Birgitta König-Ries, Michael Klein (Hrsg.)

Mobile Datenbanken: heute, morgen und in 20 Jahren

8. Workshop des GI-Arbeitskreises "Mobile Datenbanken und Informationssysteme"
28.02. – 01.03.2005

im Rahmen der BTW 2005 in Karlsruhe

Gesellschaft für Informatik 2005

Vorwort

Der Workshop „Mobile Datenbanken – heute, morgen und in 20 Jahren“ ist der nunmehr achte Workshop des GI Arbeitskreises „Mobile Datenbanken und Informationssysteme“. Der Workshop findet im Rahmen der BTW 2005, der GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web, vom 28. Februar bis zum 01. März 2005 in Karlsruhe statt.

Das Workshopprogramm umfasst zwei eingeladene Vorträge sowie sieben wissenschaftliche Beiträge, die vom Programmkomitee aus den Einreichungen ausgewählt wurden. Für den zweiten Workshoptag, der im Zeichen intensiver Diskussionen stehen soll, wurden zwei weitere Einreichungen als Diskussionsgrundlage ausgewählt.

Inhaltlich spannt der Workshop einen weiten Bogen: Von fast schon klassischen Fragen aus dem Kernbereich mobiler Datenbanken, wie etwa der Transaktionsbearbeitung in diesen Systemen, bis hin zu neuen Multimediaanwendungen auf mobilen Geräten und von der Anfragebearbeitung in Ad-hoc-Netzen bis zur Analyse des Stands der Technik beim Entwurf mobiler Anwendungen. Diese Breite spiegelt die Breite der Fragestellungen, die bei der Betrachtung von mobiler Informationsnutzung zu Tage treten, wider. Wir hoffen mit unserem Workshop einen Beitrag zum besseren Verständnis dieser Fragestellungen zu liefern und ein Forum zum Austausch von Fragen, Lösungsansätzen und Problemstellungen zwischen Praktikern und Forschern aus dem universitären Umfeld zu bieten.

Der Workshop wäre nicht möglich ohne die Unterstützung vieler Personen: Hier seien an erster Stelle die Mitglieder des Programmkomitees genannt, die den Autoren wichtige und umfangreiche Hinweise zur Verbesserung der Arbeiten gaben. Besonderer Dank gebührt auch den Organisatoren der BTW, die zum einen die Durchführung des Workshops im Rahmen der BTW ermöglicht haben, und die zum anderen den gesamten Anmeldeprozess und die lokale Organisation übernommen haben. Dank auch an Michael Klein für die Fertigstellung dieser Proceedings.

Ich wünsche uns allen einen interessanten, neue Impulse gebenden Workshop!

Jena, im Februar 2005

Birgitta König-Ries

Organisation

Programmkomitee:

- Christian Becker, Uni Stuttgart
- Susanne Boll, Uni Oldenburg
- Martin Breunig, Uni Vechta
- Thomas Fanghänel, IBM San Jose
- Christoph Gollmick, SAP AG
- Hagen Höpfner, Uni Magdeburg
- Helmut Krcmar, TU München
- Birgitta König-Ries, Uni Jena
- Franz Lehner, Uni Passau
- Rainer Malaka, EML Heidelberg
- Pedro José Marrón, Uni Stuttgart
- Holger Meyer, Uni Rostock
- Daniela Nicklas, Uni Stuttgart
- Marco Plack, Metop GmbH, Magdeburg
- Key Pousttchi, Uni Augsburg
- Kai-Uwe Sattler, TU Ilmenau
- Heiko Schuldt, UMIT Innsbruck
- Heinz Schweppe, FU Berlin
- Günther Specht, Uni Ulm
- Wolffried Stucky, Uni Karlsruhe
- Can Türker, ETH Zürich
- Klaus Turowski, Uni Augsburg

externer Gutachter:

- Jürgen Vogel, EML GmbH

Inhalt

Eingeladene Vorträge

Wissenschaftliche Beiträge

Diskussionspapiere

Transaction Processing On Mobile Devices -- Real-Life Challenges And Solutions

Thomas Fanghänel, IBM San Jose
fanghaen@us.ibm.com

The architectural evolution of mobile computing platforms during the past few years shows a convergence towards a common memory hierarchy. The most popular devices currently in the market are all built around relatively simple processors equipped with small caches, and have a fair amount of persistent main memory as well as some standardized external storage interface. The vast majority of the external storage media is flash-based, with capacities that exceed the size of the main memory by more than an order of magnitude.

Unlike in traditional database systems, databases on mobile devices may be stored in either the battery-backed internal memory or on an external storage media, i. e. on different levels of the memory hierarchy. For the design of a mobile database management system, such as IBM DB2 Everyplace, one needs to pay particular attention to this characteristic trait.

The talk gives an overview of the different natures of main memory and storage cards, with respect to both data consistency and performance. Based on this, a number of problems that impose challenges for transaction processing in such an environment are identified and discussed. Eventually, the techniques are outlined, which DB2 Everyplace implements to overcome these problems, and achieve both high performance and a maximum level of data consistency and durability.

Nutzerzentrierte Mobile Multimedia-Anwendungen

Susanne Boll, Universität Oldenburg
Susanne.Boll@informatik.uni-oldenburg.de

Mobile Anwendungen und Systeme werden immer mehr Teil unseres privaten und beruflichen täglichen Lebens. Die Mobilität selbst, die Bedürfnisse des mobilen Menschen als auch die heterogene Welt von Geräten, Plattformen und Netzen stellen dabei besondere Herausforderungen dar. Vor diesem Hintergrund stellt der Vortrag aktuelle wissenschaftliche Arbeiten zur Unterstützung der Entwicklung nutzerzentrierter mobiler Multimedia-Anwendungen und -Systeme vor: Ausgehend von der Unterstützung der Entwicklung mobiler Anwendungen durch eine modulare Systemplattform, präsentiert der Vortrag Arbeiten im Bereich der Personalisierung multimedialer Inhalte für mobile Anwendungen, wie etwa einer mobilen Stadtführung, die an sich dynamisch an die individuellen Nutzerinteressen sowie die Eigenschaften des jeweiligen Endgerätes anpasst. Die Einbeziehung des auditiven Wahrnehmungssystems in die Gestaltung der Nutzungsschnittstelle der mobilen Anwendung erlaubt es, den Nutzer, die Nutzerin über eine weitere Modalität in der mobilen Aufgabe wie etwa bei der Erkundung einer Stadt zu unterstützen. Die Modellierung und Verwendung von Kontextinformation wie etwa Ort, Zeit oder aktuelle Rolle für die kontextsensitive Auswahl, Aufbereitung und Visualisierung von Informationen adressieren den aktuellen Nutzungskontext und tragen so zur Nutzerzentrierung mobiler Anwendungen bei. Der Vortrag adressiert dabei die Anforderungen an mobiles Informationsmanagement aus der Sicht nutzerzentrierter mobiler Multimedia-Anwendung.

Kontextsensitives mobiles Marketing

Rebecca Bulander[1], Michael Decker[1], Bernhard Kölmel[2], Gunther Schiefer[1]

[1]Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB)
Universität Karlsruhe (TH)
76128 Karlsruhe
{bulander, decker, schiefer}@aifb.uni-karlsruhe.de

[2]CAS Software AG
Wilhelm-Schickard-Str. 10-12
76131 Karlsruhe
bernhard.koelmel@cas.de

Abstract: Mobile Endgeräte wie Mobiltelefone oder PDAs stellen eine viel versprechende Zielplattform für Marketing-Aktionen dar: Die Geräte sind weit verbreitet, ermöglichen die Darstellung personalisierter und interaktiver multimedialer Inhalte und bieten als fast ständig mitgeführte persönliche Kommunikationsmittel eine hohe Erreichbarkeit des Benutzers. Aber gerade wegen der zuletzt genannten Eigenschaft ist bei der Entwicklung eines Systems für mobiles Marketing ein besonderes Augenmerk auf Datenschutzaspekte und Spamvermeidung zu richten. Darüber hinaus stellt die sehr eingeschränkte Benutzerschnittstelle mobiler Endgeräte eine besondere Herausforderung an die Usability dar. Der folgende Beitrag beschreibt den hierfür im Rahmen des BMWA-geförderten Projektes „Mobiles Marketing“ (MoMa) entwickelten Lösungsansatz.

1 Einführung

Unter Marketing im engeren Sinne versteht man die Gesamtheit aller verkaufsfördernden Maßnahmen einer Unternehmung. Neben Distributions-, Preis- und Produktgestaltung ist hier vor allem die Kommunikationspolitik – etwa in Form von Werbung – zu nennen. Für zielgerichtetes mobiles Marketing kommen unter Einbeziehung von Kontext (z.B. Profile, aktueller Standort) mobile Endgeräte wie z.B. Smartphones als Zielplattform zum Einsatz.

Mobile Endgeräte als Medium sind für das Marketing aus folgenden Gründen interessant:

- Sie sind sehr verbreitet: Alleine in Deutschland gibt es mit mehr als 64 Millionen Geräten bereits mehr Mobiltelefone als Festnetzanschlüsse [Re04]. Die weltweite Zahl der Mobilgeräte wird derzeit mit rund 1,5 Milliarden angegeben [Co04].
- Sie werden als persönliche Kommunikationsgeräte fast überall mitgetragen und ermöglichen somit eine hohe Erreichbarkeit. Mobiltelefone sind im Durchschnitt 14 Stunden pro Tag eingeschaltet [So04].

- Da jedes Endgerät einzeln angesprochen werden kann, ist eine zielgerichtete Kommunikation möglich; dies ist eine wichtige Grundvoraussetzung für die Verminderung von Streuverlusten und Personalisierung der Inhalte.
- Sie ermöglichen Interaktion und je nach Ausstattung auch die Darstellung von multimedialen Inhalten.
- Die gerade aufkommenden Mobilfunknetze der dritten Generation (vor allem UMTS) ermöglichen mit ihren enormen Bandbreiten bisher undenkbare mobile Datendienste auch im Bereich des Marketings.

Diesen Vorzügen mobiler Endgeräte als Marketing-Plattform stehen einige Nachteile gegenüber:

- Spam: Aufgrund des ständig wachsenden Spam-Aufkommens im E-Mail-Verkehr – es gibt Statistiken, die für den Spam-Anteil am E-Mail-Verkehr einen Wert von weit über 50 % nennen (etwa [hei04]) – besteht die Befürchtung, dass diese Spam-Welle auch auf mobile Netzwerke überschwappt. Spam-Nachrichten in mobilen Netzwerken wären noch weitaus problematischer, da mobile Endgeräte nur über sehr eingeschränkte Ressourcen (Speicherplatz für Nachrichten, Bandbreite und Akkulaufzeit) verfügen.
- Privacy: Ein Nutzer wird nur dann bereit sein, persönliche Daten (Alter, Familienstand, Interessengebiete, usw.) für eine mobile Marketingapplikation bereitzustellen, wenn gewährleistet ist, dass die einschlägigen Datenschutzbestimmungen streng eingehalten werden und es ihm einen Mehrwert bringt.
- Usability: Aufgrund ihrer geringen Größe verfügen mobile Endgeräte nur über eine sehr eingeschränkte Benutzerschnittstelle, z.B. keine Tastatur und kleine Displays. Dem Endanwender sollten deshalb so wenig Eingaben wie möglich abverlangt werden, und darzustellende Inhalte müssen speziell für die einzelnen Typen von Endgeräten aufbereitet werden.

Im Rahmen des MobilMedia-Leitprojektes „Mobiles Marketing (MoMa)“ wird deshalb ein System entwickelt, dessen Design und Geschäftsmodell darauf ausgelegt sind, mobiles Marketing unter der Berücksichtigung dieser Problembereiche zu ermöglichen. Derzeit wird der weitgehend fertig gestellte Labor-Demonstrator im Rahmen einer Usability-Studie evaluiert.

Der vorliegende Beitrag ist wie folgt aufgebaut: Im zweiten Kapitel führen wir für das weitere Verständnis benötigte Definitionen ein und zeigen die für das kontextsensitive mobile Marketing rechtlichen Bestimmungen auf; im dritten Kapitel werden die Funktionsweise, das Geschäftsmodell und die Architektur des MoMa-Systems eingehend beschrieben. Anschließend wird in Kapitel vier auf die verschiedenen Kontext-Arten des MoMa-Systems genauer eingegangen; Kapitel fünf enthält die Schlussbetrachtung.

2 Grundlagen

2.1 Kontext

Im Zusammenhang mit mobilen Anwendungen verstehen wir unter Kontext eine Menge an Informationen, welche die aktuelle Situation eines Nutzers beschreiben [SAW94]. Eine kontextsensitive Anwendung nutzt diese Informationen zur Anpassung an die sich daraus ergebenden Bedürfnisse des Nutzers. Dies ist bei mobilen Applikationen besonders wichtig, da hier die Endgeräte aufgrund ihrer Größe und Portabilität über eine sehr eingeschränkte Benutzerschnittstelle (z.B. keine vollständige oder überhaupt keine Tastatur, kleines Display, usw.) verfügen und dem Nutzer somit möglichst Eingaben abgenommen werden sollten.

Das am häufigsten zitierte Beispiel für kontextsensitive Dienste sind die sog. Location Based Services (LBS): In Abhängigkeit seines aktuellen Aufenthaltsortes wird der Nutzer mit entsprechend angepassten Informationen versorgt, z.B. in Form eines Touristenführers, der Erläuterungen über die gerade in der näheren Umgebung befindlichen Sehenswürdigkeiten präsentiert (z.B. [Ch00]). Technisch kann dieser Ortskontext beispielsweise über einen GPS-Empfänger oder die benutzte Funkzelle bestimmt werden.

Andere denkbare Umsetzungen kontextsensitiver Dienste greifen auf Profilinformationen zurück. Diese Profile können unter expliziter Mitwirkung des Nutzers gewonnen werden (aktive Profilierung: z.B. Abfrage von soziodemographischen Daten, Interessengebieten und Nutzungsgewohnheiten) oder durch Auswertung von Nutzersitzungen (passive Profilierung: z.B. Anwendung von Data-Mining-Techniken). Aktive Profilierung bedeutet Mehrarbeit für den Nutzer, ist aber für ihn vollkommen transparent; zudem können nicht alle relevanten Informationen wie z.B. das Alter eines Nutzers ohne weiteres mittels passiver Profilierung erhoben werden.

2.2 Datenschutzbestimmungen

Für kontextsensitives mobiles Marketing in Deutschland sind vor allem das Bundesdatenschutzgesetz (BDSG) und das Gesetz über den Datenschutz bei Telediensten (TDDSG) maßgebend. Die Datenschutzbestimmungen beziehen sich auf die personenbezogen Daten, welche im BDSG definiert werden als „Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person“. In § 3 Abs. 6 des BDSG wird Anonymität wie folgt definiert: „Anonymisieren ist das Verändern personenbezogener Daten derart, dass die Einzelangaben über persönliche oder sachliche Verhältnisse nicht mehr oder nur mit einem unverhältnismäßig großen Aufwand an Zeit, Kosten und Arbeitskraft einer bestimmten oder bestimmbaren natürlichen Person zugeordnet werden können." Bei Pseudonymen hingegen existiert eine Abbildung (Korrespondenz-Regel), die das Pseudonym einer einzelnen Person zuordnet (BDSG § 3 Abs. 6a): „Pseudonymisieren ist das Ersetzen des Namens und anderer Identifikationsmerkmale durch ein Kennzeichen zu dem Zweck, die Bestimmung des Betroffenen auszuschließen oder wesentlich zu erschweren.“

Das BDSG hat den Zweck, den Anwender davor zu schützen, dass er durch den Umgang mit seinen eigenen personenbezogenen Daten in seinem Persönlichkeitsrecht beeinträchtigt wird. § 3a zur Datenvermeidung und Datensparsamkeit schreibt vor, dass bei der Gestaltung und Auswahl von Datenverarbeitungssystemen von den Möglichkeiten der Anonymisierung und Pseudonymisierung Gebrauch zu machen ist. § 30 regelt die geschäftsmäßige Datenerhebung und -speicherung zum Zweck der Übermittlung in anonymisierter Form. Darin wird vorgeschrieben, dass die Merkmale gesondert zu speichern sind, mit denen Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person zugeordnet werden können. Hier wird die für Werbung und Marktforschung wichtige Möglichkeit der Anonymisierung und Pseudonymisierung nochmals ausführlich geregelt. Die strengen Regelungen des § 29 BDSG für die nicht anonymisierte Datenerhebung und -speicherung zum Zweck der Übermittlung werden dabei explizit ausgeschlossen [Du03].

