stellen geografische Routingverfahren dar [11]. Hierbei werden in der Kostenfunktion die Koordinaten des Ziels und möglicher Zwischenknoten berücksichtigt. Umfangreiche Tests und Simulationen [12] haben gezeigt, dass durch geografisches Routing die Skalierbarkeit in Abhängigkeit von der Anzahl der Knoten wesentlich verbessert werden kann.

2.2 Adress-Autokonfiguration

Neben dem zentral arbeitenden DHCP [7] existiert mit IP Auto Config [8] eine Möglichkeit, die Adresskonfiguration ohne zentral arbeiten Instanz zu realisieren. Dabei wählt das zu konfigurierende Gerät eine IP-Adresse aus einem vorgegebenen Bereich. Mittels ARP-Requests [13] wird innerhalb der Broadcast-Domain überprüft, ob die gewählte Adresse bereits vorhanden ist. Ist das der Fall, wird eine neue Adresse zufällig ausgewählt.

In ad hoc Netzwerken gibt es unter anderem mit [14] ebenfalls Möglichkeiten zur automatischen Adressvergabe. Dabei ist zu berücksichtigen, dass diese Verfahren an ihre Grenzen geraten, sobald das Netzwerk in mehrere Teilnetze zerfällt.

2.3 Positionsbestimmung in ad hoc Netzen

In drahtlosen Netzen wird die Position häufig über die Messung der Feldstärke zu umliegenden, ortsfesten Knoten bestimmt. Daneben besteht die Möglichkeit, Laufzeiten zwischen Knoten zu bestimmen und darüber die Position abzuleiten. In einem Netzwerk mit herkömmlicher WLAN Hardware kommt nur die Feldstärkemessung in Frage.

Da die Feldstärke starken Schwankungen unterliegt, werden in der Regel statistische Verfahren eingesetzt, die zu gegebenen Messwerten der Feldstärke den wahrscheinlichsten Aufenthaltsort berechnen [1][2][3]. Um ein möglichst realistisches Modell der Feldstärke - eine sogenannte „Radiomap“ - zu erhalten, werden häufig Kalibrierungsmessungen eingesetzt. Dabei wird in einem möglichst engen Raster die Feldstärke der umliegenden Knoten gemessen. Für dazwischen liegende Punkte werden die Feldstärkevektoren entsprechend eines Ausbreitungsmodells (z.B. Wall Attenuation [5]) interpoliert.

In einem mobilen ad-hoc Netzwerk erübrigt sich das Aufstellen einer statischen Radiomap wegen der Mobilität der Knoten. Unter anderem in [4] haben wir eine Möglichkeit vorgestellt, mit sehr wenigen ortsfesten Knoten auszukommen und über ein iteratives Verfahren eine ausreichend präzise Radiomap zu erstellen. Dabei werden die zu erwartenden Feldstärken zwischen ortsfesten Knoten bestimmt und mit der Realität verglichen. Der dabei gemessene Fehler wird in der Fläche interpoliert. Die wahrscheinlichste Position eines mobilen Knotens wird von diesem berechnet. Der so in seiner Position bestimmte Knoten kann wiederum für andere Knoten zur Positionsbestimmung dienen. Die zu erwartende höhere Ungenauigkeit wird dabei berücksichtigt. Durch die zu erwartende Überbestimmtheit der Positionsangaben kommt es zu einer iterativen Annäherung.

3 Konzept

In diesem Abschnitt stellen wir kurz die Basisidee vor und motivieren die gewählte Architektur. Ziel ist die Entwicklung eines effizienten Verfahrens zu Auto-Adresskonfiguration.

3.1 Basisidee der ortsbasierten Autokonfiguration

Wir möchten Adressen anhand der Position der neu zu konfigurierenden Knoten vergeben. Dabei müssen eine Reihe von Beschränkungen beachtet werden. 1. Es ist vorgesehen, dass sich Knoten nach der Konfiguration aus dem Gebiet, in dem sie konfiguriert wurden, herausbewegen. 2. Knoten können das Netzwerk ohne Ankündigung verlassen. 3. Es stehen nur beschränkt viele Adressen zur Verfügung. 4. Es kann zum Auseinanderfallen des Netzwerkes oder zur Wiedervereinigung kommen. 5. Die Gesamtausdehnung des Netzwerkes bleibt konstant - anderenfalls erfolgt eine Neukonfiguration.

Wir gehen davon aus, dass in einem ad hoc Netzwerk nahezu zwingend ein Link State Routing Verfahren eingesetzt wird. Dadurch ist allen am Netzwerk bereits beteiligten Knoten eine vollständige Liste der übrigen Knoten bekannt, solange das Netz miteinander verbunden ist oder bei der Konfiguration neuer Knoten verbunden war.

3.2 Untersuchung der statistischen Verteilung der Knoten in einem ad hoc Netzwerk

Wir gehen weiter davon aus, dass die Knoten nicht gleichmäßig in der Fläche verteilt sind. Das ermöglicht uns eine effizientere Tessellation der Fläche, also eine angepasste Aufteilung. In jeder Teilfläche sollen flächenspezifische Adressen vergeben werden. Nach Angaben in der Literatur [15] und ersten Tests in realen Netzwerken gehen wir von einer Poisson-Verteilung der Anzahl der Knoten aus pro gleich großer Teilfläche. Es gibt also Teilbereiche, in denen die Knoten geballter auftreten und Flächen mit deutlich niedrigerem Knotenbesatz. Wir haben die Positionen der gegenwärtig etwa 150 Knoten in dem von uns unterstützten Mesh-Netzwerk [6] untersucht und konnten mittels eines Anpassungstests eine entsprechende Verteilung der Knoten nachweisen. Entsprechend soll die Fläche nicht gleichmäßig aufgeteilt werden.

3.3 Aufteilung der Fläche

Die Aufteilung der Fläche soll in Quadrate erfolgen, wie in Abbildung 1 dargestellt. Das Quadrat 0. Ordnung umfasst die gesamte Fläche. Es wird in 4 gleich große Quadrate 1. Ordnung unterteilt. Das Teilquadrat, das am

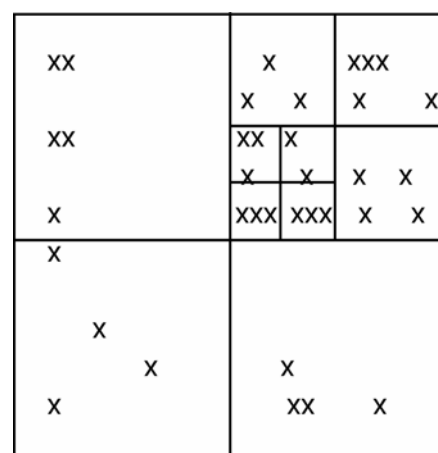


Abb. 1: Tessellation der Fläche

stärksten mit Knoten besetzt ist, wird wiederum in 4 gleiche Teilquadrate unterteilt. Die maximale Ordnung, bis zu der dieses Verfahren fortgesetzt wird, bestimmt sich aus der Anzahl der zur Verfügung stehenden Adressen und der Gesamtfläche des Netzwerks. Nach ersten Tests halten wir Teilflächen mit ca. 10 Knoten für ausreichend. Jeder Teilfläche wird ein spezieller Adressbereich zugewiesen.

3.4 Ablauf

Nach der Wahl einer Adresse im für die aktuelle Position reservierten Adressbereich überprüft der neu konfigurierte Knoten nach einem noch in Einzelheiten zu definierenden Protokoll, ob einer der anderen Knoten in dieser Teilfläche Kenntnis davon hat, dass die Adresse bereits vergeben ist.