Das TDDSG dient dem Schutz personenbezogener Daten der Nutzer von Telediensten im Sinne des Teledienstegesetzes (TDG) bei der Erhebung, Verarbeitung und Nutzung dieser Daten durch Diensteanbieter. Sofern kein Widerspruch des Nutzers vorliegt, darf der Diensteanbieter gemäß § 6 Abs. 3 TDDSG für Zwecke der Werbung, der Marktforschung oder zur bedarfsgerechten Gestaltung der Teledienste Nutzungsprofile erstellen, wenn hierbei Pseudonyme verwendet werden. Die Nutzungsprofile dürfen nicht mit Daten über den Träger des Pseudonyms zusammengeführt werden. Der Nutzer muss in jedem Fall über sein Widerspruchsrecht informiert werden, und zwar im Rahmen der obligatorischen Unterrichtung nach § 4 Abs. 1 TDDSG.

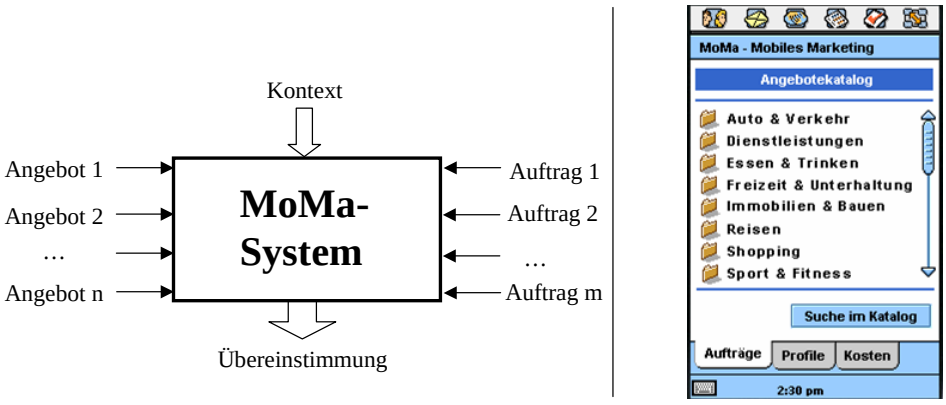
3 Beschreibung des MoMa-Systems

3.1 Überblick

Das grundlegende Funktionsprinzip des MoMa-Systems wird in Abbildung 1a dargestellt: Die Endnutzer formulieren anhand eines vom MoMa-Betreiber vorgegebenen Katalogs sog. *Aufträge* für Angebote (siehe Abbildung 1b), wobei hier auf Kontextparameter zurückgegriffen werden kann. Der Katalog besteht aus einer Hierarchie von Dienstleistungs- und Produktangeboten (z.B. auf oberster Ebene „Essen & Trinken“, weiter unterteilt in „Cafés/Kneipen/Biergärten“, „Gaststätten/Restaurants“ und „Lieferservice/Catering“), welche durch vorgegebene Attribute (im Beispiel des Restaurant z.B. „Preislage“ und „Ambiente“) genauer spezifiziert werden können. Bei der Erstellung von Aufträgen wird dabei automatisch auf geeignete Kontextparameter zurückgegriffen. Bei der Suche nach einer gastronomischen Einrichtung werden zum Beispiel Ort und Wetter berücksichtigt: Die Lokalität sollte sich in der Nähe des aktuellen Aufenthaltsortes befinden und es sollten je nach Witterungslage bestimmte Lokalitäten nicht berücksichtigt werden (z.B. keine Freiluftgastronomie bei Regen).

Auf der anderen Seite stellen die Werbetreibenden ebenfalls anhand des Katalogs formulierte *Angebote* in das System. Das System sucht nach zueinander passenden Paaren von Aufträgen und Angeboten. Im Falle einer „Übereinstimmung“ erhält der Auftraggeber

eine Benachrichtigung vom MoMa-System. Er kann dann entscheiden, ob er zur Wahrnehmung des Angebotes den Werbetreibenden kontaktieren möchte. Die eigentliche Abwicklung einer so angebahnten Geschäftsbeziehung ist nicht mehr Gegenstand des MoMa-Systems.



(a) (b)

Abbildung 1: Grundprinzip des MoMa-System (a) und Screenshot der Client-Anwendung für Symbian OS (b)

Der Endnutzer erhält nur dann Angebote, wenn er diese ausdrücklich wünscht und diese seinen zuvor festgelegten Vorgaben entsprechen. Sollte ein Anbieter Angebote unter der falschen Kategorie einstellen, so hat der Endnutzer die Möglichkeit, über einen Feedback-Loop eine Beschwerde mit der Angebots-ID an den MoMa-Systembetreiber zu richten. Erst wenn sich der Endnutzer zur Kontaktaufnahme mit dem Werbetreibenden entscheidet, verliert er seine Anonymität gegenüber diesem. Durch Einsatz eines zwischen Endnutzer und MoMa-System geschalteten Anonymisierungsdienstes, der von einer vertrauenswürdigen Drittpartei betrieben wird, kann sogar Transaktionspseudonymität[1] gegenüber dem MoMa-Betreiber gewährleistet werden.

3.2 Geschäftsmodell

Die Geld- und Informationsflüsse zwischen den einzelnen Rollen des MoMa-Geschäftsmodells sind in Abbildung 2 dargestellt. Das Geschäftsmodell weist dabei folgende sechs Rollen auf: das werbetreibende Unternehmen, den MoMa-Systembetreiber, den Kontext-Provider, den Mobilfunk-Provider, die vertrauenswürdige Partei und den Endnutzer.

Für den Endnutzer entstehen bis auf die Verbindungsentgelte seines Mobilfunk-Providers keine Entgelte durch die Benutzung des MoMa-Systems. Dafür hat der Werbetreibende für die tatsächlich erfolgten Kontakte in Abhängigkeit der jeweiligen Produkt-

[1] Transaktionspseudonyme werden nur für eine einzelne, evtl. aus mehreren Operationen bestehende, geschäftliche Transaktion verwendet und repräsentieren damit die höchste Stufe der Pseudonymität [PK00].

oder Dienstleistungskategorie eine Zahlung zu leisten, unabhängig davon, ob dieser zu einem Geschäft geführt hat, da dies außerhalb der Kontrolle des MoMa-Betreibers liegt.

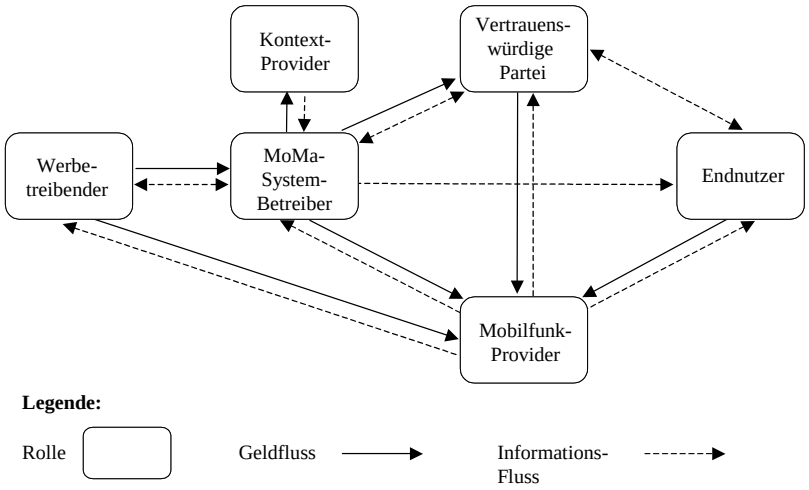


Abbildung 2: MoMa-Geschäftsmodell

Die einzelnen Kategorien des Katalogs werden unterschiedlich bepreist, wodurch auch das Werbeaufkommen gesteuert werden kann. Für ein Angebot aus dem Bereich der Gastronomie (Mittagstisch, Happy Hour, usw.) wird der Kontaktpreis im Bereich von einem Cent oder Bruchteilen davon liegen, zzgl. der Verbindungskosten für die Benachrichtigung des Endnutzers. Werbekontakte für hochwertigere Güter wie Immobilienangebote oder Urlaubsreisen sowie Kategorien, welche andere subsumieren, können höher bepreist werden. Zusätzlich kann der MoMa-Betreiber anonymisierte statistische Auswertungen des Nachfrageverhaltens der MoMa-Nutzer an die Werbetreibenden verkaufen.

Die Bereitstellung der Kontextinformationen und der Anonymisierungsdienst werden vom MoMa-Betreiber vergütet.

3.3 Architektur und technische Details

Jeder Endnutzer des MoMa-Systems (siehe Abbildung 3) hat eine eindeutige Nutzer-ID sowie jeweils mindestens ein allgemeines Nutzer- und ein Benachrichtigungsprofil. Im Nutzerprofil sind Informationen über den Nutzer hinterlegt, die für die Formulierung eines Auftrages nötigt sein könnten, z.B. Alter, Familienstand, Interessengebiete; der Nutzer muss einem konkreten Auftrag nur die Angaben beifügen, welche speziell dafür benötigt werden und die er auch tatsächlich machen möchte. Ein Benachrichtigungsprofil beschreibt, wie ein Nutzer über mögliche Treffer in Kenntnis gesetzt werden möchte, z.B. SMS/MMS, E-Mail oder Text-to-Speech-Nachricht an bestimmte Endadressen. In Abhängigkeit der Uhrzeit, des Wochentags usw. können auch alternative Benachrichtigungswege festgelegt werden, z.B. keine Text-to-Speech-Anrufe von 20 bis 8 Uhr, sondern statt dessen eine E-Mail-Nachricht. Sowohl die Benachrichtigungs- als auch die all-

gemeinen Nutzerprofile werden auf einem Server des Anonymisierungsdienstes abgelegt, um den Abgleich auf verschiedene Endgeräte des Nutzers zu ermöglichen. Hierbei müssen nur die Benachrichtigungsprofile für den Anonymisierungsdienst lesbar sein, die allgemeinen Profildaten können so verschlüsselt sein, dass nur der Nutzer sie lesen kann.

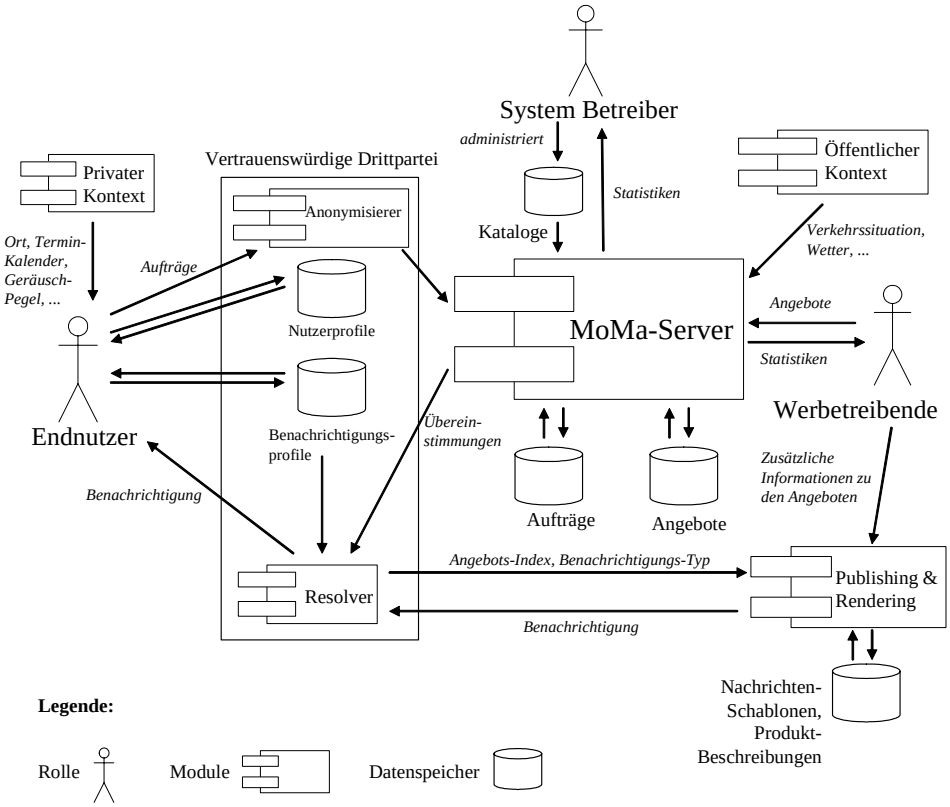


Abbildung 3: Architektur des MoMa-Systems

Zur Erzeugung eines Auftrages X wählt der Endnutzer ein Nutzer- und Benachrichtigungsprofil aus und spezifiziert seinen Auftragswunsch anhand der Kategorien und jeweiligen Attribute des Katalogs. Hierbei werden ggf. Attribute aus dem Nutzerprofil und private Kontextparameter automatisch übernommen. Der Auftrag X selbst enthält keinerlei Angaben über die Identität oder Endadressen des Nutzers. Weiter werden die ID des Nutzers, der Index seines gewünschten Benachrichtigungsprofils sowie ein zufällig erzeugter Bitstring gemeinsam verschlüsselt; der so gewonnene Chiffretext C ist vom Anonymisierungsdienst entschlüsselbar, nicht aber vom Betreiber des MoMa-Systems oder gar einem Werbetreibenden[2]. Das Paar {X, C} wird nun über den Anony-

[2] Wird ein symmetrisches Verschlüsselungsverfahren herangezogen, so kann der Endnutzer selbst wieder C entschlüsseln, bei asymmetrischen Verschlüsselungsverfahren nicht. Symmetrische Verschlüsselungsverfahren sind rechentechnisch deutlich weniger aufwendig, setzen allerdings einen sicheren Kanal zur Übertragung des Schlüssels voraus.

misierungsdienst an den MoMa-Server übermittelt; der Umweg gewährleistet, dass der MoMa-Betreiber nicht über die MSISDN- oder IP-Adresse Rückschlüsse auf die Identität des Auftragserstellers ziehen kann. Selbst wenn der Nutzer mehrere identische Aufträge (gleiches X) mit dem selben Benachrichtigungsprofil erstellt, entsteht durch die Verwendung der Zufallszeichenkette ein abweichender Chiffretext. C ist also ein Transaktionspseudonym und gewährleistet hiermit die höchste Form der Pseudonymität. Sollte sich ein bei der Spezifikation eines Auftrages verwendeter Kontextparameter ändern (z.B. Aufenthaltsort), so wird der entsprechend aktualisierte Auftrag X' mitsamt C erneut über den Anonymisierungsdienst an den MoMa-Server übertragen, wo der Auftrag mit der Chiffre C gesucht wird und X durch X' ersetzt wird.

Der Werbetreibende definiert sein Angebot Y ebenfalls anhand des Kataloges und übermittelt es direkt an den MoMa-Server. Darüber hinaus hinterlegt er auf den Publishing- & Rendering-Server passende Schablonen zur Benachrichtigung der Endkunden im Falle eines Treffers; es können auch zusätzliche Informationen über das Angebot oder das Gewerbe des Anbieters abgelegt werden.

Initiiert durch Systemereignisse wie neue bzw. geänderte Angebote und Aufträge oder Änderungen relevanter öffentlicher Kontextparameter versucht der MoMa-Server zueinander passende Kombinationen von Angebot X und Auftrag Y zu finden. Für jedes so gefundene passende Paar {{X, C}, Y} wird C zusammen mit der ID des Angebotes Y an die Resolver-Komponente des Anonymisierungsdienstes weitergeleitet. Hier wird C entschlüsselt, so dass zuerst die gewünschte Benachrichtigungsform festgestellt werden kann, um beim Publishing-Server die entsprechende Nachricht anzufordern. Die dem Empfangsgerät angemessen aufbereitete Nachricht wird dann an die entsprechende Empfangsadresse weitergeleitet.

4 Klassifikation von Kontextinformationen

Bedingt durch die Anonymisierung der Nutzeraufträge im MoMa-System ist die Unterscheidung von privatem und öffentlichem Kontext (c_{i1}, c_{i2} in Tabelle 1) notwendig:

- **Privater Kontext:** An der Erhebung von Parametern des privaten Kontextes ist das mobile Endgerät samt Peripherie wie etwa Sensoren des jeweiligen Nutzers zumindest beteiligt oder alleine dafür verantwortlich. Der private Kontext kann deshalb nicht anonymisiert erhoben werden; er kann jedoch anonymisiert weiterverarbeitet werden.
  Beispiele: Aufenthaltsort, Hintergrundgeräuschpegel, Umgebungstemperatur, Körpertemperatur, Terminkalender, zur Verfügung stehende technische Ressourcen wie Displaygröße oder Speicherplatz
- **Öffentlicher Kontext:** Diese Kontextinformationen sind auch ohne Kenntnis der Identität des jeweiligen Nutzers und damit anonymisiert ohne direkte Mithilfe seines Endgerätes erhebbar.
  Beispiele: Wetter, Verkehrslage, Börsenkurse

Für die sinnvolle Verarbeitung von einigen öffentlichen Kontextparametern können private Kontextparameter benötigt werden. So ist z.B. die aktuelle Wettersituation in

einer bestimmten Stadt ein öffentlicher Kontextparameter; die Stadt in der sich ein Nutzer gerade aufhält, ist ein privater Kontextparameter.

Zusätzlich kann noch eine Einteilung anhand des Variabilitätsgrades in statischen, semistatischen und dynamischen Kontext (c_{1j}, c_{2j}, c_{3j} in Tabelle 1) vorgenommen werden:

- **Statische** Kontextparameter ändern sich nicht bzw. äußerst selten, z.B. Geschlecht oder Muttersprache.
- **Semistatische** Kontextparameter ändern sich, allerdings nur in größeren Zeitabständen (mehrere Wochen bis Jahre). Beispiele sind das Alter oder der Familienstand.
- **Dynamische** Kontextparameter können sich häufig ändern, beispielsweise der Aufenthaltsort eines Nutzers.

In Tabelle 1 werden diese beiden Klassifikationsmöglichkeiten von Kontextparametern in einer Matrix übereinander gelegt. Dabei sind in Tabelle 1 für jede der so entstehenden Kategorien Beispiele angegeben.

Kontextdimensionen c_{ij} (Beispiele)	**Öffentlich c_{i1}**	**Privat c_{i2}**
Statisch c_{1j}	**c_{11}** (Abrechnungswährung, Zeitformat, Netzfrequenz)	**c_{12}** (Geschlecht, Geburtsdatum)
Semistatisch c_{2j}	**c_{21}** (Jahreszeit, saisonale Gegebenheiten (z.B. Badesaison))	**c_{22}** (Einkommen, berufliche Tätigkeit, Anzahl der Kinder)
Dynamisch c_{3j}	**c_{31}** (Wetterlage, Verkehrssituation, Verspätungen im ÖPNV, Aktienkurse)	**c_{32}** (Aufenthaltsort, Umgebungsgeräuschpegel, Displaygröße)

Tabelle 1: Kontextdimensionen

In Abhängigkeit von diesen Kategorien können nun Aussagen über die Gewinnung der jeweiligen Kontextparameter gemacht werden:

- Öffentliche statische Kontextparameter (c_{11}) werden bei der Installation des MoMa-Systems durch die Konfiguration festgelegt.
- Öffentliche semistatische Kontextparameter (c_{21}) werden seitens des System-Betreibers manuell gesetzt.
- Öffentliche dynamische Kontextparameter (c_{31}) werden vom MoMa-Betreiber von spezialisierten Providern abgerufen.
- Private statische und semistatische Parameter (c_{12}, c_{22}) werden durch aktive Profilierung gewonnen. Da sie sich definitionsgemäß recht selten oder nie ändern, ist der hierdurch entstehende Aufwand für den Endnutzer akzeptabel.
- Die Parameter des privaten dynamischen Kontexts (c_{32}) sind durch das Endgerät des Nutzers für jeden Auftrag zu erheben.

5 Schlussbetrachtung

Das vorgestellte MoMa-System ermöglicht mobiles Marketing unter besonderer Ausnutzung von Kontextinformationen und gewährleistet trotzdem ein hohes Maß an Anonymität, womit den datenschutzrechtlichen Anforderungen entsprochen wird. Dies macht die Unterscheidung von privaten und öffentlichen Kontextparametern notwendig. Der Endnutzer erhält nur personalisierte Angebote, die seinen vorher definierten Aufträgen entsprechen. Darüber hinaus wird die Gefahr des Spammings durch die unterschiedliche Bepreisung der Kundenkontakte und den im System vorgesehenen Beschwerdemechanismus in Form eines Feedback-Loops auf ein Minimum reduziert.

Die Usability wird durch die automatisierte Verwendung von Kontextinformationen und die Nutzer- sowie Benachrichtigungsprofile unterstützt. Zudem ist die Verwendung von zum jeweiligen Endgerät passenden Templates für die Benachrichtigung vorgesehen.

Die beteiligten MoMa-Kooperationspartner aus der Industrie planen das vorgestellte MoMa-System im Rahmen der WM 2006 produktiv einzusetzen.

Literaturverzeichnis

[Ch00] Cheverst, K., Providing Tailored (Context-Aware) Information to City Visitors. In (Brusilovsk, P.; Stock, O.; Strapparava, C., Hrsg.)): Proc. of the conference on Adaptive Hypermedia and Adaptive Webbased Systems, Trento, 2000.

[Co04] Connect: Weltweit hat jeder Fünfte ein Handy. Connect.de, http://www.connect.de/d/55354, Abruf am: 14.12.2004.

[Du03] Duhr, E.; Naujok, H.; Danker, B.; Seiffert, E.: Neues Datenschutzrecht für die Wirtschaft, Teil 2. Datenschutz und Datensicherheit, 2003.

[He04] Heise: Spam-Anteil bei E-Mails im Mai auf Rekordhoch. Heise Online, http://www.heise.de/newsticker/meldung/48076, Abruf am: 05.11.2004.

[PK00] Pfitzmann, A.; Köhntopp, M.: Anonymity, unobservability, and pseudonymity: A proposal for terminology. In (Federrath, H., Hrsg.): Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability, Berkley, Springer, Heidelberg, 2000, S. 1-9.

[Re04] Regulierungsbehörde für Telekommunikation: Jahresbericht 2003 – Marktdaten der Regulierungsbehörde für Telekommunikation und Post. 2004.

[SAW94] Schilit, B. N.; Adams, N. I.; Want R.: Context-Aware Computing Applications. In: Proc. Of the IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, Ca, 1994. IEEE Computer Society, 1994; S. 85-90.

[So04] Sokolov, D.: Rabattstreifen per SMS. Spiegel Online, 20.10.2004. http://www.spiegel.de/wirtschaft/0,1518,323749,00.html, Abruf am: 01.12.2004.

Nokia Lifeblog – towards a truly personal multimedia information system

Andreas Myka

Nokia Corporation
Nokia Ventures Organization
PO Box 407
FIN-00045 Nokia Group
andreas.myka@nokia.com

Abstract: Nokia Lifeblog is a novel commercial application for mobile phone and PC that automatically creates a multimedia diary from the user's personal mobile data. This electronic diary helps users to collect, find, organize, move, share, publish and archive personal multimedia data in an effortless way. It collects personal multimedia data automatically and enhances it with reliable metadata. Thereby, the mobile phone acts as the data collector, data enhancer, portable data viewer and sharer, whereas the PC acts as the data archive, enhanced data viewer, and search tool. The data repository of Nokia Lifeblog implements a distributed data repository that consists of a data store on the phone and a data store on the PC.

1 Introduction

In addition to their original use, today's mobile phones are turning more and more into life recorders: They are equipped with camera functionalities that can be used for taking images and videos, they have inbuilt audio recording capabilities, they store outgoing and incoming messages such as text messages, multimedia messages and emails, they log incoming and outgoing calls, they store all contact details and calendar data, and they can hold up to gigabytes of digital data. At the same time, many people have their phones always with them and thus have them always at hand if something noteworthy happens. As a result, a lot of personal digital data is either created on or routed through mobile phones.

The increase in personal digital data is further fostered by the widespread use of digital cameras. Due to the facts that digital memory is comparably cheap and does not require a lot of physical space, it is not uncommon for individuals to create thousands of digital images per year in addition to hundreds of personal digital messages incl. emails and text messages. This data, even including low quality images, is not thrown away light-heartedly, because users are often emotionally attached to it.

There are two main challenges that are originating from the significant amount of personal digital data coming from and stored on mobile phones: On the one hand, it is becoming more and more challenging to organize and hence retrieve the information that has been created by the user [Weinb04]. On the other hand, even though mobile phones can already have a lot of memory, for example, when MMCs are used, this memory is, and will continue to be, more limited and more expensive than secondary storage in personal desktop computers such as, for example, storage space on hard disks.

But modern mobile phones also provide for the remedies that help to meet and overcome these challenges:

- They provide for a lot of context information (such as location information, calendar data, and connectivity information) that can be tapped in order to describe and subsequently organize and retrieve the user's digital data.
- They also allow the transfer of data either via local or via remote connectivity to different storage media and/or to backup media.
- For users who want to annotate and/or edit mobile digital data –such as digital images-, mobile phones, in contrast to ordinary digital cameras, provide for tools that enable them to do that while being on the move; for example, T9 can be used to enter information.
- Phones that are based on Series 60 provide for a simple relational DBMS and are easily programmable [Digia02, EdBa04].

The goal of Nokia Lifeblog was to create a digital shoebox that enables end users to manage their personal mobile data in an effortless way. Here, the term management covers especially capturing, organization, retrieval, sharing, and archiving of such data.

Currently, mobile data types in Nokia Lifeblog comprise images, videos, text and multimedia messages, text notes and blog entries; thus, Nokia Lifeblog is a true multimedia system. It makes use of context data –timestamps, names, location information, and sender/recipient information– in order to describe such multimedia data and render in searchable.

Nokia Lifeblog implements a distributed data repository with partial replication. This data repository spans across two physical devices (PC and Nokia Series 60 Phone) and uses several data containers on these two devices, incl. relational databases and folder structures in file systems.

This paper provides for a "tour d'horizon" through Nokia Lifeblog. It is clear that such an end-user system is inherently different than research systems, such as Microsoft's MyLifeBits [GBLDW02], especially in terms of its scope and goal-setting. It is also clear that machines and software that implement personal archives have been thought about and designed already a long time before; see [Bush45]. Yet, such systems are still waiting for their widespread deployment. The goal of this paper is to present an overview over a multimedia archive for personal mobile data that is, to the best of the author's knowledge, the first of its kind as commercially available software.

2 Nokia Lifeblog

2.1 Background

Nokia Lifeblog is originating from a research project that had started at Nokia Research Center five years ago. The focus of this project was on supporting the user in various ways through intelligent use of mobile data through analysis of any personally recorded digital data and related metadata. It included work on diverse topics such as image analysis and location data clustering, as well as on concept creation and validation by means of performing focus group studies. The latter were especially useful in order to find out what users are currently doing with their analogue personal data and hence, what core set of functionalities would make sense in an application that is targeted at common end users rather than at “techies” or at business people who deal with software in their everyday life.

Later, the project was moved to Nokia Ventures Organization for commercialization. The first version of Nokia Lifeblog was released for Nokia 7610 or Nokia 6670, respectively and for PCs in September 2004. Nokia Lifeblog 1.5, which also supports the Nokia 6630 and the Nokia 6260, followed in December of 2004.

2.2 Considerations

In the following section, some of the main design drivers for Nokia Lifeblog 1.0 and 1.5 will be explained.

Automatic

Even though it is clear that the quality of data management and retrieval can be significantly improved by tapping the user, it is important to provide for an end-user application that can work without such interaction: Any kind of interaction requires effort and possibly disturbs the user; thus, users are normally only willing to take this burden for very important data. Therefore, data import and organization in Nokia Lifeblog is as automatic as possible: On the phone, it detects all new text messages, multimedia messages, images, and videos automatically and imports them into the system.

On the other hand, however, it is important not to over-engineer: Some kind of data analysis might not be necessary or might even be disturbing, especially if it provides for questionable results. Therefore, the decision for the first version of Nokia Lifeblog was to use the most reliable metadata: timestamps, object names, location (country information), and –for messages- sender and recipient information.

Figure 1 Timeline navigation in PC version of Nokia Lifeblog

Time-based organization

Even though the organization of material in folders is compelling due to its logarithmic access complexity, it comes with severe disadvantages: Keeping a consistent hierarchical organization of data is tedious and requires a lot of effort from the user both at the time of storing and at the time of retrieval. To make things worse, normally, there is more than one hierarchy that users would want to use and thus, need to maintain.

In contrast, time as an organization paradigm for personal data is straightforward. Most of the data can be dated according to the time when it originated or according to the time when it entered into the user's life. Nevertheless it has to be made sure that users are provided with a natural way of navigating through time; for most of the data it would be hard to come up with the precise point in time each time when such data is looked for. Therefore, it was decided for Nokia Lifeblog to make use of a horizontal timeline that arranges all material in a time-based ordering without the strict layout as known from electronic calendars. Thereby, the appearance adapts to the amount and the distribution of data; see Figure 1.

When applying time as an ordering parameter, another advantage of mobile phones is made use of: In a PC environment, every attached device has its own clock and it is common that their clocks are drifting; therefore, in such an environment it is possible that the time-based ordering of data, coming from different origins, is incorrect. In mobile phones, it can be made sure that the sequence, in which digital data has either been created or was received, can be reproduced correctly. In addition, the clock of a mobile phone can be adjusted through the network; thereby, the phone can even provide for a rather accurate time in addition to the correct time-based sorting of material.

Mobile memory

As the different data types that are covered by Nokia Lifeblog –images, videos, messages, notes, and blog entries– each have their own storage area on the mobile phone, either in the file system or in an application-specific database, it would be tempting to copy all data into a separate database and handle them there.

However as already pointed out, compared with secondary storage on PCs, memory on phones is limited and expensive. Therefore, it is important not to waste any memory on the mobile phone and store references to objects rather than objects themselves in the database of Nokia Lifeblog. Of course, these references need to be updated each time item names and locations are modified or items are deleted by other applications. Thus, a much more complex solution is needed in order to monitor any changes.

Privacy

Personal data has to be kept private as long as the user wants it to be that way. Therefore, it should be obvious to the users what they are sharing when they are sending an object to somebody else. As compelling it may seem to store metadata as part of the actual data –thus, making sure that the metadata is not lost when the object is moved– it would come with the disadvantage that the user might share information she does not want to share, for example, if the location information is embedded in the image file.

Openness

On the other hand, it is also important to give users full control over their data. It is thus important to use standard and open ways of storing data that allows the user to access their data even when Nokia Lifeblog is unavailable.

Therefore, in Nokia Lifeblog the basic multimedia data is left in its original environment on the mobile phone; thereby, it remains accessible through the usual interfaces and applications. On the PC, the metadata is stored in a relational database through SQLite [SQLite] and the multimedia objects are stored in the file system according to the following table:

Table 1 File types

Data type	File type
Images	JPEG (plain JFIF or EXIF, depending on the type of the original image)
Videos	3gpp (as created by mobile camera application)
SMS	Text file
MMS	Text file (SMIL) + attachments in files according to their respective data types
Notes	Text file
Blog entries	Files according to the data types in the blog post

Simplicity

As mentioned earlier, simplicity was one of the key drivers in order to make Nokia Lifeblog a suitable solution for personal mobile data management for a wide range of users. Thus, it was also decided to exclude non-core functionalities such as image editing functionalities.

Fun

Even though Nokia Lifeblog can be described in serious terms such as "distributed personal information system" or "data warehouse", it was important to create a system that is appealing and fun to use and thus is used during leisure time.

2.3 Features

Nokia Lifeblog is an application that is running on PCs equipped with MS Windows XP and MS Windows 2000 and on the following mobile phones: Nokia 7610, Nokia 6670, and Nokia 6630 which are based on Series 60 2.1 and 2.6, and thus on Symbian OS 7.0s. Both ends –PC and mobile phone– are connected via local connectivity when synchronization is to take place based on SyncML over Obex; see Figure 3.

Figure 2 Screenshots of S60 Nokia Lifeblog: timeline, single image, details, and full screen view

The most important features of Nokia Lifeblog are:

- *Data import:* Data (images, videos, MMSs and SMSs) and metadata (name, time, country location based on cellid, sender/recipient information) are collected automatically on the phone and transferred to the PC when the user starts a synchronization session. Thus, the phone acts as the main data collector, whereas the PC acts as the main data archive. On the PC, data import can be triggered manually by the user; in such a case, only timestamps –for example, as part of EXIF information–, are used as metadata automatically. Additional metadata can be entered manually.
- *Notes:* Users are able to insert notes into Nokia Lifeblog which are then created as independent objects that are also visible in the timeline. These notes can be independent or related to other objects. The creation of independent notes enables users to use Nokia Lifeblog in a similar way as they would use an ordinary diary.

- *Data viewing:* Independent of its physical location, data of all types is shown with a thumbnail representation in a timeline; see Figure 1 and Figure 2 (left screen). The user can navigate in time by dragging a handle (PC) or by means of using the left and right cursor keys (Phone). In addition, the user can select a specific day and move there directly. The user can also adjust the timeline representation via filters on data types, thus, getting a more concise view. From the timeline, the user can select individual items for detail viewing, see Figure 2, or for further operations.
- *Data searching:* All text information that is part of the multimedia data and the metadata is searchable on the PC. The user can initiate a full text search by means of entering one or more search terms. If the user triggers a search, it acts similar to a filter, that is, the search results are presented in the timeline.
- *Favorites:* The user can signify items as "Favorites" and thereby move them to a virtual box both on PC and on the phone. If an image is moved to the Favorites on the PC that is currently not on the phone, it will be transferred from the PC to the mobile phone during the next synchronization. Thus, the PC can act as the main data store from which data can be loaded onto the mobile phone as needed. Also, items in the Favorites box are used for two-way synchronization whereas items that reside in the timeline only, will only be used for one-way synchronization from phone to PC.
- *Data sharing:* From the phone, items can be shared with other people by means of MMS, E-Mail, Bluetooth, and, if applicable, by means of SMS. From the PC, items can be shared by E-Mail. Both from PC and from the phone, the user can also add items to a weblog.
- *Data modification:* Users can modify most of the metadata for objects and can rotate images. If any such modification affects objects within the Favorites, these modifications will be propagated to the counterpart database, either on the phone or on the PC. In case modifications of the same metadata happen in both databases, conflict resolution will take place based on the time when the modifications happened.