Nach Ablauf einer festzulegenden Lease-Zeit muss ein Knoten seine Adresse erneut bestätigen. Hat er zwischenzeitlich die ursprüngliche Teilfläche verlassen, muss er in der aktuellen Teilfläche eine Adresse wählen.

Ein bisher ungelöstes Problem stellt die Positionsbestimmung eines noch nicht konfigurierten Knotens dar. Wir denken zur Zeit über die Nutzung von Adressen nach, die im ad hoc Netzwerk nicht geroutet werden, aber trotzdem zur Kommunikation mit den unmittelbaren Nachbarknoten verwendet werden können.

4 Diskussion

Das vorgestellte Verfahren scheint sehr viel versprechend für eine effiziente Autokonfiguration zu sein. Wir sind aktuell dabei, Protokolle zu definieren und verschiedene Szenarien mit NS-2 zu simulieren. Noch offene Probleme scheinen in kurzer Zeit lösbar zu sein.

Literatur

- [1] T. Kitasuka, T. Nakanishi, and A. Fukuda: "Wireless LAN based indoor positioning system wips and its simulation", in Proc. of The 2003 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'03), IEEE Press, Victoria, BC., Aug. 2003.
- [2] S. Ito and N. Kawaguchi: "Bayesian based location estimation system using wireless LAN", In Proc. of the Third IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'05), IEEE Press, Kauai, Hawaii, Mar. 2005.
- [3] V. Seshadri, G. V. Zaruba, and M. Huber: "A bayesian sampling approach to in-door localization of wireless devices using received signal strength indication", in Proc. of the Third IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'05), IEEE Press, Kauai, Hawaii, Mar. 2005.
- [4] Mundt, Thomas: Authenticated Positioning in Wireless Mesh Networks, In: Proc. of IEEE GCC 2006, IEEE Press, Manama, Bahrain, March 2006
- [5] H. Hashemi. The indoor radio propagation channel. In Proceedings of the IEEE, pages 943–968, July 1993.
- [6] Liste der Knoten im Opennet Rostock: http://wiki.opennet-initiative.de/index.php/Opennet_Nodes
- [7] R. Droms; RFC 2131 – Dynamic Host Configuration Protocol
- [8] R. Troll; Internet Draft: DHC-IPV4-AUTOCONFIG - "Automatically Choosing an IP Address in an Ad-Hoc IPv4 Network"
- [9] C.E. Perkins; Mobile-IP, Ad-Hoc Networking, and Nomadcity; COMPSAC '96 - 20th Computer Software and Applications Conference; August 1996
- [10] Tomas Krag und Sebastian Büttrich; Wireless Mesh Networking; O'Reilly 2004
- [11] Aaron Zollinger, Roger Wattenhofer (Herausgeber); Networking Unleashed: Geographic Routing and Topology Control in Ad Hoc and Sensor Networks; ISBN: 386628022X; Hartung-Gorre, 2005
- [12] Ahmed Sobeih, Wei-Peng Chen, Jennifer C. Hou, Lu-Chuan Kung, Ning Li, Hyuk Lim, Hung-Ying Tyan, Honghai Zhang, J-Sim: A Simulation Environment for Wireless Sensor Networks; In Proc. of 38th Annual Simulation Symposium (ANSS'05) ; 2005
- [13] David C. Plummer; RFC826: An Ethernet Address Resolution Protocol
- [14] C. Perkins, et. al, "IP Address Auto configuration for Ad Hoc Networks", draft, IETF, Nov. 2001
- [15] Peter J. Diggle. Statistical Analysis of Spatial Point Patterns. Arnold Publishers, 2003.

Self-Organizing Information Systems for Disaster Management

Artin Avanes, Timo Mika Gläßer, Markus Scheidgen

Humboldt-Universität zu Berlin,
Institute for Computer Science,
10099 Berlin, Germany

{avanes, glaesser,scheidge}@informatik.hu-berlin.de

1 Introduction

In recent years, the utilization of self-organizing, wireless, ad-hoc networks, such as sensor networks and embedded systems, has become more and more attractive. Potential applications include automation of business activities, traffic control as well as advanced monitoring systems for buildings or wildlife.

Our goal is to integrate different self-organizing networks with geographic information systems to implement computer-supported disaster management. Hence, in our understanding a so-called *self-organizing information system* contains both dynamic, self-contained networks and stable information systems. A self-organizing network consists of individual nodes which execute location-based services, i.e. measure the temperature or vibration in a specific region. Even if these nodes autonomously monitor their environment, they are not isolated but interact with other nodes. This is the only way to achieve a gainful analysis and evaluation of collected data. An application example how different self-organizing networks can efficiently cooperate in a disaster event is given in Figure 1.

Such an integration poses new technical challenges. First, these networks are characterized by self-contained, heterogeneous nodes which independently perform services for higher-level system layers. There is no central component co-

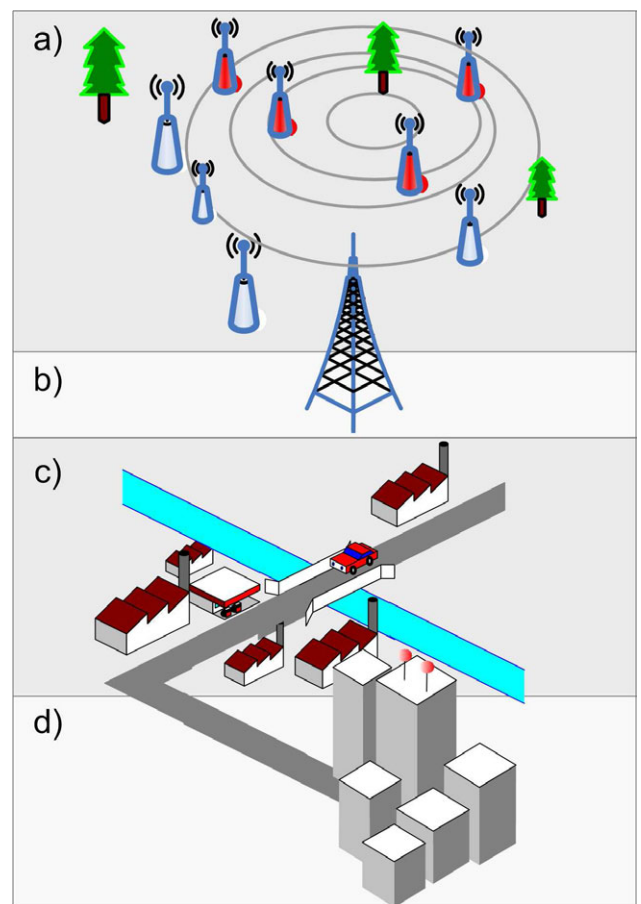


Figure 1: A possible scenario: Sensor nodes of network a) detect an earthquake. The information is transmitted to the nearest gateway node, e.g. b). Embedded systems in networks c) and d) will be informed to close the gate on the bridge and the gas supply for buildings respectively.

This work is sponsored by the Graduate School METRIK.

ordinating the different activities through out the system. Second, self-organizing networks have to deal with frequent topology changes. The dynamics of the system landscape is caused either by mobile system participants or by devices failures. Finally, limited resources require strategies to reduce communication efforts for an enhanced system life span.



Figure 2: Logical layers of our system architecture.

Our ambition is to build a multi-layered system architecture by defining common interfaces and requirements as well as concepts and algorithms to face the challenges mentioned above. In the following sections, we will discuss the relevant building blocks of our system infrastructure stated in Figure 2.