Some of the features are described in a more intuitive manner on the Nokia Lifeblog web page [Lifeblog].

2.4 Database Architecture

Nokia Lifeblog data repository

Although this complexity is hidden from the user, data is stored in Nokia Lifeblog in various locations and environments. One personal data store that is created in a specific account on the PC is tightly coupled with the mobile data store on a specific mobile phone. The user can link a different mobile Nokia Lifeblog data store to the PC version at any time, but this requires an explicit decision and unlinks the previously linked mobile data repository.

As can be seen in Figure 3, the data store on the mobile phone consists of a central database, the file server and the messaging store. The file server and the messaging store contain the actual multimedia objects that are part of Nokia Lifeblog, whereas the database contains the references to the objects and any information about the objects, such as the information whether they have already been transferred to PC, whether they are part of the user's Favorites, when they have been last modified and all other object-related metadata. As a result of the fact that the pure multimedia objects continue to reside in their native environments, all other applications can still access the data without noticing the presence of Nokia Lifeblog. The Nokia Lifeblog database is a relational database that is managed via the standard Symbian relational database management system.

Figure 3 also shows the data store on the PC. Except for the newest data that has not been transferred from the phone yet, it contains all of the personal multimedia objects that the user has collected with Nokia Lifeblog. The PC repository consists of a folder hierarchy that stores the multimedia objects –see also the mapping from data types to file types as listed in Table 1– and a relational database that is managed via SQLite [SQLite]. Again, the database contains the metadata and the references to the actual objects. Also, any kind of text information that is included in the multimedia objects is stored in the database in addition to the text metadata that is already stored there. Thus, when a search is initiated, the search engine can cover all text information that is related to the multimedia objects.

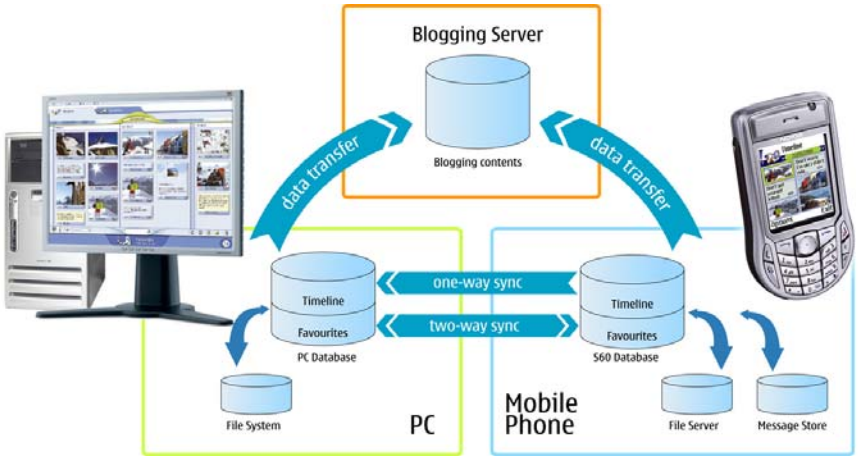


Figure 3 Lifeblog data repository

The data store on the phone and the data store on the PC are synchronized by means of SyncML over Obex when phone and PC are connected via USB or Bluetooth: For objects that are residing in the timeline only, a transfer from phone to PC takes place. For objects that are part of the Favorites on both the mobile phone and the PC and for objects that have been added to the Favorites since the last synchronization session on either the mobile phone or the PC, a two-way synchronization is performed.

One or more additional data repositories come into play if users decide to upload their data to a weblog. An upload to a blog can be done from either the phone or from the PC and is based on an extension of ATOM that is implemented on the Typepad server [Typepad]. The contents of an upload operation are then wrapped into a new multimedia object and this object is then stored in Nokia Lifeblog.

Data import and data tracking

On the mobile phone, one of the most crucial elements of Nokia Lifeblog is the automatic data tracker; see Figure 4. The data tracker is running even when the main application is not and is monitoring all changes in the file server and in the messaging store that are related to Lifeblog-compatible data types. When such changes occur -new objects are created or old objects are modified or deleted-, the Nokia Lifeblog database gets updated by the data tracker. Thus, the tracker enables the mobile Nokia Lifeblog application to provide for an integrated view on all compatible objects that are currently stored on the phone.

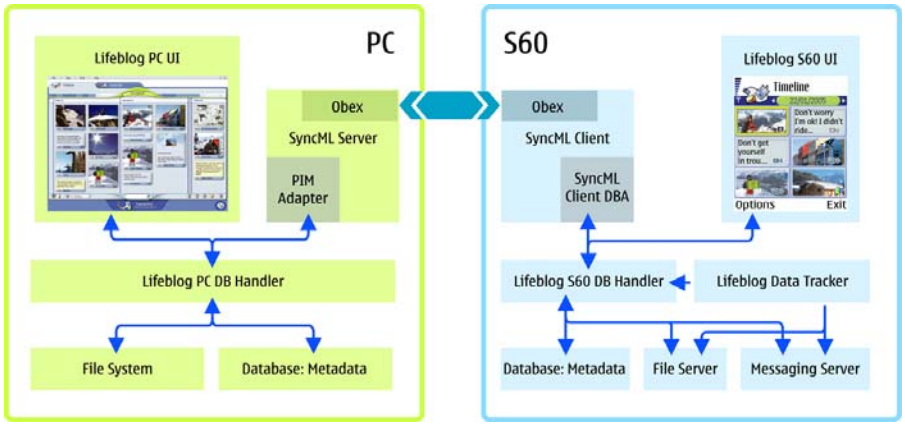


Figure 4 Components of S60 Nokia Lifeblog involved in data tracking and synchronization

On the PC, no separate data tracker is running. Instead, objects enter the PC database either via data transfer from the phone or via manually triggered import. If the user and/or another application modify the data that is stored in the file system, these changes are noticed the next time a modified object is opened in Nokia Lifeblog. Such changes are then promoted to the PC data store and, subsequently, to the phone's data store, if needed.

3 Outlook

Nokia Lifeblog provides for an automatic multimedia diary writing tool that organizes the main mobile data types by means of using metadata. It is the first of its kind and thus, it will evolve based on user needs, technology evolution and the fit in the market place. In this context, adaptations could especially affect the coverage of data types, the use of metadata and the architecture of the data repository.

With regard to the data types that are covered by Nokia Lifeblog, Series 60 phones host more mobile data types than what are currently available in Nokia Lifeblog. Even though the Lifeblog data types can be considered to be the mobile core data types, it has to be continuously verified, also via user studies, whether this assumption is correct in order to make sure that the user's electronic diary really includes all relevant life snippets.

Series 60 phones still offer for many more options where context information could be tapped in order to create metadata for multimedia objects and render them searchable in more powerful ways. However, for an end user application such as Nokia Lifeblog, it is important to make sure that several criteria are met in addition to pure technical feasibility before a feature finds its way into the product (see Section 2.2). Nevertheless, it is intended to integrate more metadata into Nokia Lifeblog in order to enrich the user's search experience.

Also, even though the database setup for Nokia Lifeblog serves its purpose for the current data contents, it might have to undergo changes in order to cope with a more holistic set of data and the aforementioned adaptations and extensions.

References

[Bush45] Bush, Vannevar: As we may think. Atlantic Monthly, July 1945, pp. 101-108.

[Digia02] DIGIA Inc: Programming for the Series 60 Platform and Symbian OS. John Wiley & Sons, 2002.

[EdBa04] Edwards, Leigh and Barker, Richard: Developing Series 60 Applications. A Guide for Symbian OS C++ Developers. Addison-Wesley Professional, 2004.

[GBLDW02] Gemmell, Jim; Bell, Gordon; Lueder, Roger; Drucker, Steven; Wong, Curtis. MyLifeBits: Fulfilling the Memex Vision. ACM Multimedia '02, pp. 235-238.

[Lifeblog] Nokia Lifeblog, Homepage, URL: http://www.nokia.com/lifeblog/.

[SQLite] SQLite, Homepage, URL: http://www.sqlite.org/.

[OMA] SyncML initiative, URL: http://www.openmobilealliance.org/tech/affiliates/-syncml/syncmlindex.html.

[Typepad] Typepad Weblogging Service, URL: http://www.typepad.com/.

[Weinb04] Weinberger, David: Point. Shoot. Kiss It Good-Bye. WIRED Magazine, Issue 12.10, October 2004.

Mobile Application Development – now and then
Towards a Handbook for User-Centered Mobile Application Design

Vision Paper

Susanne Boll, Universität Oldenburg (susanne.boll@informatik.uni-oldenburg.de)
Martin Breunig, Universität Osnabrück (mbreunig@uni-osnabrueck.de)
Birgitta König-Ries, Universität Jena (koenig@informatik.uni-jena.de)
Florian Matthes, Technische Universität München (matthes@in.tum.de)
Thomas Schwarz, Universität Stuttgart (thomas.schwarz@informatik.uni-stuttgart.de)

Abstract. Mobile applications are rapidly gaining importance. However, when developing these applications, one notices that existing design methodologies and tools do not sufficiently support these new classes of applications. In this paper, we take a closer look at the distinguishing features between mobile design and "traditional" software design. We argue that a mobile design methodology is needed and identify key issues that such a methodology needs to support. Finally, we propose the development of a handbook for mobile application design.

1 Introduction

This paper stems from a working group on "Mobile Application Design" that was part of the Dagstuhl seminar on "Mobile Information Management" which took place in October 2004. The results of this working group in a nutshell were: Mobile application design is more difficult than traditional software design. Therefore, a specific mobile design methodology is needed. In this vision paper, we describe how mobile applications are developed today and how we envision a design methodology to support mobile application development in the future. Our vision is that this methodology should stem from a community effort and should materialize in a handbook for mobile application design.

The starting point for the discussions in the working group was the observation that it is harder to design mobile applications than "normal" ones. Starting from there, we tried to identify reasons why this is the case. Our approach to doing so was to first collect and discuss sample scenarios for mobile applications and their characteristics. We then shared our experiences in developing applications for some of these scenarios. We quickly realized that the term "mobile applications" covers an area that is too broad to come up with any common characteristics for all the different application domains and scenarios. Therefore, we decided to concentrate on applications supporting mobile users. From there, we tried to identify design

dimensions that need to be regarded when designing user-centred mobile applications. The sheer number of dimensions identified is an indication of the complexity of the mobile design process. We discussed the notions and the issues of mobile application development along these identified dimensions. We concluded our discussions with the realization that a handbook for mobile application design would be extremely helpful. Such a handbook gives hints about how to handle the individual dimensions and identifies interrelationships between those dimensions.

The remainder of this paper is structured as follows: In Section 2, we briefly summarize the scenarios regarded. Section 3 describes our experiences with developing mobile applications. Section 4 gives an overview of important design dimensions, while Section 5 provides a compilation of existing approaches. Finally, Section 6 explains about the idea to create a design handbook.

2 Scenarios for Mobile User-Centered Applications

In order to understand why mobile application design is more complex than traditional software design, it is helpful to take a look at the different classes of mobile applications. In this paper, we are going to focus on mobile user-centered applications, i.e., applications supporting mobile users (in contrast to, e.g., applications where exclusively or primarily the code is mobile or applications with no human user).

In the *Tourist Scenario*, mobile users with their devices need location-related and context-related information provided by an institution. The presented information ranges from background information on sights, interactive maps, and route planning to hotel and restaurant guides including finding nearby ones and displaying their ratings. Additionally, special behind-the-scenes or themed tours (e.g. Heidelberg in the 1800s) are conceivable. There are both non-commercial and commercial variants of this scenario.

The *Mobile Gaming Scenario* is an Adventure Game in the "real world", without intrusive AR equipment. In this scenario mobile users in different roles explore a physical area and solve virtual and real world riddles and tasks, alone but also cooperatively to achieve a common game goal. Besides the game idea and the technical realization, important aspects of these games include learning from social interaction as well as an increase of creativity of mobile users.

An example application within the *Travel Scenario* is a "Bring me home button" on the mobile device. This scenario is task-oriented and has high practical relevance due to a high frequency of usage.

The *Personal Memory Scenario* represents the use of mobile applications for remembering personal experiences, improving access to personal information (date, location, people, ...), and easily sharing and extending personal "memory" by direct interaction with peers.

The Sales Force Support Scenario and "blue collar" scenarios focus on the support of mobile workers, either by making relevant information available outside of the office or by enabling the tracking of objects, e.g. via RFID tags.

Quite a variety of mobile applications can be regarded within the *Health-Care Scenario.* Examples include workflow-support for doctors and nurses, e.g. making patient information available during ward rounds, but also applications to enable patients to remain connected to health-care providers and support for monitoring patients in home healthcare settings.

The *"In the field" Scenarios for different forces*, e.g. military, police, rescue personal etc., have in common the use of sensors, the need for monitoring moving objects, the necessary support for decision making in the field etc.

The *Dating scenario* represents interacting with peers by toothing (bluetooth contacting), i.e. to organize conference meetings or taxi sharing at airports.

This list shows that even when restricting the focus to user-centered mobile applications, there still remains a wide variety of applications and thus a wide variety of problems to be tackled. Nevertheless, the next section will show some common issues that need to be taken into account.

3 Experience with developing mobile applications

This section makes the attempt to summarize a brain-storming like discussion on our experiences with developing mobile applications. Most of the experiences listed were made by a significant number of the working group participants. Thus, we believe, that all the aspects mentioned are relevant not only for the concrete situation in which they occurred, but are of broader interest. Also, we tried to identify those experiences that were strongly related to the fact that we were developing *mobile* applications, i.e. experiences that would not have occurred if the software developed had been for a desktop computer system.

The experiences can be classified in three major groups: scenario-related, prototype-related and general.

Scenario-Related Experiences:

Scenarios are an excellent tool for communicating ideas and visions. They can be used to motivate the need for funding, to make sure that user requirements and

expectations have been properly understood, and also to communicate within a development group. However, even a well worked-out scenario is not a replacement for a requirements analysis and a clear specification.

Scenarios are particularly suitable for new (visionary) application domains. They might be less appropriate when the task is the optimization or mere adaptation to mobility of existing applications. However, even in these cases, a scenario-driven approach might help to exploit the full potential of mobility.

A big advantage of scenarios is that it is easy to center them on and around the user. Thus, a scenario will help to focus application development on the real user needs. This is even more important in mobile applications than in desktop ones. The main reason for this is that often mobile users can not be focused on the mobile application, but will be in a more complex usage situation and prefer to use intuitive, appliance-like applications. For some scenarios, e.g. vehicle ones, determining what degree of intrusiveness is appropriate is a non-trivial task.

Scenario-driven development should be clearly distinguished from technology-driven development. In the first case, a (visionary) usage scenario drives the development, in the latter case, a "neat" technology is the driving force and a showcase scenario is added later on to justify the development. In our experience, scenario-driven development brings up the fancier, farther reaching, and more influencing, but sometimes also easier to implement ("simple things work better") applications than the technology-driven development. The hard thing is to get both together: to showcase cutting-edge technology using fancy applications.

Prototype-related Experiences:

On the one hand, prototype development is harder for mobile applications, on the other hand it is even more crucial here than in the traditional case. One reason why it is harder is the comparatively high cost of development of meaningful prototypes. In particular, the cost and difficulty of content creation (in particular rich AV content) have to be considered. Also, if prototypes are to be used in evaluations, they need to run on appropriate, i.e. often expensive, devices. For instance, users will not be willing to evaluate a mobile game if forced to use heavy laptops. The limited resources of small but handy devices increase the complexity of the prototype development even further.

Prototypes are particularly crucial in mobile application development because often they are the only way to identify non-anticipated "difficulties" arising from new technology and new usage situations. Some difficulties that have been encountered by members of the working group and that are specific to mobile applications are for example:

- GPS coordinates are in practice less precise than in theory.
- There can be a mismatch between the DB model and the state of the real world.

- In one case, the designated users refused to take laptops with them to field work for fear of their cars being broken into with such a valuable cargo.

Also, users may use the system different from the expectation the designers had.

General Experiences

There exists a certain conflict between "practical relevance" (& non-commercial funding) and "openness towards innovation". Typical examples for this are mobile gaming applications. They allow developers and users to realize new and innovative mobile applications but may run into the difficulty of the practical and even more the commercial relevance.