2 Data Management

The data management layer of our distributed information system will be responsible for providing a coherent view of the underlying network and the data gathered. The interface abstracts from the network dynamics, e.g. node mobility or failure, and heterogeneity, e.g. computational faculties or availability of resources. It performs node and data localisation for higher-level layers while providing a rich and intuitive interface.

Even though different approaches of data management in peer-to-peer and sensor networks exist [12, 7], we feel they do not satisfy the requirements in the domain of disaster management where guaranteed serviceability is needed.

Furthermore, to the best of our knowledge, no existing architectures has proven to scale beyond a few dozens of nodes, e.g. the largest deployment of TinyDB [7] contains 80 nodes.

Declarative, Event-Driven Interface: The data management layer will provide a declarative, SQL-like interface capable of relating to user-defined events as shown in the example query below. The query in Figure 3 triggers an alarm if an earthquake (event is verified collaboratively) in some area is reported.

```

ON EVENT earthquake
SELECT AVG(vibration), area
FROM sensors
WHERE
  vibration > t AND location WITHIN( $x_1, y_1, x_2, y_2$ )
GROUP BY SQUARES( $1km^2$ )
SAMPLE PERIOD 5s TRIGGER ACTION alarm

```

Figure 3: Sample Query

The interface of the data management will provide a intuitive query language and data model for the workflow and modelling layers.

Fault-Tolerance: Providing a robust data management layer will be one of the most-critical challenges in the domain of disaster management where the system must guarantee serviceability in emergency situations. We will investigate the limits under which most parts of the network will stop working and conduct research on caching and replication algorithms to enable the system to cope with the loss of potentially a large number of nodes.

Energy-Awareness: Energy-aware query optimization and execution will be a key requirement for our system to conserve energy resources of the autonomous sensor nodes. To achieve this goal we will conduct research in the area of in-network execution of queries including aggregations and user-defined functions. While for the former some approaches e.g. [10] exist, the latter remains an open issue for sensor networks. Additionally one might employ statistical queries on a small fraction of the available nodes to assemble approximate results in less time and with fewer communication efforts compared to the exact query.

3 Decentralized Workflow Management

To coordinate and streamline activities executed by the nodes of self-containing networks before and after the disaster, a robust, flexible and adaptable workflow management system (WFMS) is required. In general, a WFMS supports the definition, creation and execution of different workflow activities through the utilization of software on the system nodes. Even if several approaches in this area have been proposed, none of them fully satisfy our specific requirements. Some solutions [8][5] assume a central monitoring component which is not applicable for a decentralized, dynamic network. Other approaches are fully-decentralized but assume either a stable communication infrastructure or do not consider resource-constraint devices [1]. To bring all these aspects together, we will focus on the following research issues:

Dynamic Resource Management: One major task of WFMS is the assignment of activities to resources, e.g. given devices. In decentralized, dynamic systems resources frequently move or are disconnected. Especially for disaster management, this is critical if deployment of location-dependent services fails. Therefore, we will investigate efficient resource tracking strategies which consider the trade-off between information accuracy and communication effort. *Optimistic strategies* (inaccuracy high, communication efforts low) are compared with *pessimistic strategies* (accuracy low, communication efforts high).

Failure Handling and Data Management: Since the failure of resource-constraint devices is very likely, algorithms are needed to guarantee continuation of system operations if such failure events occur. We will focus on the classification of failure events such as communication (including network partitioning), hardware and software failures. Based on the identified failure classes, we will propose suitable failure treatment protocols. In this context, different approaches will be evaluated: (a) migration of process and data (b) replication of process and data and (c) consensus protocols.

Similar solutions [6][11] exist for distributed systems, but do not consider our specific requirements. Again, we will compare the communication costs for each strategy and failure event. The goal is that the WFMS is able to dynamically select an optimized failure protocol based on the communication costs for a specific failure event.

4 Modelling

Disaster management applications for self-organized information systems include combinations of: (1) analysis of accumulated data, i.e. detecting disaster events; (2) spatial information systems; (3) distributed automated decision conclusion and information propagation. Each of these applications requires a different modelling paradigm: A disaster is characterised by a mathematical model of sensor data; information systems are based on data modelling, query languages, and user interfaces; computing event-based decisions is described through process calculi or automata theory.

Those applications demand three main requirements for the languages used to model them. First, disaster management denotes a specific domain with specific vocabulary. Second, the diversity of the applications demands a mixture of existing modelling paradigms. And third, the targeted platform will include novel concepts based on the characteristics of the underlying architecture. Therefore, we need to specify domain specific languages (DSL) that cross multiple paradigms with domain specific vocabulary tailored for self-organized networks. This also requires a development process for those languages which can integrate different modelling methods into a concise DSL workbench.

Languages should form a transparent abstraction from the details of the underlying platform. But in this specific domain, an application should interact with platform details. For instance, in case that a disaster management application detects a disastrous event, it should inform the platform to trigger suitable failure handling protocols, e.g. replicate critical data into save regions of the network. In general this forms an additional requirement for platform and languages to

express and implement such cross layer concerns.

Existing technologies for effective language development in domain specific environments [2] are either based on translational [3] or operational [9] semantics. Language integration is based on meta-model based language definitions and mappings between meta-models [4]. These techniques allow a tight integration on syntax level, but their support for semantics is limited to a single method. Therefore, it is challenging to combine different modelling paradigms which require different semantic domains.

5 Summary

In this architecture vision paper, we discussed different research challenges for the design and implementation of decentralized information systems on top of dynamic, highly flexible, self-organizing networks. Existing research in this area mainly addresses low level concerns, such as routing protocols and energy consumption. To cope with the growing complexity of future applications, more sophisticated architectures are required. In this context, we presented a multi-layered system infrastructure and discussed challenges as well as possible solutions regarding each layer. We also described potential interfaces and dependencies, e.g. between the workflow and data management layer. Finally, we focused on a model-based approach that will be applied to all layers of the system, i.e. required for modelling of workflows and data.

References

- [1] ALONSO, G., AGRAWAL, D., ET AL. Exotica/fmqm: A persistent message-based architecture for distributed workflow management. In *Proceedings of the IFIP WG8.1 Working Conference on Information Systems Development for Decentralized Organizations. Trondheim, Norway, 1995*.
- [2] AMYOT, D., FARAH, H., AND ROY, J.-F. Evaluation of Development Tools for Domain-Specific Modeling Languages. In *Proceedings of the 5th Workshop on System Analysis and Modelling* (2006), Springer.
- [3] CHEN, K., SZTIPANOVITS, J., ABDELWALHED, S., AND JACKSON, E. Semantic Anchoring with Model Transformations. In *Model Driven Architecture – Foundations and Applications* (2005), Springer.
- [4] CZARNECKI, K., AND HELSON, S. Classification of Model Transformation Approaches. In *OOPSLA'03 Workshop on Generative Techniques in the Context of Model-Driven Architecture* (2003).
- [5] DEELMAN, E., BLYTHE, J., GIL, Y., ET AL. Mapping Abstract Complex Workflows onto Grid Environments. *J. Grid Comput.*, 1 (2003).
- [6] LIU, M. L., AGRAWAL, D., AND ABBADI, A. E. The Performance of Two Phase Commit Protocols in the Presence of Site Failures. *Distrib. Parallel Databases*, 2 (1998).
- [7] MADDEN, S. R., FRANKLIN, M. J., HELLERSTEIN, J. M., AND HONG, W. Tinydb: An Acquisitional Query Processing System for Sensor Networks. *ACM Trans. Database Syst.*, 1 (2005).
- [8] MEDINA-MORA, R., WONG, H. K. T., AND FLORES, P. ActionWorkflow as the Enterprise Integration Technology. *IEEE Data Eng. Bull.*, 2 (1993).
- [9] MULLER, P.-A., FLEUREY, F., AND JÉZÉQUEL, J.-M. Weaving Executability into Object-Oriented Meta-languages. In *Model Driven Engineering Languages and Systems: 8th International Conference* (2005), Springer.
- [10] SHARAF, A., BEAVER, J., LABRINIDIS, A., AND CHRYSANTHIS, K. Balancing energy efficiency and quality of aggregate data in sensor networks. *The VLDB Journal*, 4 (2004).
- [11] SKEEN, D. A Quorum-Based Commit Protocol. In *Berkeley Workshop* (1982).
- [12] STEGMAIER, B., KUNTSCHKE, R., AND KEMPER, A. Streamglobe: adaptive query processing and optimization in streaming p2p environments. In *DMSN '04: Proceedings of the 1st international workshop on Data management for sensor networks* (2004), ACM Press.

Eine Rechteverwaltungsplattform für heutige LBS-Anwendungen

Tobias Kölsch*

Marc Wilhelm†

Zusammenfassung

In dieser Arbeit stellen wir die Implementierung¹ einer Plattform zur Datenübermittlung, Bezahlung und Rechteverwaltung in Ortsbezogenen Diensten vor.

Die Plattform bietet für den Kunden hinsichtlich des Datenschutzes und der Rechteverwaltung eine klare Verbesserung zu anderen heute existierenden Systemen. Da sie die Möglichkeit erhalten den Zugriff feingranular einzustellen. Ein weiterer Vorteil dieser Plattform ist, dass sie insbesondere auf der Seite der Dienstanbieter und der Endkunden sehr niedrige Infrastrukturvoraussetzungen stellt und entsprechend relativ kostengünstig in eine existierende Produktivumgebung eingegliedert werden könnte.

1 Problem & Ansatz

Mit der weiten Verbreitung des Mobilfunks ist das Interesse an der Standortinformation von mobilen Telekommunikationsteilnehmern für kontextbezogene Dienste gewachsen. Viele Mobilfunkanbieter bie-

ten 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.

Ein weiteres Problem ist der Zugangsschutz zur Ortsinformation der Benutzer. Die meisten Mobilfunkanbieter bieten einen grundsätzlichen Schutz der Benutzer. Dieser besteht im Allgemeinen darin, dass der Benutzer angeben kann, ob er damit einverstanden ist einen Dienst zu nutzen, oder nicht. Allerdings sind die Einstellungen in einigen Fällen kaum auffindbar und/oder so grobkörnig, dass sie praktisch nur mittelmaßigen Schutz vor Missbrauch durch den Dienstanbieter bieten.

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. Wie in Abb. 1 veranschaulicht. Eine umfassende Begründung für die Sinnhaftigkeit eines solchen Mittlers findet sich in [1].

Zusätzlich zur Verwaltung der Ortsdaten, wird aber auch die Bezahlung des Dienstes durch den Mittler geschützt, sodass nur sol-

*koelsch@i4.informatik.rwth-aachen.de

†marc.wilhelm@futureaspects.de

¹Die Erstellung des Prototypen wurde durch das 6. Rahmenprogrammes der europäischen Kommission unterstützt.

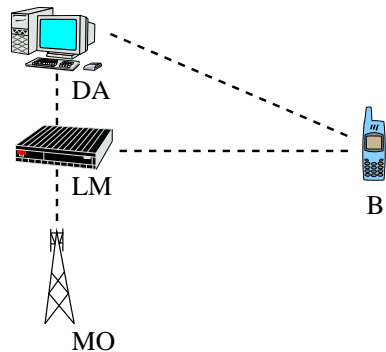


Abbildung 1: Der Benutzer fragt einen Dienst beim Dienstanbieter an. Dieser benutzt den Mittler um die Ortsinformation des Benutzers vom Mobilfunkbetreiber zu ermitteln. Über den Mittler kann der Benutzer den Zugriff auf seine Daten einschränken.

che Beträge abgebucht werden können, denen der Kunde im Vorhinein zugestimmt hat.

Dadurch dass der Mittler zwischen Mobilfunkanbieter und Dienstanbieter ist, wird aber auch die Dienstonutzung des Kunden vor dem Mobilfunkanbieter verschleiert, was den Heutzutage recht umfangreichen Zugriff des Mobilfunkanbieters auf Kundenprofile etwas schmälert. Der Mittler sieht zwar die Dienstonutzung, erfährt aber die Identität des Benutzers nicht, da dieser pseudonymisiert wird.

In der implementierten Lösung erhält der Mittler die Anfragen der Dienstanbieter, überprüft ob die Beantwortung zulässig ist. In dem Falle einer positiven Antwort, werden dem Dienstanbieter die Ortsdaten übermittelt und er erhält die Möglichkeit die Dienstonutzung über den Mobilfunkanbieter abzubuchen. Falls die Beantwortung der Anfrage nicht zulässig ist, erhält der Anbieter die Möglichkeit den Benutzer auf

eine spezielle Administrationsseite umzuleiten, auf der dieser seine Einstellungen ändern kann.

Bei der Umleitung wird dem Kunden ein Einstellungsvorschlag gemacht, den er gegebenenfalls nur akzeptieren muss. Er kann aber auch selber seine Einstellungen vornehmen.

Nachdem der Benutzer dem Dienstanbieter den Zugang auf seine Daten gewährt hat, wird er wieder zur Dienstseite zurückgeleitet, und kann seine Anfrage wiederholen. Welche dann erfolgreich beantwortet werden kann.

Des Weiteren hat der Benutzer die Möglichkeit über die Verwaltungsseite eine Übersicht über die auf seinem Konto vollzogenen Aktionen (sowohl Ortungen, als auch Abbuchungen) einzusehen.

Im speziellen Fall haben wir nur WAP Basierte pull-Dienste betrachtet, allerdings lässt sich die Plattform mit relativ wenig Aufwand auch um Einstellungen für push-Anwendungen erweitern.

Der Prototyp ist im Rahmen des IST PRIME Projektes unter verschiedenen Gesichtspunkten evaluiert worden [2].

2 Implementierung

In unserer Architektur haben wir einen Mobilfunkanbieter (MO) und einen von ihm getrennten Ortsbasierten Dienstanbieter (DA). Beide stellen dem Benutzer (B) gemeinsam einen Dienst zur Verfügung. Der MO dient dabei als Kommunikations- und Positionsanbieter. Wie gesagt kommunizieren die beiden Anbieter nicht direkt, sondern über einen Mittler (LM).

2.1 Ablauf: Dienstnutzung

Im folgenden werden die einzelnen Schritte vorgestellt, die durchgeführt werden um den Dienst zu erbringen. Zunächst wird der Fall vorgestellt, dass der Benutzer dem Dienstanbieter den Zugriff auf seine Daten gewährt hat.

B→DA: Dienstanfrage (Abb. 2(a))

DA→LM: Lokalisierungsanfrage (enthält Preis und IP Adresse)

LM→MO: Anfrage zur Auflösung der IP in das permanente Benutzerpseudonym

MO→LM: Permanentes Benutzerpseudonym

LM↔: Wertet Benutzereinstellungen bzgl. DA und Preis aus
→ positives Ergebnis

LM→MO: Reserviert Buchung

MO↔: Speichert Buchungsreservierung

LM→MO: Erfragt Ortsinformation

MO→LM: Ortsinformation

LM→DA: Ortsinformation & Zahlungsreservierung

DA↔: Ergebnisermittlung

DA→LM→MO: Buchung

MO→LM→DA: Buchungsbestätigung

DA→B: Dienstbringung (Abb. 2(b))

An dem fünften Schritt ist eine Abzweigung. Falls die Auswertung dort negativ ist, wird statt des verbleibenden Verlauf der folgende ausgeführt. Danach wird dann der erste Verlauf wiederholt.