One of the central experiences is that simple things may work better than complicated solutions. While this certainly is true for almost every piece of software, it seems to be particularly true in mobile environments. One reason seems to be the different usage situation: Often, the user will not concentrate on the mobile application, but will be busy doing something else, e.g., driving a car. In such a context, the user is easily overwhelmed by too much or too complex information or interaction. Examples of successful, easy solutions are: “toothing” (dating scenario) and “sound navigation” (volume only, not volume + sound pattern).

4 Design Dimensions for Mobile Applications

When trying to generalize from the characteristics described in the scenarios’ section and our own design experiences, a number of dimensions that need to be taken into account when designing mobile applications can be identified. Not all of these dimensions will be of importance to any given application; however they can be used as a guideline.

It is important to note, that these dimensions are not necessarily orthogonal. In the contrary, a number of them are highly dependent on one another. This makes it impossible to regard them individually, e.g., in a sequential order. Rather, it is important to be aware of the interdependencies and to ensure common modeling of interleaved aspects. Existing and different mobile applications in different application domains for different user groups show different characteristics. However, to our observation, they share the commonality that they all realize different dimension, though to a different degree.

During the working group meeting, we compiled the following list of dimensions. It is quite possible, that further analysis of the problem will identify further dimensions or result in a more detailed understanding of the dimension. We believe, however, that this list is a good starting point for a more systematic approach to mobile application design. The order in which the dimensions are listed is random:

Scenario-related dimensions

- Story design
  - This aspect covers the question what the central story of the design is. Even though certain mobile applications do not necessarily actually instantiate a "story" the central underlying message and the central goal of the application needs to be identified.
- Task design
  - Within the task design the single tasks a user performs to achieve a certain goal need to be identified. These can be among others navigation, orientation, finding, seeing, investigating, and learning in and with the mobile environment.
- Spatial layout
  - Here, movement of the application and the user needs to be modeled. In particular, it has to be taken into account whether movement occurs indoors or outdoors. Depending on the application one may need to keep track about users returning to the same place. Also, the design where to put base stations and where to go for (inter-)actions is covered by this dimension.
- Temporal layout
  - This dimension deals with all temporal aspects of the application: When is the application running? How long does it take in total and/or in different steps? How many users will there be at the same time? Will they be interacting, collaborating or competitive? Will the application be used predominantly/ exclusively in a synchronous or asynchronous mode? What does that imply for the communication mechanisms?
- Spatio-temporal design, user movement
  - The spatio-temporal design aspects of a mobile application concern those design aspects that have to be additionally considered, because the users´ location and topology in 2D (or 3D) space as well as its change over time are influencing the software design process. This concerns the requirements analysis, the implementation and the evaluation of the software developed. During the requirements analysis it is important to select special spatio-temporal requirements of the mobile application, such as retrieval of the location of the mobile users in time, maximum required spatial area, expected movements, expected speed of the mobile users, expected size of areas for ad-hoc networks, other types of spatial or spatio-temporal database queries, topology of the users, neighborhood relationships, history of software development etc.

Interaction-related dimensions

- Interaction design, interaction patterns
  - This dimension deals with modeling the interactions between mobile application and the user, in particular: Which

interactions are expected or needed between the mobile application and the user? Can we identify interaction patterns?

- Collaboration
  - Are the users achieving a collaborative task? If so, is this done alone, in pairs, in groups, sequentially or in parallel? Does the collaboration interrelate with the spatial position of the users or the temporal course of the application?
- Modality Design
  - Mobile devices typically support more than one modality. Since mobile users are oftentimes not concentrating exclusively on the application at hand, the appropriate modality needs to be decided on carefully. The input and output modalities need to be determined: which are the most suitable input and output modalities for the user group and application task at a certain time, in a certain situation, at a certain point, in a certain context? The degree of intrusiveness needs to be taken into account: To which degree should or even must the user be left alone or warned/informed about an application or environmental change?

User-related dimensions

- User groups
  - It is crucial, to clearly identify the targeted user group(s). For whom is this application, e.g., a single user or user groups. What is the average age and background of the expected user? What are his or her physical and cognitive capabilities? , What is his or her cultural background? What are typical user situations and intentions at different points in time? This is of course true for traditional applications, too. However, the impact of these decisions is bigger in mobile environments.
- Usage (user) situation,
  - This dimension takes a closer look at the constraints under which the application is used: Typically aspects to take into consideration are: Where is the user? What is she/he currently doing? Who else is close to the user?
- Personalization design
  - Here, it needs to be decided how user adaptive is the application to be.
- User (profile) design
  - Should users be able to use the application anonymously or does the system call identified users,? Should information about language, preferences etc. be kept? How is this information to be used?
- Role of the participants
  - Are there stationary users? , Who are the mobile users? Is there an audience? Are there different roles of users with respect to the application?

Data/Content-related dimensions

- Content design
  - For the application it is to be decided what content is needed for the application and in which different media types.
- Data design
  - This dimension covers on the one hand "traditional" aspects like the data model, on the other hand, mobility induced (or complicated) issues like the distribution of data and data movement design are dealt with here.
- Context design
  - The aspects covered here are: how context adaptive is the application? What is the relevant context? How should it influence the application?

Communication-related dimensions

- Communication
  - Are the users communicating with other users, and/or with a central / distributed server and/or with a location? Do they leave messages, send messages, and find messages? Is communication synchronous, or asynchronous; unidirectional or bidirectional, point-to-point or broadcast? How does the application support disconnected operation?

Orthogonal aspects that cannot be subsumed under one of the dimensions above, but apply for mobile application development are, e.g.

- Scalability, cost of the game, devices needed, security, privacy issues, network traffic, local storage, etc.
- Device "independent" design, heterogeneous devices, changing capabilities of devices
- Technology design and selection
  - which technology is used, software, protocols, devices, hardware, network, etc.
- Content implementation design
  - rely in existing content, get it, integrate it, up to date content
- Implementation design, implementation plan
- Prototype planning, prototype realization
- Field test planning and evaluation planning
- Field test and evaluation

5 From Ad-hoc Experiences towards a Design Handbook

The main part of the paper has shown that mobile application development is indeed a complex task. We have listed different classes of mobile applications, talked about common experiences and in particular have identified the design dimensions which

need to be taken into account. While there are a number of approaches that support the design of one or the other of the dimensions mentioned above, to our knowledge, up to now, there is no unifying design methodology applicable for mobile applications.

Without doubt, efficient design of mobile applications is highly desirable. This is particularly true, since it is to be expected, that in the not so far future the majority of applications to be developed will be mobile or will at least have some relation to mobility. Therefore, in our opinion, a design methodology is definitely needed. First our goal is to evaluate the example to elaborate relevant aspects of user-centered mobile application and develop a design handbook as a result. The handbook possibly plays a role in proposing new process model concepts.

The potential uses of a design handbook could be mobile application development in education and as a research methodology. Moreover an industrial usage of the handbook is also possible for mobile application development projects.

Our discussions in the working group at Dagstuhl revealed that many participants have a great interest in the development of such a design handbook. In order to bring it to life we decided on the following approach: This and similar documents will be used to solicit further contributions to such a handbook. At the same time, efforts are undertaken to obtain the possibility to edit a special issue of an appropriate journal on the design process. Contributions to this journal should be made by members of the working group, but also by people from the outside.

Acknowledgements: This paper reflects the discussions of a working group at the Dagstuhl Seminar "Mobile Information Management" which took place in October 2004. Besides the authors, the following researchers participated in the working group: Nigel Davies (Lancaster University), Christian Jensen (Aalborg University), Rainer Malaka (European Media Lab, Heidelberg), Christoforos Panayiotou (Cyprus University), Simonas Saltenis (Aalborg University). They made valuable contributions to the discussion which are reflected in this paper.

A Appendix: A Compilation of Existing Approaches

This appendix contains a preliminary compilation of existing approaches in the area of mobile applications design. We do not claim that this compilation is complete. However, it should be a good starting point for any reader wishing to explore one or several of the dimensions and scenarios mentioned above. Also, it is meant as a starting point for the work on a design handbook described in Section 5.

Some of the papers explicitly describe approaches to the design of mobile applications, while others describe more en passant how the applications where

developed. When looking at the list, it becomes pretty obvious, that, while a lot of work exists on individual aspects of the design of mobile applications and on designing specific, typically very narrowly focused mobile applications, a uniting, widely accepted method that integrates the treatment of different aspects is still lacking. Also, quite often, mobile applications are developed without any explicit design methodology. In our opinion it is time to change the design of mobile applications from being an art to being an engineering discipline.

The following list is structured as follows: First, approaches that are focused on particular applications are presented. Then, approaches dealing with cross-cutting concerns like infrastructures, user interfaces etc are listed. Finally, extensions to the UML modelling method are compiled.

A.1 Approaches towards the Development of Mobile Tourist Guides

[AA+97] G.D. Abowd, C.G. Atkeson, J. Hong, S. Long, R. Kooper, M. Pinkerton: Cyberguide: A Mobile Context-Aware Tour Guide. In: ACM Wireless Networks 3 (1997), pp. 421–433.

[CDM00] K. Cheverst, N. Davies, K. Mitchell, A. Friday, C. Efstratiou: Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences. In: CHI' 00, Netherlands, 2000.

[KAA96] S. Long, R. Kooper, G.D. Abowd, C.G. Atkeson: Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study. In: 2nd ACM International Conference on Mobile Computing and Networking, Rye, New York, USA, 1996.

[KBB04] J. Krösche, S. Boll, J. Baldzer: MobiDENK – Mobile Multimedia in Monument Conservation. IEEE MultiMedia 11 (2004), pp. 72–77.

[MZ00] R. Malaka, A. Zipf: DEEP MAP challenging IT research in the framework of a tourist information system. In: 7th International Congress on Tourism and Communications, ENTER2000, Barcelona, Spain, 2000.

[PKK01] G. Pospischil, H. Kunczier, A. Kuchar: LoL@: a UMTS location based service. In: International Symposium on 3rd Generation Infrastructure and Services, Athens, Greece, 2001.

[UL02] S. Uhlirz, M. Lechthaler: LoL@ - City Guide. Prototyp einer kartenbasierten UMTS-Applikation. In: Geowissenschaftliche Mitteilungen, Schriftenreihe der Studienrichtung Vermessungswesen und Geoinformation TU Wien (in German), vol. 58, pp. 171-182, 2002.

A.2 Approaches to the Development of Mobile Games

[BF+01] S. Bjrk, J. Falk, R. Hansson, P. Ljungstrand: Pirates! - using the physical world as a game board. In: Interact'2001, IFIP TC.13 Conference on Human-Computer Interaction, Tokyo, Japan, July 9-13, 2001.

[BKW03] S. Boll, J. Krösche, C. Wegener: Paper chase revisited - a real world game meets hypermedia. In: 14th Conference on Hypertext and Hypermedia, Nottingham, UK, 2003.

[C+02] A. C. Cheok et al. Game-City: A Ubiquitous Large Area Multi-Interface Mixed Reality Game Space for Wearable Computers. In: Intl. Symp. on Wearable Computers (ISWC 2002), Seattle, WA, USA, Oct 2002.

A.3 Papers Dealing with the Development of Personal Memory Applications

[AGL04] A. Aris, J. Gemmell, R. Lueder: Exploiting Location and Time for Photo Search and Storytelling in MyLifeBits, Microsoft Research Technical Report MSR-TR-2004-102, October 2004.

[GLB03] J. Gemmell, R. Lueder, G. Bell: The MyLifeBits Lifetime Store. In: ACM SIGMM 2003 Workshop on Experiential Telepresence (ETP 2003), Berkeley, CA, November 7, 2003.

[GW+04] J. Gemmell, L. Williams, K. Wood, G. Bell, R. Lueder: Passive Capture and Ensuing Issues for a Personal Lifetime Store. In: 1st ACM Workshop on Continuous Archival and Retrieval of Personal Experiences (CARPE '04), New York, NY, USA, 2004.

[Nok05] Nokia, Foto Life Blog. http://www.nokia.com/nokia/0,1522,,00.html?orig=/lifeblog

A.4 Approaches for the Design of Mobile Business Applications

[KG04] A. Köhler, V. Gruhn: Analysis of Mobile Business Processes for the Design of Mobile Information Systems. In: EC-Web 2004.

[VH02] P. Valiente, H. van der Heijden: A Method to Identify Opportunities for Mobile Business Processes. In: SSE/EFI Working Paper Series in Business Administration No 2002:10, August 2002.

A.5 Approaches for Context-Aware Application Development and Context-Modeling

[DA99] A. K. Dey, G. D. Abowd: Towards a Better Understanding of Context and Context-Awareness. Institute of Technology, College of Computing, Technical Report GIT-GVU-99-22, June 1999.

[DR+00] A. Dix, T. Rodden, N. Davies, J. Trevor, A. Friday, K. Palfreyman: Exploiting space and location as a design framework for interactive mobile systems. In: ACM Transactions on Computer-Human Interaction, vol. 7, no. 3, pp. 285-321, 2000.

[HI04] K. Henricksen, J. Indulska: A Software Engineering Framework for Context-Aware Pervasive Computing. In: IEEE International Conference on Pervasive Computing and Communications (PerCom), 2004.

[KMK+03] P. Korpipää, J. Mäntyjärvi, J. Kela, H. Keränen, E.-J. Malm: Managing context information in mobile devices. Pervasive Computing, 2(3):42–51, 2003.

[NM04] D. Nicklas, Bernhard Mitschang: On building location aware applications using an open platform based on the NEXUS Augmented World Model. In: Software and Systems Modeling, 2004.

[RG+03] N. Rump, K. Geramani, J. Baldzer, S. Thieme, A. Scherp, J. Krösche, J. Meyer: Potentials of pervasive computing in highly interactive workplaces. In: 10th ISPE International Conference on Concurrent Engineering: Research and Applications, Madeira Island, Portugal, 2003.

[TOT+04] S. Tamminen, A. Oulasvirta, K. Toiskallio, A. Kankainen: Understanding mobile contexts. Personal Ubiquitous Comput., 8(2):135–143, 2004.

A.6 Papers on Infrastructures for Mobile and Context-aware Applications

[BH+04] C. Becker, M. Handte, G. Schiele, K. Rothermel: PCOM--A Component System for Pervasive Computing. In: IEEE International Conference on Pervasive Computing and Communications (PerCom), 2004.

[GS+02] D. Garlan, D. Siewiorek, A. Smailagic, P. Steenkiste: Project Aura: Towards Distraction-Free Pervasive Computing. In: IEEE Pervasive Computing, special issue on Integrated Pervasive Computing Environments, Vol.1, Nr. 2, 2002.

[JS03] G. Judd, P. Steenkiste: Providing Contextual Information to Pervasive Computing Applications. In: IEEE International Conference on Pervasive Computing and Communications (PerCom), 2003.

[SDG99] D. Salber, A. Dey, G. Abowd: The Context Toolkit: Aiding the Development of Context-Enabled Applications. In: CHI, 1999.

[CP+02] N. H. Cohen, A. Purakayastha, L. Wong, D. L. Yeh: iQueue: a pervasive data-composition framework. In: 3rd Intl. Conf. on Mobile Data Management, 2002.

[WD+04] X. Wang, J. Dong, C. Chin, S. Hettiarachchi, D. Zhang: Semantic Space: An Infrastructure for Smart Spaces. In: Pervasive Computing, IEEE, 3(3), July-September 2004.

A.7 Approaches for the Design of Multimodal User Interfaces

[DKG02] Donker, H., Klante, P., Gorny, P.: The design of auditory user interfaces for blind users. In: NordiCHI ’02, ACM Press, 2002.

[KKB04] P. Klante, J. Krösche, S. Boll: AccesSights – A Multimodal Location-Aware Mobile Tourist Information System. In: 9th International Conference on Computers Helping People with Special Needs (ICCHP’2004) – Special Thematic Session (STS): Accessible Tourism, July 2004.

[MSB04] W. Mueller, R. Schaefer, S. Bleul: Interactive Multimodal User Interfaces for Mobile Devices. In: 37th Annual Hawaii International Conference on System Sciences (HICSS'04) – Track 9 January 05 - 08, Big Island, Hawaii, 2004 .

[WDJ+04] F. Wegscheider, T. Dangl, M. Jank, R. Simon: A Multimodal Interaction Manager for Device Independent Mobile Applications. In: 13th International World Wide Web Conference, New York, NY, USA, May 17-22 2004.
[W3C02] W3C, Multimodal Interaction Activity. http://www.w3.org/2002/mmi/.

A.8 Papers Describing Spatio-Temporal Aspects of Mobile Application Development

[Brin02a] T. Brinkhoff.: The Impact of Filtering on Spatial Continuous Queries. In: 10th International Symposium on Spatial Data Handling (SDH 2002), 2002.
[Brin02b] T. Brinkhoff: A Framework for Generating Network-Based Moving Objects. In: Geoinformatica, Vol. 6, No. 2, Kluwer, 2002, pp. 153-180.
[Brin99] T. Brinkhoff: Requirements of Traffic Telematics to Spatial Databases. In: 6th International Symposium on Large Spatial Databases (SSD 1999), Hongkong, 1999.
[BTB+03] M. Breunig, C. Türker, M.H. Böhlen, S. Dieker, R.H. Güting, C.S. Jensen, L. Relly, P. Rigaux, H.-J. Schek, M. Scholl: Architectures and Implementations of Spatio-Temporal Database Management Systems. In: Koubarakis et al. (eds.), Spatio-Temporal Databases: The CHOROCHRONOS Approach, Springer-Verlag, 263-318, 2003.
[GAD04] R.H. Güting, V.T. de Almeida, Z. Ding: Modeling and Querying Moving Objects in Networks. Fernuniversität Hagen, Informatik-Report 308, April 2004. To appear in the VLDB Journal, 2004.

A.9 Approaches Extending UML to Allow for the Design of Mobile Applications

[BKKW02] H. Baumeister, N. Koch, P. Kosiuczenko, M. Wirsing: Extending Activity Diagrams to Model Mobile Systems. In: NetObjectDays, 2002.
[Kos02] P. Kosiuczenko: Sequence Diagrams for Mobility. In: ER Workshops, 2002.
[KRS+01] C. Klein, A. Rausch, M. Sihling, Z. Wen: Extension of the Unified Modeling Language for Mobile Agents. Unified Modeling Language: Systems Analysis, Design and Development Issues 2001: pp. 116-128.
[KT04] M. Kang, K. Taguchi: Modelling Mobile Agent Applications by Extended UML Activity Diagram. ICEIS (4) 2004, pp. 519-522.

Atomicity in Mobile Networks

Joos-Hendrik Böse[1], Stefan Böttcher[2], Sebastian Obermeier[2], Heinz Schweppe[1], Thorsten Steenweg[2]

[1] Freie Universität Berlin, Institute of Computer Science, Takustr. 9, 14195 Berlin, Germany
boese@mi.fu-berlin.de, schweppe@mi.fu-berlin.de
[2] University of Paderborn, Computer Science, Fürstenallee 11, 33102 Paderborn, Germany
stb@uni-paderborn.de, so@uni-paderborn.de, steenweg@uni-paderborn.de

Abstract: Atomicity, one of the essential transaction properties, is guaranteed in fixed-wired network by protocols, like the 2-phase commit protocol, which assumes that faults like node and link failures rarely occur. However, node and link errors which are considered as an exception in fixed-wired networks can be assumed to be the default case in mobile networks. Therefore, depending on the application, mobile networks require different protocols for guaranteeing strict atomicity. Within this paper, we look at different application scenarios and their requirements w.r.t. atomicity. We distinguish classes of scenarios in which atomicity can be guaranteed from classes of scenarios in which atomicity cannot be guaranteed. Furthermore, we will show what kind of atomicity guarantees can be guaranteed in the different scenarios.

1. Introduction

1.1. Motivation

Providing transaction processing with guaranteed transactional behavior in volatile environments is a difficult task. Constant change in available connections and frequent link failures leave little to build guarantees on.
Atomicity has been studied in different areas of computer science and is fundamental to simplify reasoning about the correctness of distributed applications and dealing with concurrency and fault tolerance in complex systems, like distributed databases, peer-to-peer systems and service oriented architectures. The notion of atomicity in this work is understood as a means to maintain consistent states in distributed systems, because any execution schedule of atomic operations will result in a correct state of the overall system. We will only consider strict atomicity, i.e. concepts like relaxed atomicity or semantic atomicity are beyond the scope of this work.
In distributed systems, communication between participants of atomic operations is needed to agree on the new correct state of the overall system. If communication between the participants is assumed to be asynchronous and unreliable, certain problems like the distributed attack problem [Gr78] are unsolvable. Although protocols like the 2-phase commit protocol show a blocking behavior [BHG87], they are found to be practically applicable in fixed-wired networks, because in contrast to

mobile environments, these cases of blocking are rare in fixed-wired networks and can be recovered using extensive recovery strategies. So they do not significantly affect the operability of the system.
However, in mobile networks we assume link failures to occur frequently. Hence in such environments, protocols like the 2-phase commit protocol are not applicable as blocking must be expected as normal behavior of the system and not as the exceptional case. Nevertheless, applications deployed in mobile networks form complex distributed systems, which have strong demands for atomicity in terms of agreements on new states. Examples for such atomicity requirements are money atomicity and goods atomicity as introduced by [Ty96].
In this paper, we will describe what atomicity guarantees can be given in mobile networks for different kinds of applications.
Settings like the coordinated attack problem are well understood and proved to be unsolvable in case of link failures and asynchronous communication as shown in [Gr78]. Hence, we will weaken the requirements of applications w.r.t. atomicity and show what kind of transactional atomicity guarantees can be given.
We will set up a specific system-model and show which atomic guarantees can be given within identified classes of applications and coordination problems. We will show that a certain class of applications depends on requirements of the distributed attack problem with link failures, whereas other applications do not depend on these requirements. We will identify characteristics of the latter class of problems and show how atomicity can be achieved.

The remainder of the paper is organized as follows. Section 1.2 reviews the research done on the problem of distributed consensus with link failures and summarizes the current state of the art. Section 2 describes basic assumptions of our system-model. In Section 3 two different classes of application scenarios are described, each representing a class with common requirements. Section 4 classifies problems as well as the atomicity guarantees that can be provided. Finally, Section 5 concludes and summarizes this work.

1.2. Related Work

The impossibility of the deterministic version of the coordinated attack problem is proved in [Gr78]. Approaches to solving this problem by making some probabilistic assumptions about the loss of messages while keeping processes deterministic have been tried. Another approach, described in [VL92], allows processes to use randomization, and considers the possibility of violating the agreement and/or validity condition. Both approaches guarantee to reach correct agreements.
Activities that directly treat the basic problem of unreliable links are approaches using cross-layered designs, as the layered approach to computer networks introduces an abstraction level that loses useful information about the connection context. But this information can be used to calculate parameters, e.g. the probability of message loss. For example, communication with nodes that move in different directions is more likely to fail than a link between nodes moving in the same direction. The same idea can be applied to information about energy resources as done in [FG04].

How to design distributed data processing systems in fixed networks is well researched in [Ly94]. Most conventional distributed databases deployed for fixed-wired networks use the 2-phase commit protocol ([Gr78]) or variations like the 3-phase commit protocol ([Sk81]) to achieve an atomic commit of distributed transactions. Relaxing strict atomicity to semantic atomicity has led to optimistic 2-phase commit protocols like [LKS91]. Protocols like [HRG00] propose a commit protocol for transactions with timing constraints, so-called real-time transactions for distributed systems. In contrast to [Li02] these protocols were designed for fixed-wired systems. A protocol for real-time transactions in mobile environments is proposed in [Li02]. However, this protocol offers only relaxed atomicity and not strict atomicity.

Most work about transaction processing in mobile environments like [DHB97, WC99, Ch93] assumes a system model where transactions are processed between mobile clients and a database server residing in a fixed network, that can be reached via base stations that are connected to the network. Transaction processing between mobile nodes is not the scope of these activities.

A timeout based approach to atomic commit decisions is proposed in [Ku02], where the coordinator decides for global commit if it does not receive a failure message from a node in the commit set within a predefined timeout period. Mobile nodes can tell the coordinator to extend this period if they need an extension. The main problem here is to calculate an appropriate value for the time-out period as it depends on a number of system variables which could be difficult to quantify.

2. Assumptions of our system-model

In our system-model, we assume that mobile nodes communicate with each other directly using some kind of wireless technology. Because of node movement, direct communication between mobile nodes is very unreliable, i.e. every node and every link may fail at and for an unforeseeable time. Mobile nodes can also communicate with mobile support stations that are connected to a fixed-wired network. We assume that failures of the fixed-wired network occur rarely and especially that the commit decision information that is stored in the fixed-wired network will not get lost. However, due to movement of the mobile client, communication between the fixed-wired network and the mobile clients is also unreliable.

Furthermore, we assume that when a device reconnects to the system after a link failure, it can check the status of the transaction it was processing when the failure occurred, i.e. we assume that commit status information of transactions is stored at a safe place (e.g. within the fixed-wired network).

3. Application Requirements and Scenarios

This section describes general requirements to atomicity and two classes of application scenarios which differ in their atomicity requirements. We use these scenarios to identify common characteristics and to classify the addressed problems.

3.1. General Requirements to Atomicity in Mobile Networks

The requirements of atomic commit protocols can be grouped into general requirements which are common to all our mobile application scenarios and specific requirements which additionally apply only to some of the scenarios. The general requirements are mostly similar to atomic commit protocols in fixed-wired networks. These requirements include:

1. The atomic commit protocol must direct all participating processes to the same decision.
2. If a participating process disappears irrevocable (e.g. a mobile device gets lost or damaged) before its vote is received by a coordinator, its vote has to be treated as a vote for abort.
3. No process can withdraw its decision. This includes dying processes, whose votes have already been received by a coordinator.
4. The global decision must be to commit, if all involved processes vote for commit, and these commit decisions are received by a coordinator within a predefined time-out period.
5. If all failures are recovered and no new failures occur for a sufficiently long time, every process reaches a final decision.
6. If no failures occur, all messages are delivered without delay, and all participants vote for commit, the protocol must decide for commit.
   (Note that the fourth requirement and the last requirement are more relaxed than the usual requirements for atomic commit protocols because we do not require a common decision to be commit if messages do not arrive in time.)

Another group of general requirements limits the effects of link failures to the directly involved nodes. Hence we define further requirements:

7. No party shall reside in a state of uncertainty for an arbitrary long time due to link failures of another node. More precisely, when a link from a mobile partner to a safe coordinator fails, at most this mobile partner should be blocked. However, other participants that have voted for a commit should be able to continue their work before the failing link is recovered.
8. Similarly, when a mobile node participating in a transaction fails, running participants with a working link to a coordinator should not be blocked.

3.2. Class 1: Recursive Atomic Decision Scenarios

Within the first class of atomic commit scenarios, the individual participant's decision depends on whether or not the decision itself is guaranteed to be communicated in

time to the other parties. We call these atomic commit decision problems *recursive atomic commit decisions*.

A well known example for such a recursive atomic commit decision is the coordinated attack problem outlined in [Gr78]. Here, two generals must agree on a time for a common attack using an unreliable communication channel. Within this scenario, a guaranteed decision is not possible under the assumption of message loss. The reason is that the decision of one general is recursively dependent on the decision of the other general and on the knowledge that the communication of the decisions does not get lost. The decision is recursively dependent in the sense that one party can promise to attack at a certain time only if that party knows that the other party is going to attack at the same time, which implies that the first party knows that the second party knows that the first party is going to attack at this time. The main problem here is that because of the unreliable communication no party can be sure that messages containing a commitment to a certain attack time are received by the other party before that time. Thus the origin of this recursive relationship lies in the demand to agree on a certain time to attack.

An example from the real world involves two cars exchanging information (e.g. about traffic conditions) while passing each other. The first car that provides useful information for a second car wants to receive some kind of electronic cash in return from the second car. And vice versa, the second car is only willing to pay, if it will get the information. Communication is limited and unreliable in the sense that information can only be exchanged within a short time frame. Furthermore, we assume that there is no global coordinator available. The second car will not pay if it is not sure that it will receive the information paid for (and vice versa the first car does not uncover the information if this car is not sure that it gets paid). There is no guarantee that a message reaches the other party within the time that the distance between the two cars allows for communication. Here, the commit decision is recursive in the sense, that the decision to pay or to uncover a piece of information depends on the knowledge that payment or information reaches the other party.

A similar situation occurs between mobile nodes that want to exchange electronic goods between each other, if we assume that nodes do not trust each other and therefore, no node wants to send its items first since it must assume to get nothing in return. Note, that in the cars' scenario the problem of a recursive decision occurs even if the parties trust in the "correct behavior" of the other party, but cannot trust in the channels used to transmit their messages, i.e. there is no guarantee that their messages are delivered in time.

Within recursive commit decisions (i.e. class 1 scenarios), the commit decision result depends on whether or not this decision is guaranteed to reach the other parties within a given time. In other words, one requirement is to come to a decision before a deadline is reached *and* the result of the decision depends on whether or not the deadline can be met.

3.3 Class 2 Scenarios: Independently Completing Transactions

The characteristic of this second class of scenarios is that after agreeing on a commit decision, the transaction can be executed on each client, independently of the other clients' execution status. The transaction can be executed immediately on each client after a commit decision has been received.
When a node disappears forever (i.e. a mobile device is irreparably damaged) after the coordinator received its vote, we assume that the transaction state reached on this mobile device has no influence on the transactions states of the other involved nodes. However, the other mobile devices participating in the transaction have to follow the commit decision, if the commit status has been decided.
To emphasize the difference between this class of scenarios and the recursive class 1 scenarios, we create a slightly modified coordinated attack problem. In this scenario, the two generals do not need to agree on a certain attack time, but on a capitulation decision: e.g. raise a white flag for capitulation or a black flag for continuing the defense. It is not necessary that these flags are raised at the same time, but all parties must agree on the same decision. This means the siege is over only if all parties raise the white flag.
Another scenario that applies to this group is trading of electronic goods in a trusted environment. Within this scenario, information is sold for E-cash. Assuming that all partners within this scenario trust each other, the decision to send an item (E-cash or information) first has no effect on the outcome, as every node trusts in receiving items in exchange at a later point in time.
Another scenario we have in mind is the scenario where a number N of mobile customers, that meet in a store, pool together in a corporate buying transaction to attain a volume-N discount. It must be guaranteed that each individual customer voting for "buy" deposits his share of the payment at a coordinator. In addition, he should not be able to withdraw his buying decision. If all the N customers agree on the decision "buy", the discount buying transaction can be completed – otherwise the buying transaction is aborted and the mobile customers get their money back. The decision is not recursive if we assume that all customers are willing to wait until they are informed about the result ("buy" or "not buy") of the buying decision.

4. Atomicity in Mobile Environments

Assuming the requirements of the system-model described above are fulfilled, this section will show what kind of atomic guarantees can be given in such an environment and what kind of guarantees cannot be achieved.
For all class 1 scenarios, atomic commit decisions are limited by guaranteed message delivery times in the following sense. If a maximum time interval within which message delivery can be guaranteed does not exist, a safe atomic decision for commit is not possible. In other words, the only safe decision is then to abort – however, this prevents transactions from being processed and therefore is not desirable. Timeout based protocols can also not guarantee a correct strict atomic commit decision.

For all class 2 scenarios, an atomic commit protocol that fulfills the requirements exists, i.e. when a node disappears forever (i.e. a mobile device is irreparably damaged), we assume that for the correctness of the overall application it does not matter which transaction state is reached on this mobile device. However, the other mobile devices participating in the transaction have to follow the commit decision if the commit status has been decided. If the node unexpectedly recovers, it must follow the global decision. The difference to scenarios of class 1 is that none of the involved nodes has to commit itself to finish a local transaction and to inform another participant or coordinator at a certain time. The promise a participant makes here is to commit if a coordinator tells the participant that all the others voted for commit, regardless of any time constraints. Class 1 scenarios, in contrast, cannot make this promise due to their recursive character, i.e. that the result of the decision depends on whether or not that decision is communicated at a certain time.

While in principle the 2-phase commit protocol of distributed databases could be used for class 2 scenarios, this protocol has the following major disadvantage. Every local transaction locks local resources as long as it's uncertain about the global decision. As the global decision depends on all votes, a single link failure delays the overall commit. In contrast to this, it is desirable to finish a transaction as fast as possible, respectively to reduce the probability of blocking, equivalently to striving for independent recovery under as many circumstances as possible. One possibility as also described in [UG01] is to associate the voting-phase with a time-out as follows. A commit coordinator chooses an appropriate time-out period. Whenever one of the participants or a communication link fails such that the commit coordinator does not get all the votes during that time, the coordinator treats the node from which it has not received a vote as if it had send an abort message. Otherwise, the coordinator has all votes and proceeds according to the protocol. Unless we use a cross-layer approach that does not hide status information of mobile devices from an application, it is impossible to determine if a mobile client will recover or has disappeared forever. Therefore, this time-out based mechanism is one of the few promising approaches we see to reduce blocking. Hence, it is important to find an optimal time-out period. Dynamically calculating this value based on the connection context of mobile clients should be topic of further research.

If a node participates in a transaction and causes a time-out or is temporarily unavailable, it must be informed about the transaction status after the recovery of this node. In a fixed-wired scenario, this is done by communication with the coordinator or with another participant of the transaction. In a mobile scenario, the coordinator may also be a mobile node. Therefore it is possible that the coordinator is not available. To preserve the transaction status, some kind of shared storage must be provided where recovering nodes can pick up the state of transactions they were involved in before their communication channel fails.