LM↔: Wertet Benutzereinstellungen bzgl. DA und Preis aus
→ negatives Ergebnis

LM↔: Merkt sich erfolglose DA-Anfrage

LM→DA: Teilt negatives Auswertungsergebnis mit

DA→B: Teilt unzureichende Rechte mit und leitet ihn auf Administrationsseite um (Abb. 2(c))

B→LM: Fragt configurationsseite an

LM→MO: Stellt Anfrage zur Auflösung der IP in das permanente Benutzerpseudonym

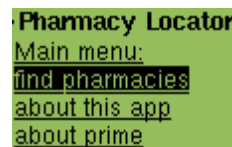
LM↔: Sucht nach letztem erfolglosen Dienstbringungsversuch

LM→B: Bietet Zugangsberechtigung für DA an

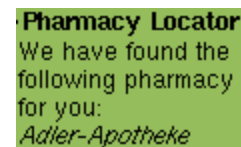
B→LM: Nimmt Zugangsberechtigung für DA an (Abb. 2(d))

LM→B: Leitet ihn auf die DA Anfrageseite zurück

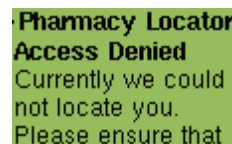
B→DA: Dienstanfrage (startet den ersten Ablauf)



(a) Startseite



(b) Ergebnis



(c) Startseite



(d) Ergebnis

Abbildung 2: Screenshots

2.2 Technische Aspekte

Der Prototyp ist in Java implementiert worden, wobei die einzelnen Server innerhalb von Tomcat 5.5.9 Containern liefen. Die inter-Server Kommunikation fand anhand eines einfachen Request-Response Protokolles statt, wobei das PRIME² Identitätsverwaltungssystem zur Prüfung des Zugangsschutzes und zur Übertragung der persönlichen Daten verwendet wurde.

Die Sicherheit des Prototypen basiert auf Netzwerk Authentifizierung, diese bietet i.A. zwar keinen sehr hohen Schutz, da Adressen "gespoofed" werden können. Allerdings kann das vereitelt werden, indem sich die DA innerhalb eines VPN mit dem MO befinden.

3 Bewertung & Ausblick

Der vorgestellte Ansatz bringt den Vorteil, dass der Mittler von der eigentlichen Ortsdatenquelle abstrahiert. Ein Dienstanbieter verbindet sich zu ihm unabhängig davon, ob die Daten etwa von einem WLAN-Hotspot oder aus einem GSM-Netz stammen. Außerdem erfährt die Datenquelle wenig über den Dienst, der vom Benutzer beansprucht wird.

Die technische Rechteverwaltung durch eine vertrauenswürdige Partei erleichtert die Verhandlungstechnische und Betriebliche Seite, da Schutzmechanismen nicht mehr Vertraglich ausgehandelt, und manuell überprüft werden müssen.

Die Architektur versteckt die Komplexität des Systems relativ gut vor dem Benutzer, der zur Dienstnutzung lediglich

zwischen 1 und 5 Klicks eingeben muss. Die Komplexität für den Dienstanbieter ist durch die Verwendung einfacher Protokolle auch niedrig gehalten.

Über die für den Benutzer einsehbaren Transaktionsgeschichte hat dieser einen guten und zeitnahen Überblick darüber was mit seinen Daten (und seinem Geld) geschieht.

Die Verwendung von WAP ermöglicht die Dienstnutzung für die meisten handelsüblichen Mobilfunkgeräte.

Im folgenden soll der Ansatz nun auf Push-Dienste erweitert werden. Dafür wird bei den Einstellungen die Granularität der Zustimmungseinschränkung sowohl für die Lokalisierung als auch für die Bezahlung noch einmal erhöht. Außerdem wird der Schutz der Privatsphäre ebenfalls dadurch erhöht, dass ein spezieller Browser entwickelt wird, der Pseudonymisierung von sich aus unterstützt. Dadurch wird die Abhängigkeit von der Netzwerk-Authentifizierung aufgehoben.

Literatur

- [1] Tobias Kölsch, Lothar Fritsch, Markulf Kohlweiss, and Dogan Kesdogan. Privacy for profitable location based services. Springer, 2005.
- [2] PRIME WP 4.2. Evaluation of initial Application Prototypes. Technical report, 2006. https://www.prime-project.eu/prime_products/reports/eval/.

²www.prime-project.eu

A MIDDLEWARE FOR DECENTRALIZED WIRELESS NETWORKS

Wolfgang Golubski
Zwickau University of Applied Sciences
FB Physikalische Technik / Informatik
Dr.-Friedrichs-Ring 2a, 08056 Zwickau, Germany
golubski@fh-zwickau.de

1. INTRODUCTION

Mobile computing is omnipresent. The number of devices (like handhelds, personal digital assistants, mobile phones, laptops) equipped with wireless network interfaces and their facilities to interconnect is continuously growing. The systems are characterized by (1) wireless communication, (2) self-organization, (3) spontaneous collaboration and (4) limited hardware resources (memory, cpu, communication bandwidth). Due to these characteristics host and service discovery is one of the most important challenges in developing mobile applications.

In this paper we present a middleware for mobile wireless networks being suitable for application development in the context of collaborations (like information and data sharing services, entertainment and game services, chat and instant messaging, profiling). The interesting fields of application are:

- Tourist information - Tourists in a city can collect a lot of interesting data about the city like sights or catering. When a tourist comes new to the city, he can participate on the experiences of other tourists by exchanging respectively downloading their data collection (i.e. experiences).
- Conference situation - At a large conference, with participants from all over the world, one would like to meet persons with similar interests.
- Entertainment - Network gaming is a promising increasing business industry.

The middleware called MaJo acts as an intermediary between the hardware network interface and the application layer. MaJo controls the complete communication area including host and service discovery and management. That means that the software developer is unburdened from recurrent tasks and can focus on the real important things (like business logic, data modeling and user interfaces). We investigate and discuss the facilities, advantages and drawbacks of using our middleware approach.

As a case study we realized a game service based on the middleware. The application is a service of MaJo. The game we have implemented is a German card-game called *Schwimmen* (swimming).

2. THE MIDDLEWARE MAJO

The MaJo middleware was implemented in J2ME (Java 2 Micro Edition) and was tested on iPAQ H3630 and H3850 handhelds equipped with WLAN. As Java virtual machine the J9 from IBM [IBM] was used.

The middleware MaJo offer the developers an interface which they can use to develop mobile distributed applications for networks in a comfortable way.

Applications developed by using MaJo are regarded as services. The concepts service and application are therefore used synonymously. The developed middleware is put on the open source JRRA system [JRRA] for peer-to-peer networks and provides an application interface to the outside.

Host and Service Discovery

The middleware is designed for use in mobile ad-hoc networks. Essentially, the middleware must locally organize the complete network infrastructure. The middleware must have self-organized features.

Due to lack of administrative components in a mobile ad-hoc network no information about participants in a central place can be questioned. Therefore the information about the complete network topology must be locally available for each participant.

At the initialization of the middleware it is necessary to find out which mobile devices are active in the environment. Active means here that the middleware is running on the mobile device. Host discovery can exclusively be realized with the help of the UDP broadcast mechanism. The corresponding component of MaJo sends a broadcast packet at the start and informs the other active devices about its own network address. At the receipt of such a packet the already active participants return their own addresses to the sender. If all active devices have received and answered the broadcast packet, the view of the network topology is identical for all participants.