The most obvious technique is to place this data on a central server within a fixed network that can be reached via base stations by every mobile client. As we assume that communication with base stations is also unreliable and sometimes is not possible, other approaches to implement a shared storage that every mobile node can access should be evaluated.

5. Summary and conclusions

We have presented two classes of scenarios with slightly different requirements to atomic commit protocols. While applications that follow class 1 scenarios are not solvable by atomic commit protocols, in the sense that no protocol exists which allows deciding for commit, applications that follow class 2 scenarios are solvable by an atomic commit protocol. Because the traditional 2-phase commit protocol is not appropriate for mobile networks due to a high probability of node and link failures, we have proposed two actions that improve the performance of the 2-phase commit protocol. First, we have proposed to define appropriate time-out values during the vote-phase based on the connectivity context, and secondly to provide some kind of shared storage for mobile clients saving the state of transactions processed by this client before link failures. The concrete design and implementation of these actions is left for further research.

We consider our contribution to be a useful basis for the development of transactional applications in mobile environments.

References

[BHG87] P.A. Bernstein, V. Hadzilacos, and N. Goodman, “Concurrency Control and Recovery in Database Systems”, *Addison-Wesley*, 1987

[Ch93] Panos K. Chrysanthis. Transaction Processing in Mobile Computing Environment. In *Proceedings of the IEEE Workshop on Advances in Parallel and Distributed Systems*, pages 77--83, Princeton, New Jersey, October 1993

[DHB97] Margaret H. Dunham, Abdelsalam Helal, and Santosh Balakrishnan. A mobile transaction model that captures both the data and movement behavior. *ACM/Baltzer Journal on Special Topicsin Mobile Networks and Applications*(MONET), 1997.

[FG04] Fife, L. and L. Gruenwald, "TriM: Tri-Modal Data Communication in Mobile Ad-Hoc Networks ", DEXA 2004

[Gr78] J. N. Gray. Notes on data base operating systems. In *R. Bayer, R. M. Graham, and G. Seegmüller, editors, Operating Systems: An Advanced Course*, volume 60 of *Lecture Notes in Computer Science*, chapter 3.F, page 465. Springer-Verlag, New York, 1978

[HRG00] Haritsa, J. R., Ramamritham, K., Gupta, R.: The PROMPT real-time commit protocol. IEEE Transaction on Parallel and Distributed Systems, 11(2): pp.160-180, 2000.

[Ku02] V. Kumar, N. Prabhu, M. Dunham, Y.A. Seydim, "TCOT - A Timeout-based Mobile Transaction Commitment Protocol", IEEE Trans. on Computers, Vol. 51, No. 10, 2002

[Li02] Yunsheng Liu, GuoQiong Liao, Guohui Li, and JiaLi Xia. Lynch. Relaxed Atomic Commit for Real-Time Transactions in Mobile Computing Environment. In *Advances in Web-Age Information Management, Third International Conference, WAIM 2002*, pages 397-408, Beijing, China, August 11-13, 2002.

[LKS91] Levy, E.,Korth, H. F., Silberschatz, A.: An optimistic commit protocol for distributed transaction management. In: Proceedings of ACM SIGMOD 1991 International Conference on Management of Data, Denver Colorado, pp.88-97, 1991.

[Ly94] Nancy Lynch, Michael Merritt, William Weihl, and Alan Fekete. Atomic Transactions. *Morgan Kaufmann Publishers*, 1994

[Sk81] D. Skeen, Non-blocking commit protocols. In Proceeding of ACM SIGMOD International Conference on Management of Data, Ann Arbor, Michigan, pp.133-142, June 1981.

[Ty96] J. D. Tygar. Atomicity in electronic commerce. In *Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing*, pages 8-26, May 1996.

[UG01] Hector Garcia-Molina, Jeffrey D. Ullmann and Jennifer Widom, Database Systems: The Complete Book, *Prentice Hall PTR*, 2001

[VL92] George Varghese and Nancy A. Lynch. A tradeoff between safety and liveness for randomized coordinated attack protocols. In *Proceedings of the 11th Annual ACM Symposium on Principles of Distributed Computing*, pages 241-250, Vancouver, British Columbia, Canada, August 1992

[WC99] Gary D. Walborn and Panos K. Chrysanthis. Transaction processing in pro-motion. In Proceedings of the 1999 ACM symposium on Applied computing, pages 389-398. ACM Press, 1999

Common Profile Store für Telekommunikationsnetze

Franz-Josef Banet, Rodolfo López Aladros, Dr. Stephan Rupp

Alcatel SEL AG
Lorenzstraße 10
70435 Stuttgart
FJ.Banet@alcatel.de
R.Lopez-Aladros@alcatel.de
S.Rupp@alcatel.de

Abstract: Der Einfluss der Informationstechnologie (IT) und ihrer Produkte auf die Gestaltung der Telekommunikationsnetze der Zukunft wird weiter steigen, der Einfluss der Prinzipien und Produkte der klassischen Telekommunikation wird abnehmen. Denn neben dem klassischen Sprachverkehr für Telefonie sind die Telekommunikationsnetze vermehrt auch Träger einer steigenden Anzahl neuer Dienste, die massiv Datenverkehr erzeugen. Aus diesem Grunde werden die Kommunikationsnetze der Zukunft paketorientierte und nicht mehr verbindungsorientierte Netze sein. Der Datenverkehr hat heute schon den Sprachverkehr volumenmäßig überholt. Paketnetze sind für Datenverkehr das effizientere und wirtschaftlichere Medium.

Für die Speicherung von Profildaten von Telekommunikationsteilnehmern sind heute keine proprietären Lösungen seitens der Telekommunikationshersteller mehr nötig. Eine sinnvolle Zusammenstellung und Veredelung von handelsüblichen IT-Produkten wie Datenbanken, Servern und Speichernetzen ermöglicht die Schaffung eines Common Profile Store (CPS), das in diesem Aufsatz beschrieben wird. Im CPS sollen die Profildaten aller Teilnehmer eines Netzes und weitere teilnehmerbezogene Daten, wie z. B. Videomail und Voicemail, an zentraler Stelle zusammengefasst und berechtigten Applikationen zur Nutzung bereitgestellt werden.

Voraussetzung für die Realisierung eines CPS ist eine strikte Trennung der Applikationen von „ihren“ Daten. So müssen z. B. in Mobilfunknetzen die Home Location Register (HLR) und weitere Applikationen umgestaltet werden, sie werden im Wesentlichen auf Ihre Businesslogic reduziert. Die gesamte Datenproblematik kann an ein zentrales CPS delegiert werden. Was man dadurch erreicht ist, ist nicht nur eine verminderte Komplexität des Netzes, sondern auch eine starke Vereinfachung der Abläufe bei der Planung und beim Betreiben von Telekommunikationsnetzen [Ban01].

1 Telekommunikation gestern, heute und morgen

Mobilfunk und Festnetztelefonie haben viel miteinander gemeinsam. GSM, und damit auch UMTS, ist auf der Basis des Integrated Services Digital Network (ISDN) entstanden. Für beide, Festnetz und Mobilfunk, gab es zur Zeit Ihrer Standardisierung eigentlich nur eine einzige Anwendung, die Telefonie. Datendienste, wie z. B. Fax, sind auch möglich, stehen aber nicht im Vordergrund. Die Leitungsvermittlung ist für Telefonie die optimale Netztechnik, das ISDN und der „Sprachteil" von GSM sind folgerichtig leitungsvermittelt bzw. verbindungsorientiert [Ban04].

Mittlerweile hat in den Netzen der Anteil der transportierten Daten den Anteil der transportierten Sprache überholt. Folgerichtig werden wir in der Zukunft nicht mehr „auch Daten über Sprachnetze", sondern „auch Sprache über Datennetze" übertragen.

Wir können an dieser Stelle eine These zur Zukunft der Telekommunikationsnetze wagen:

> Das Kommunikationsnetz der Zukunft wird paketorientiert sein. Die Bedeutung der Leitungsvermittlung wird stark abnehmen.

Datennetze sind jedoch anders aufgebaut als leitungsvermittelte Netze. Datennetze sind paketorientiert, und es werden andere Technologien, die der Informatik näher stehen als der klassischen Telekommunikation, eingesetzt. Dies gilt insbesondere für die Implementierung von Diensten und Applikationen für Datennetze. Die Evolutionslinie Analogtelefonie – ISDN – GSM/UMTS wird so nicht weitergeführt, es gibt einen Bruch. Die Telekommunikation verschmilzt mit der IT, in der Telekommunikation müssen neue Pfade beschritten werden [Lop01], [Rup01].

Folgerichtig wird man zukünftig vermehrt IT-Technologien in Mobilfunk- und auch in Festnetzen finden. GSM wurde bereits durch einen „Datenteil" ergänzt, dem General Packet Radio Service (GPRS). Innerhalb des GPRS-Subsystems wird auch das Internet Protokoll verwendet. Während in der Vergangenheit ausschließlich die International Telcommunication Union (ITU) die Standardisierung des Mobilfunks prägte, nimmt nun bei UMTS der Einfluss der Internet Engineering Task Force (IETF) stark zu, z. B. bei der Definition des IP-Multimedia Subsystems (IMS) [Ban04], [Rup02].

Eine weitere These kann formuliert werden:

> In den Netzen der Zukunft wird man an Stelle der großen, monolithischen Netzelemente, wie z. B. der Mobile Services Switching Centres (MSCs), in denen nahezu alles herstellerspezifisch realisiert ist (HW, Betriebssystem, Datenbank etc.), vermehrt kommerzielle Lösungen „von der Stange" finden. Und zwar für
>
> – HW
>
> – Betriebssysteme
>
> – Datenbanken

- Speichersysteme
- SW-Methoden und Tools

Das “Reservat Telekommunikation”, in dem alle Lösungen weitgehend selbstgemacht und proprietär sind, wird es nicht mehr lange geben. Im Hinblick auf Leistungsfähigkeit, Echtzeitverhalten und Ausfallsicherheit haben viele Produkte der IT-Industrie mittlerweile einen ausreichend hohen Qualitätsstandard erreicht, der ihren Einsatz in Kommunikationsnetzen möglich macht.

Ein Beispiel für die Anwendung moderner IT-Technologien in zukünftigen Telekommunikationsnetzen wird im Folgenden vorgestellt: die Zusammenfassung und gemeinsame Speicherung von Teilnehmerprofildaten in einer zentralen Datenbank, dem Common Profile Store (CPS). Für das CPS ist es unerheblich, ob die daran angeschlossenen Applikationen verbindungsorientierte oder verbindungslose Dienste bedienen.

2 Common Profile Store

Schauen wir beispielhaft auf ein Mobilfunknetz. Die Betrachtungen sind aber auch anwendbar auf andere Telekommunikationsnetze wie z. B. Festnetze. Profildaten von Mobilfunkeilnehmern, wie z. B. die Rufnummer und subskribierte Dienste, werden heute jeweils in verschiedenen Applikationsservern verwaltet und gespeichert. Die beiden wichtigsten sind:

- Das Home Location Register (HLR), die Hauptdatenbank eines Mobilfunknetzes.
- Der Service Control Point (SCP) für Zusatzdienste, wie z. B. Prepaid, Televoting, Anrufe an 0180er-Rufnummern, etc.

HLR und SCP sind „Besitzer“ ihrer Teilnehmerdaten. Es gibt aus Kapazitätsgründen in einem Netz mehrere HLRs und SCPs, die jeweils für einen Block von Teilnehmern zuständig und geografisch verteilt sind. Eine Zentralisierung von HLRs und SCPs ist durchaus denkbar, die geografische Verteilung hat keine funktionale Rechtfertigung [Ban02], [Ban03]. Schon heute sind einige große Netzbetreiber dazu übergegangen, alle HLRs für eine große Region (z. B. Nordostdeutschland) in einem Gebäude aufzustellen.

Ähnlich wird heute mit Inhalten wie Sprach- und Textnachrichten, Bildern und Videos verfahren. Sie werden in Voicemail-, Short-Message-, Video-Message und E-Mail-Systemen gespeichert. Jedes dieser Systeme ist in sich abgeschlossen und speichert „seine“ medialen Daten, z. B. Voicemails, im eigenen Speicher. Eine Zusammenfassung der medialen Daten, die ein und demselben Teilnehmer zugeordnet sind, gibt es bisher nicht. Der hier verwendete Begriff „Profildaten“ schließt auch diese medialen Daten ein.

Mit dem CPS soll hier eine Vereinheitlichung und damit auch eine Vereinfachung geschaffen werden. Das Ziel ist eine zentrale Datenbank, die auch den Anforderungen eines Telekommunikationsnetzes gerecht wird. Diese Anforderungen sind:

- Echtzeitverhalten. Die heute erreichten Antwortzeiten in Telekommunikationsnetzen dürfen durch ein CPS nicht verschlechtert werden.
- 100% Verfügbarkeit der Dienste für den Endkunden muss gewährleistet sein.
- Alle Daten müssen 100% redundant an einem Spiegelstandort verfügbar sein.
- Das Datenmodell muss ohne Unterbrechung des Wirkbetriebs erweiterbar sein. Die Einführung einer neuen Applikation kann z. B. die Erweiterung des Datenmodells erfordern.
- Skalierbarkeit.

Die Zielsetzung des CPS ist in Abbildung 1 noch einmal verdeutlicht.

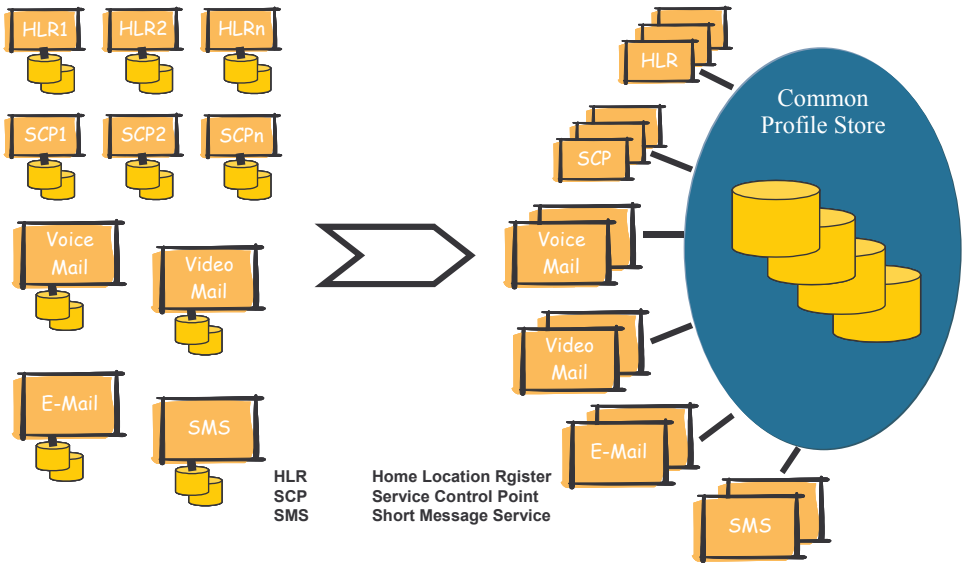


Abbildung 1: Common Profile Store als zentrales Speichersystem für teilnehmerbezogene Daten

3 Aufbau des CPS

Entscheidend beim CPS ist die Trennung der Applikationen von den Daten. Abbildung 2 zeigt den Aufbau des CPS.

Eine Applikation besteht nur noch aus der jeweiligen Business-Logic, die sich für einen Auftrag den zugehörigen Teilnehmerdatensatz aus dem CPS besorgt, diesen bearbeitet und, wenn erforderlich, modifiziert und dann ins CPS zurückschreibt. Hierzu bietet das CPS geeignete Schnittstellen an, wie z. B. LDAP (Lightweight Directory Access Protocol) und CORBA (Common Object Request Broker Architecture). Nach Auftragserfüllung werden die Teilnehmerdaten wieder vergessen. Somit ist eine Applikation nicht mehr auf einen bestimmten Block von Teilnehmern begrenzt, sondern kann jeden Teilnehmer eines Netzes bedienen.