The topology of the network can change fast by the fact that new participants join or leave the network. The ability to take new participants is provided by the broadcast packet mentioned above. If active participants want to leave the ad-hoc network, they can notify the other participants. If an active participant loses, however, the contact to the other participants due to movement from the environment, it cannot cancel. However, leaving must be noticed by all participants. For this a periodical reactive mechanism is realized. The host discovery is realized so that every participant sends periodically a alive packet whereby the network topology is permanently up-to-date with an acceptable delay.

Service Discovery and Current Service Topology

The middleware offers a permanently current view of the available services in the network. The actual service list is immediately accessible and therefore must be stored locally.

At the start of the middleware the new user must inform the other members of the network about his own services. Furthermore, information about the services

which are offered by other participants must be collected. Information about a service of a user consists of the unique service name and the information whether the user just takes part of the service or not. The service discovery concept is similar to the host discovery model stated above and can be combined with it. Each user can start, finish and delete a service. In turn these three activation must be transmitted to the other users of the middleware. Hence, it suffices to send the information consisting of the service name and the way of activation. The alive packets can be extended by some service information.

The Service API

The MaJo middleware provides an abstract programming interface, see Figure 2, and predefines several classes and method implementations suitable for mobile ad-hoc peer-to-peer communication. Service-specific data channels can be implemented in a simple way. Besides this, file transfer will be supported. Services residing on a peer can easily transferred to another peer by means of jar-archives. The only restriction is that a service must be conform to the MaJo API.

3. SWIMMING GAME SERVICE - A CASE STUDY

To represent the strength of the middleware MaJo introduced above exemplarily, a mobile distributed game application was developed based on MaJo. The application is a service of MaJo, the terms *application* and *service* are therefore used synonymously in the following.

The game we have implemented is a German card-game called *Schwimmen* (*swimming*). Swimming is a card-game, where the goal is to receive the highest score as possible.

The swimming service shall be capable in a mobile ad-hoc network on PDAs and make use of MaJo. Since it is a network game, the service requires a lot of communication between the players caused by game requests and information exchanges. The service needs an own data channel for the communication provided by MaJo. At every time, the service must be informed about the state of all participants. Several games are possible in the network at a time. By means of MaJo these requirements can be fulfilled and the game can be implemented in a comfortable way. At least two and at most nine players can be involved in one game. The player names must be unique. Players involved in a game are able to finish the game at every time.

4. EXPERIENCES AND FUTURE WORK

Currently we have developed two applications based on MaJo. The game service is described above. The other service is a chat service. In both cases the software development time was relatively short-time. Before the middleware could be used a lot of work has to be done. The realization of the host and service discovery and management was a time-consuming process.

Since the mobile devices work in an ad-hoc manner without any wireless access points, the initial IP-address must be assigned by hand at the start of the middleware. All peers must be a member of the same subnet.

The network performance is moderate for up to 7 simultaneous users. Similar results are reported in [P2P].

All in all, we believe that MaJo is a good choice to significantly reduce the development time for mobile ad-hoc applications.

The next research steps are (1) integration of an intelligent profile model based on user preferences, (2) involving more security features and trust management and (3) using Bluetooth communication. In a similar project we have made the first experiences with Bluetooth and J2ME.

REFERENCES

- [IBM] IBM Corporation: Websphere Studio Device Developer. <http://www-3.ibm.com/software/wireless/wsdd>.
- [JRRA] Offshore Algorithms, 2004. Java Rockyroad API (JRRA). <http://www.jrra.org>.
- [JXTA] Sun Microsystems. JXTA Initiative. <http://www.jxta.org>.
- [MID] Mascolo, C., Capra, L., and Emmerich, W., 2002. XMIDDLE: Middleware for Mobile Computing (A Survey). *Advanced Lectures on Networking - Networking 2002 Tutorials*, Springer Verlag, Pisa, Italy. volume 2497 of LNCS, pp 20-58.
- [P2P] Wang, A.-I., Sørensen, C.-F., and Fossum, T., 2005. Mobile Peer-to-Peer Technology used to Promote Spontaneous Collaboration. *International Symposium on Collaborative Technologies and Systems (CTS 2005)*, Saint Louis, Missouri, USA, May 15-19.
- [PROEM] Kortuem, G, 2002. A methodology and software platform for building wearable communities. *PhD thesis, University of Oregon*.
- [XMID1] Mascolo, C., Capra, L., Zachariadis, S., and Emmerich, W., 2002. XMIDDLE: A Data-Sharing Middleware for Mobile Computing. *International Journal on Wireless Personal Communications*, Kluwer Academic Publisher. Vol. 21, No. 1, pp 77-103.

Eine Plattform zur Unterstützung von proaktiven ortsbezogenen Mehrbenutzer-Anwendungen

Johannes Martens, Georg Treu, Peter Ruppel, Diana Weiss, Axel Küpper und
Claudia Linnhoff-Popien

LMU München, Lehrstuhl für Mobile und Verteilte Systeme
Oettingenstr. 67, 80538 München

[johannes.martens|georg.treu|peter.ruppel|diana.weiss|axel.kuepper|linnhoff]@ifi.lmu.de

Zusammenfassung

Die erste Generation der Location-based Services erwies sich nicht als der erhoffte Durchbruch für die Nutzung mobiler Datendienste. Die nächste Generation wird momentan durch die Verbreitung von GPS-Empfängern in Mobiltelefonen und das Auftreten neuer Inhalteanbieter geprägt. Diese neuen Gegebenheiten erfordern eine LBS Middleware, die endgeräte-basierte Ortungsverfahren unterstützt. Dieser Beitrag zeigt die LBS Middleware TraX, die über ein effizientes Position Management und fortgeschrittenen Funktionen zur Erkennung räumlicher Beziehungen zwischen mehreren Zielobjekten verfügt. Dank des generischen und offenen Ansatzes lässt sich TraX auch an verschiedenste Dienstanforderungen und Umgebung anpassen.

1 Einleitung

Ortsbezogene Dienste (Location-based Services, LBSs) berücksichtigen die geographische Position einer oder mehrerer Zielpersonen, um für Nutzer ortsbezogenen Informationen bereitzustellen [KTL06]. LBSs lassen sich anhand folgender Merkmale einteilen:

- **Reaktive und proaktive LBSs:** Während bei reaktiven LBSs dem Nutzer ortsbezogene Informationen nur auf Anfrage ausgeliefert werden, lösen proaktive LBSs beim Eintreten vordefinierter räumlicher Ereignisse, wie etwa dem Annähern einer Zielperson an einen Point of Interest (PoI), Dienstaktionen aus.
- **Selbst- und querverweisende LBSs:** Bei selbstverweisenden LBSs ist der Nutzer mit der Zielperson identisch, d.h. die dem Nutzer präsentierte Information bezieht sich auf dessen eigene Position. Im Gegensatz dazu stehen querverweisende LBSs, bei denen Nutzer bezüglich des Aufenthalts anderer Personen informiert werden, also nicht mit der Zielperson gleichgesetzt sind.
- **Einzelnutzer und Mehrbenutzer LBSs:** Die Einzelnutzer LBSs verbinden die Position eines Nutzers mit statischem geographischen Inhalten, wie zum Beispiel der Darstellung des aktuellen Aufenthaltsort eines Nutzers auf einer Karte. Bei den Mehrbenutzer-LBSs liegt der Fokus hingegen auf der Verknüpfung der Positionen mehrerer Personen, um z.B. zu erkennen, welche anderen Personen sich in der Nähe befinden.
- **Zentralisierte und Peer-to-Peer LBSs:** Zentralisierte LBSs werden von einem zentralen Location oder Application Server angeboten und verwaltet. Die Peer-to-Peer LBSs organisieren sich selbstständig, d.h. die Positionsdaten werden direkt zwischen den Nutzern ohne die Anwesenheit von Intermediären ausgetauscht. Im Folgenden liegt der Fokus auf den zentralisierten Ansätzen.

- **Outdoor und Indoor LBSs:** Outdoor LBSs verfügen über ein großes Abdeckungsgebiet im Freien und verwenden in der Regel Satelliten-gestützte oder zellbasierende Positionierungsverfahren. Im Gegensatz dazu unterstützen die Indoor LBSs den Nutzer innerhalb eines Gebäudes und basieren daher auf lokalen Ortungsverfahren, bei denen meist Nahbereichsfunktechniken zum Einsatz kommen.

Abbildung 1 zeigt eine Übersicht dieser Klassifizierung.

Wurden anfänglich LBSs noch als Killerapplikation gepriesen, stellte sich alsbald heraus, dass die Akzeptanz bei den Nutzern eher moderat war und bis heute ist. Die Ursachen hierfür sind vielfältig. Zum einen beanspruchte die Errichtung von 3G-Netzen zur Jahrtausendwende sämtliche finanziellen und personellen Ressourcen der Netzbetreiber und folglich wurde die Entwicklung von LBSs nur sehr halbherzig und mit wenig Kreativität betrieben. Zum anderen fehlte es zu diesem Zeitpunkt an wichtigen Schlüsseltechnologien, beispielsweise einem effizienten Management von Ortsinformationen, inklusive deren Verteilung und Speicherung in einer verteilten Infrastruktur und Mechanismen zum Schutz der Privatsphäre.

Ein weiterer Grund für die mangelnde Nutzerakzeptanz ist ferner, dass Netzbetreiber und Endgeräte neue Ortungsverfahren nicht unterstützen und stattdessen der Aufenthaltsort eines Nutzers von dem Standort der bedienenden Basisstation abgeleitet wird, was jedoch für viele Anwendungen zu ungenau ist.

Die nächste Generation der LBSs wird gegenwärtig durch die standardmäßige Ausstattung von Mobiltelefonen und PDAs mit GPS-Empfängern und das Auftreten von neuen Inhaltenanbietern, wie z. B. Google oder Yahoo Maps, geprägt. Diese geänderten Rahmenbedingungen erfordern eine Middleware für LBSs, die endgeräte-basierte Ortungsverfahren unterstützt und Funktionalitäten für Mehrbenutzer-LBSs anbietet.

2 TraX Plattform

Einige Ansätze für LBS-Middleware fokussieren sich auf die Auswahl und die Kontrolle der Ortungsverfahren und verschatten diesen Prozess vor der entsprechenden Anwendung. Beispiele hierfür stellen u.a. die Location API der Java 2 Micro Edition (J2ME) [JSR179] oder die PlaceLab Plattform [LCC+05]. Andere Plattformen beschäftigen sich mit der Repräsentation und dem Zugriff auf geographische Inhalte, wie zum Beispiel die vom Open Geospatial Consortium (OGC) spezifizierten Open GIS Location Services (Open LS) oder die Nexus [FrVo03] bzw. die MiddleWhere [RAMC+04] Plattformen, um nur einige zu nennen.

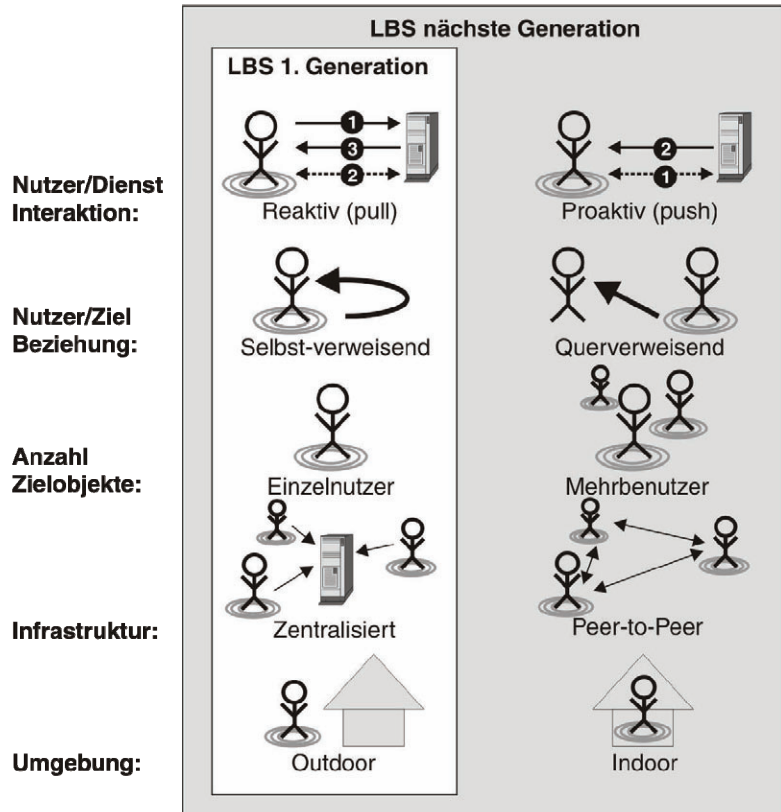


Abbildung 1: Funktionale Klassifikation der LBSs

Die TraX Plattform wurde in [TMK06] erstmalig skizziert und wird am Lehrstuhl weiterentwickelt. Sie unterscheidet sich von den oben genannten Ansätzen dadurch, dass sie auf alle Akteure der LBS-Wertschöpfungskette verteilt ist. Zusätzlich ist sie beliebig erweiterbar und dadurch sehr flexibel an neue Dienstanforderungen und Rekonfigurationen der LBS-Wertschöpfungskette anpassbar. Diese Erweiterbarkeit ermöglicht auch das Hinzufügen von zukünftigen oder proprietären Ortungsverfahren. Darüber hinaus integriert die TraX-Plattform auch bestehende Frameworks und Standards, wie z.B. die J2ME Location API, das Mobile Location Protocol (MLP) der Open Mobile Alliance (OMA) und die OpenLS. Abbildung 2 zeigt die verschiedenen Schichten, welche die TraX-Plattform bilden. Diese werden im Folgenden näher erläutert.

Der **Positioning Layer** realisiert die Ortung auf dem mobilen Endgerät. Hierfür implementiert es verschiedene endgeräte-basierte Ortungsverfahren, wie zum Beispiel GPS, Enhanced Observed Time Difference (E-OTD) oder WLAN Fingerprinting (siehe [Kuep05] für einen Überblick über Ortungsverfahren). Ziel dieser Schicht ist es, die Heterogenität der unterschiedlichen Ortungstechniken vor den folgenden Schichten zu verschatten. Daher erhält der Positioning Layer Ortungsanfragen und abstrakte Anforderungen, wie z.B. benötigte Genauigkeit oder Datenformat der Positionsangaben, wählt darauf hin das adäquate Ortungsverfahren aus und transformiert gegebenenfalls die ermittelte Position in das gewünschte Datenformat. Potentielle Kandidaten zur Realisierung dieser Schicht umfassen die J2ME Location API oder auch die PlaceLab Plattform.

Im Gegensatz zur vorherigen Schicht ist der **Position Management Layer** auf das mobile Endgerät und die zentrale Komponente (Location Server) verteilt. Ziel dieser Schicht ist es eine effiziente Überwachung der Position des mobilen Endgeräts über die Luftschnittstelle zu ermöglichen. Neben der klassischen synchronen Anfrage nach sofortiger Positionsübermittlung (Polling) bietet das Position Management auch asynchrone Anfragestrategien. Bei letzteren wird das mobile Endgerät über eine Nachricht vom Location Server so konfiguriert, dass es gemäß der ausgewählten Strategie periodisch, nach dem Zurücklegen einer bestimmten Distanz oder auch beim Betreten oder Verlassen einer bestimmten geographischen Zone die aktuelle Position in Form eines Position Updates (PU) asynchron an den Location Server übermittelt. Diese Aktualisierungsstrategien sind besonders für proaktive LBSs erforderlich.

Der **Location Services Layer** ist auf dem Location Server realisiert und bietet eine Menge von Diensten, die auf der Funktionalität des Position Managements basieren. Diese lassen sich entsprechend der LBS-Klassifikation in zwei Gruppen aufteilen: Einzelnutzer LBSs verwenden das Standard Location Reporting (reaktive Dienstnutzung) bzw. das Triggered Location Reporting (proaktive Dienstnutzung). Die Mehrbenutzer LBSs sind hingegen auf die räumlichen Beziehungen zwischen den Zielobjekten fokussiert. Daher

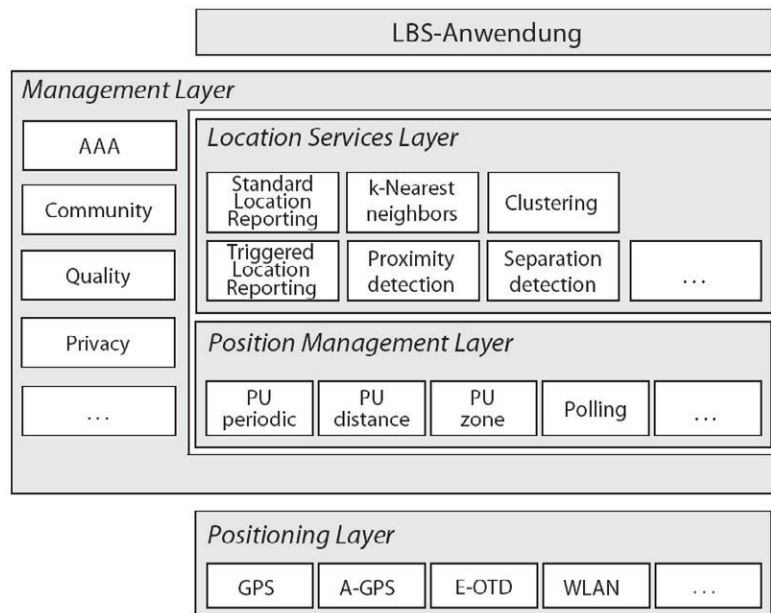


Abbildung 2: Architektur der TraX-Plattform

benötigen sie Methoden zur Erkennung von räumlichen Beziehungen zwischen den Objekten, wie z.B. Annäherung unter eine bestimmte Distanz oder auch die k-nächsten Objekte. Die proaktiven Varianten dieser LBSs registrieren sich für die Ereignisse in diesen Beziehungen, wie z.B. die Veränderung der Reihenfolge bzw. der Mitglieder der k-nächsten Objekte.

Der **Management Layer** umfasst die Methoden zur Verwaltung der Nutzerdaten, dem Accounting und der Einhaltung des Datenschutzes sowie der Privatsphäre. Neben dem Einsatz von Nutzerpseudonymen wird die Plattform in Zukunft dem Nutzer auch das Spezifizieren von Datenschutzrichtlinien ermöglichen. Des Weiteren soll ein Ansatz zur Anonymisierung von Zielpersonen, welcher sich distanzerhaltender Koordinatentransformationen bedient [RTKL06], in die TraX Plattform integriert werden. Weitere Aspekte zur Wahrung der Privatsphäre in der vorgestellten Middleware finden sich in [KTL06].

3 Fazit und Ausblick

In dieser Arbeit wurde die TraX Plattform vorgestellt, mit der es möglich ist, ortsbezogene Mehrbenutzer Dienste einfach, kostengünstig und effektiv hinsichtlich des Schutzes sensibler Daten von Zielpersonen anzubieten. TraX übernimmt dabei für den Dienstbetreiber die Aufgabe, Zielpersonen zu verfolgen sowie aus den gewonnenen Ortsinformationen räumliche Konstellationen von Zielpersonen abzuleiten. In Analogie zu den OpenLS, die u.a. einheitliche Funktionen zur Nutzung von Geo-Informationssystemen (GIS) bieten und damit kartenbasierte LBSs unterstützen, bietet TraX also Abstraktionsmechanismen vor den relativ komplexen Verfahren, die ortsbezogenen Mehrbenutzer Diensten zugrunde liegen.

TraX wird am Lehrstuhl für Mobile und Verteilte Systeme der LMU München entwickelt und soll in Zukunft mit Funktionen zur Verwaltung von sozialen Netzen/Communities und von mobilen virtuellen Objekten erweitert werden, wie sie u.a. im Bereich des Mobile Gaming erforderlich sind. Aktuelle Informationen zur TraX Plattform können von der TraX-Webseite [TraX06] abgerufen werden.

Literatur

- [FrVo03] Fritsch, D. und S. Volz. *Nexus – the Mobile GIS environment*, In Proceedings of the first Joint Workshop on mobile future and symposium on trends in communications, Bratislava, Slovakia, Oct. 2003, pp. 26 - 28
- [Kuep05] Küpper, A. *Location-based Services - Fundamentals and Operation*, NewYork: Wiley, Sept. 2005
- [KTL06] Küpper, A., G. Treu, C. Linnhoff-Popien, *TraX: A Device-centric Middleware Framework for Location-based Services*, IEEE Communications Magazine, Sept. 2006.
- [LMC+05] LaMarca, A., Chawathe Y., Consolvo S., Hightower J., Smith I., Scott J., Sohn T., Howard J., Hughes J., Potter F., Tabert J., Powledge P., Borriello, G. and Schilit, B. *Place Lab: Device Positioning Using Radio Beacons in the Wild*, In Proceedings of the International Conference Pervasive Computing, Munich, Springer-Verlag, LNCS 3468, pp. 116-133, Mai 2005.
- [JSR179] Community Process, JSR-179 Expert Group, Location API for Java 2 Micro Edition, 2003
- [RAMC+04] Ranganathan, A., J. Al-Muhtadi, S. Chetan, R. Campbell und D. Mickunas: *MiddleWhere: a middleware for location awareness in ubiquitous computing applications*. In Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware, pp. 397 - 416, New York, Springer-Verlag, October 2004.
- [RTKL06] Ruppel, P., G. Treu, A. Küpper, C. Linnhoff-Popien, *Anonymous User Tracking for Location-Based Community Services* In Proc. 2nd International Workshop on Location- and Context-Awareness (LoCA), LNCS 3987, pp. 116-133, Springer-Verlag, Dublin, Ireland, Mai 2006.
- [TMK06] Treu, G., J. Martens, A. Küpper, *TraX - Eine Plattform zur Unterstützung ortsbezogener Community Dienste*, Praxis der Informationsverarbeitung und Kommunikation (PIK), vol. 29 (1/06), pp. 18-23, München, Saur Verlag, März 2006.
- [TraX06] TraX Webseite, <http://www.mobile.ifi.lmu.de/TraX/>