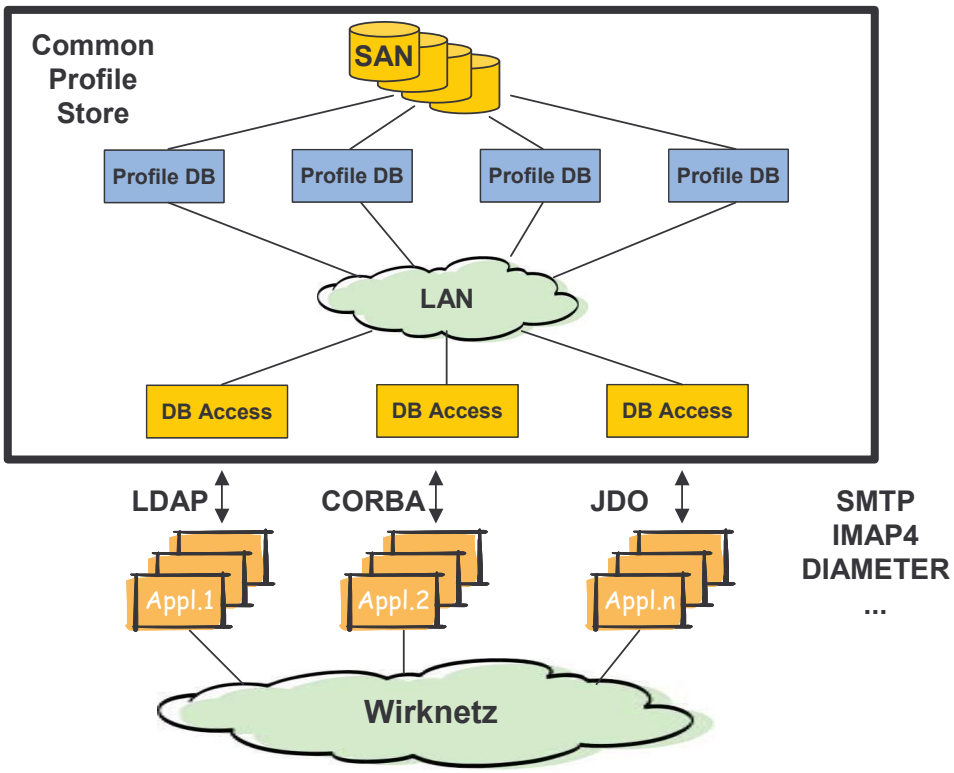


Abbildung 2: Aufbau des Common Profile Store

Das CPS selbst ist aufgeteilt in Datenbank und Speichersystem. In der Datenbank gibt es die Profile Database und das Modul Database Access. In der Profile Database werden die Profildaten bzw. die medialen Daten abgelegt. Die Profile Database ist fraktioniert, ein Verteilmechanismus sorgt für eine ausgewogene Verteilung der Datenblöcke auf die verfügbaren Datenbankserver. Das Modul Database Access interagiert mit den Applikationen, terminiert die Kommunikationsprotokolle und behandelt Prioritäten.

Über spezielle Anschlusskarten, die Host Bus Adaptor (HBA), werden die Datenbankserver an das Speichersystem, ein Storage Area Network (SAN), angeschlossen. In Abbildung 3 ist ein Datenbankserver mit nachgeschaltetem SAN skizziert.

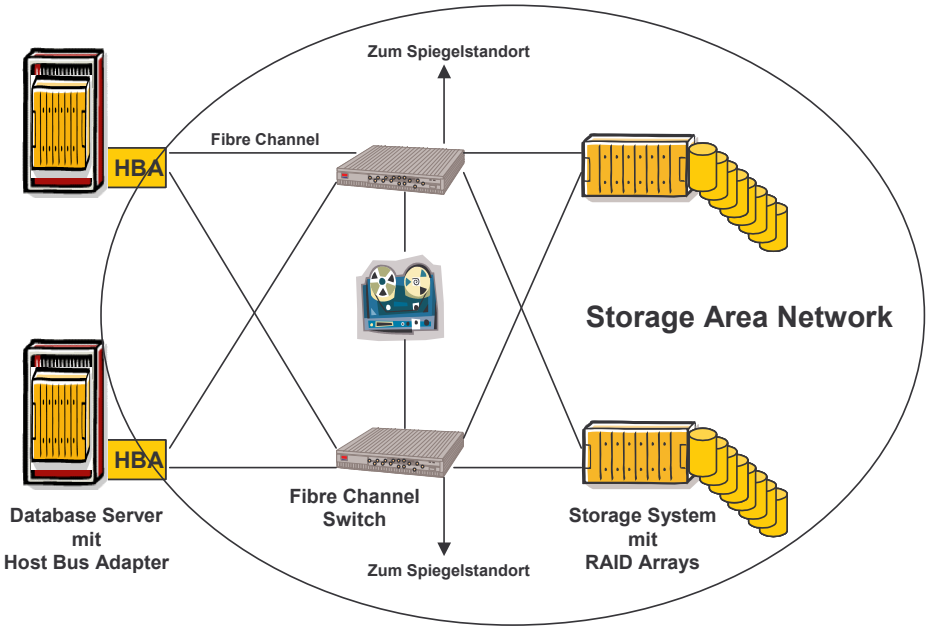


Abbildung 3: Datenbankserver mit Storage Area Network

Das Kernstück des SAN ist die Speichereinheit mit den daran angeschlossenen Festplatten. Unterschiedliche RAID-Konfigurationen sind einstellbar. Der Anschluss an Server kann direkt oder auch über Fibre Channel Switches geschehen. Die Switches, manchmal auch Directors genannt, sorgen für mehr Konnektivität. Auch eventuell gewünschte Virtualisierungen werden in den Switches eingestellt. Backupmedien wie Tape-Drives werden ebenfalls über den Fibre Channel Switch angeschlossen.

4 Vorteile eines Common Profile Store

Durch die Trennung der Daten von den Applikationen bekommen die Daten, in unserem Fall die Teilnehmerprofildaten, einen viel größeren Stellenwert, während die Applikationen auf ihre Businesslogic reduziert werden. Die Zusammenfassung der Teilnehmerprofildaten in einem CPS vereinfacht viele Abläufe bei der Planung und beim Betrieb von Telekommunikationsnetzen.

- Das CPS ist nicht beschränkt auf HLR und SCP, auch neue Applikationen können auf bereits vorhandene Profildaten zugreifen. Datenmodellerweiterungen sind „online“ möglich. Das Austesten von neuen Applikationen bezüglich der Akzeptanz beim Endkunden ist nun mit vergleichsweise geringem Aufwand möglich. Im Idealfall muss nur ein neuer Applikationsserver an das CPS angeschaltet und im Netz freigeschaltet werden.

- Eine Applikation bleibt auf ihre Businesslogic beschränkt. Datenbank und Speicherung inklusive Backup und Disaster Recovery sind bereits vorhanden.

- Die Problematik der Datenspeicherung auf Festplatten, der Anfertigung und Verwaltung von Backups, der Festplatten-Spiegelung zwischen geografisch voneinander abgesetzten Standorten, ist bereits im SAN gelöst und steht allen an das CPS angeschlossenen Applikationen zur Verfügung.

- Es gibt voneinander entkoppelte Skalierungsmöglichkeiten und Evolutionspfade in allen drei Ebenen: Applikationen, Datenbank und Speichernetz.

- In allen drei Ebenen werden vorwiegend handelsübliche Standardkomponenten für HW, Betriebssystem, Datenbank, Speicherung usw. verwendet.

- Data Mining wird durch ein CPS stark vereinfacht. Das Kapital „Teilnehmerdaten“ kann vom Netzbetreiber bzw. Service Provider viel stärker genutzt werden.

5 Geografische Redundanz

Zentralisierung bedeutet auch die Schaffung eines „Single Point of Failure“. Deshalb ist ein Spiegelstandort in ausreichender Entfernung zwingend notwendig. Bei zwei Standorten kann die Gesamtlast auf das CPS zu gleichen Teilen auf beide Standorte aufgeteilt werden. Dann wären 50% der Teilnehmerprofile im Standort 1 aktiv mit Spiegel im Standort 2, die anderen 50% im Standort 2 aktiv mit Spiegel im Standort 1. Auch eine Aufteilung auf 3 Standorte ist sinnvoll und möglich. Dann können bei geeigneter Dimensionierung während eines Wartungsfensters in einem Standort die beiden verbleibenden Standorte die gesamte Last übernehmen.

In Abbildung 4 ist die Verschaltung von zwei Standorten zu sehen. Die Synchronisierung der Daten erfolgt über eine direkte SAN-SAN-Verbindung. Fibre Channel kann mit marktüblichen Produkten in unterschiedliche Übertragungsmedien, wie z. B. Wavelength Division Mode (WDM), eingepackt und über mehrere hundert Kilometer übertragen werden. Diese Synchronisierung erfolgt auf Datenblockbasis und ist schneller als eine Synchronisierung auf der Ebene der Datenbankserver.

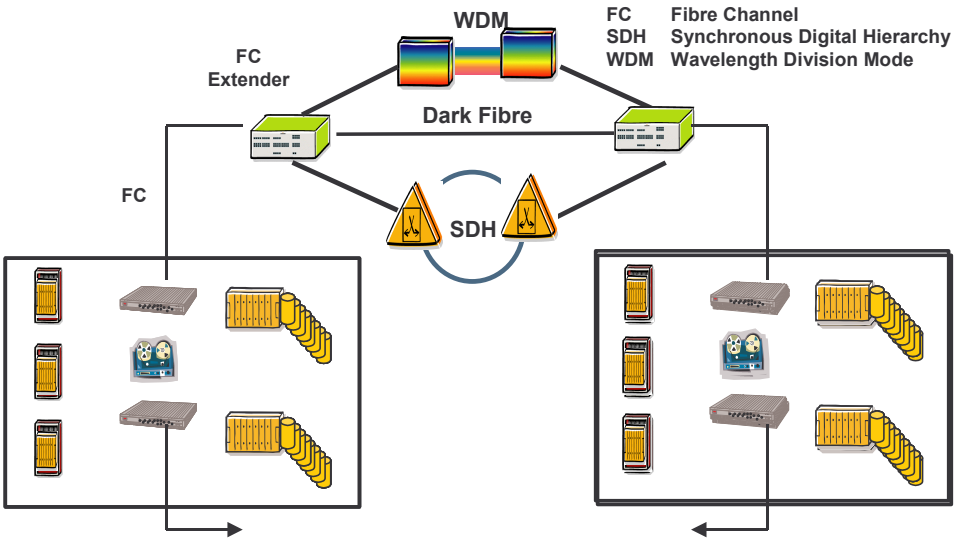


Abbildung 4: Geografische Redundanz mit zwei Standorten

Literaturverzeichnis

[Ban01] Banet, F.-J., López Aladros, R., Winter, E.: Zurück zur Ordnung – Weniger Komplexität in Mobilfunknetzen durch moderne Speichertechniken, NET Zeitschrift für Kommunikationsmanagement, Ausgabe Juni 2002.

[Ban02] Banet, F.-J., López Aladros, R., Winter, E.: Die nächste Generation operativer Netzsystemspeicher im Kernnetz von 3+G Mobilfunknetzen, VDE-Kongress 2002 Net Worlds, Dresden 2002.

[Ban03] Banet, F.-J., López Aladros, R., Rupp, S., Winter, E.: Restrukturierung der TK-Netze. Einsatz von Speichernetzen und Web Services zur Vereinfachung der Netzinfrastruktur, VDE/ITG Networkshop, Dortmund 2003.

[Lop01] López Aladros, R., Banet, F.-J., Rupp, S., Winter, E.: Restructuring the Telecommunication Networks, Working Conference HET-NETs'03, Ilkley, U.K. 2003..

[Rup01] Rupp, S., Siegmund, G., López Aladros, R., Banet, F.-J., FLEXINET – A Network Service Architecture, The Journal of The Communication Network, U.K. 2003.

[Ban04] Banet, F.-J., Gärtner, A, Teßmar, G: UMTS - Netztechnik, Dienstarchitektur, Evolution, Hüthig Telekommunikation, Bonn 2004.

[Rup02] Rupp, S., Siegmund, G., Java in der Telekommunikation – Telefonie 2.0, Grundlagen, Konzepte, Anwendungen, dpukt.verlag, Heidelberg 2004.

Ein Framework zur regelbasierten Verarbeitung von Telematik-Daten mobiler Objekte

Stefan Bell, Ulrich Derigs, Tobias Krautkremer

Seminar für Wirtschaftsinformatik und Operations Research
Universität zu Köln
Pohligstr. 1
50969 Köln
derigs@informatik.uni-koeln.de
{stefan.bell|tobias.krautkremer}@uni-koeln.de

Abstract: Die Anbieter von Telematik-Hardware haben es in den letzten Jahren versäumt, ihre Systeme mit betriebswirtschaftlichen Anwendungen aufzuwerten. Insbesondere fehlen derzeit Lösungen, die flexibel einsetzbar sind und sich für unterschiedliche betriebliche Problemstellungen in den Unternehmen konfigurieren lassen. Im Folgenden stellen wir ein System vor, das es ermöglicht, Telematik-Anwendungen flexibel zu erstellen. Eine wichtige Komponente ist dabei ein Framework, das Telematik-Nachrichten mobiler Objekte mittels einer Regelsprache in einen konsistenten und aggregierten Datenbestand überführt.

1 Einleitung und Motivation

Bereits seit mehreren Jahren rüsten Unternehmen aus verschiedenen Branchen mobile Einheiten, etwa Servicepersonal oder Transporteinheiten, mit Telematik-Technologie[1] aus, um einerseits die steigenden Anforderungen des Marktes an Serviceumfang und -qualität zu erfüllen und andererseits Kommunikations- und Informationsverarbeitungsaufwand in den Dispositions- und Controllingabteilungen zu senken. Obwohl Telematik-Hardware somit mittlerweile bereits verbreitet ist, ist in Bezug auf die effektive Nutzung dieser Technologie insbesondere bei Klein- und Mittelbetrieben noch eine große Anwendungslücke festzustellen. Üblicherweise umfassen die standardmäßig in Telematik-Lösungen integrierten Funktionen nur die einfache Visualisierung mobiler Objekte auf digitalen Karten oder ermöglichen eine elektronische Fahrzeugkommunikation. Reichhaltige, anspruchsvolle und flexible betriebswirtschaftliche Auswertungen sowie die konsequente Ausschöpfung des Potenzials der verfügbaren Real-Time-Informationen, die einen signifikanten Nutzen von Telematik-Technologie darstellen, sind bis jetzt in der Regel noch nicht implementiert.

[1] Für eine Definition des Begriffs Telematik vgl. [TuPo04] S. 187-189

Im Rahmen von Praxisprojekten wurden von Mitarbeitern des Seminar für Wirtschaftsinformatik und Operations Research an der Universität zu Köln für mittelständische Unternehmen, die bereits Telematik-Hardware im Einsatz hatten, eine Reihe unterschiedlicher , auf die speziellen Bedürfnisse zugeschnittenen Telematik-Service-Systeme entwickelt und im Unternehmen implementiert. Zu nennen sind hier:

- ein System zur Berechnung und verursachungsgerechten Zuordnung von Wartezeiten für einen Frachtführer,
- ein System zur Erhebung und Kontrolle von Maschineneinsatzzeiten für ein Bauunternehmen,
- ein System zur Erfassung von Lenk- und Pausenzeiten sowie einer vertragskonformen Berechnung von Spesen im grenzüberschreitenden Verkehr für einen Spediteur,
- ein System zur Sendungsverfolgungssystem für einen Flüssig- und Schüttguttransporteur,
- ein System zur überbetrieblichen Auftragsverwaltung und Fahrzeugdispositionssystem für einen Kurierverbund.

Die Erfahrung aus diesen Projekten zeigt eine große Überschneidung von Basisfunktionen, die eine Wiederverwendung von Systemmodulen nahelegen. Aus diesem Grund wurde am Seminar für Wirtschaftsinformatik und Operations Research an der Universität zu Köln ein generisches System konzipiert und (bisher nur prototypisch) entwickelt, das versucht, die Anwendungslücke dadurch zu schließen, dass die jeweils unternehmenindividullen Anforderungen zu einem hohen Anteil durch die Konfiguration von Standardmodulen befriedigt werden können, wodurch Entwicklungsaufwand und Entwicklungszeit akzeptabel gering gehalten werden können.

Im Rahmen der Entwicklung telematikdatenbasierter Anwendungen besteht dabei zunächst ein Problem von großer Praxisrelevanz: Die Telematik-Daten liegen aufgrund von technischen Unzulänglichkeiten oder Benutzerfehlern nicht in der aus verarbeitungslogischer Sicht richtigen (zeitlichen) Reihenfolge vor, sind lückenhaft oder widersprüchlich. Auch fallen typischerweise große Mengen von teilweise nicht relevanten Daten für einzelne mobile Objekte an, beispielsweise, wenn Positions- und Statusinformationen in Intervallen von nur wenigen Sekunden übermittelt werden. Mobile Objekte sind dabei weitgehend selbstständig operierende mobile, d.h. ortsungebundene und sich örtlich verändernde Einheiten, wie beispielsweise Transport- oder Serviceeinheiten.

Eine Lösung dieser Probleme verspricht dabei die Entwicklung eines Frameworks, das diese "rohen" Daten, ähnlich wie die ETL-Stufe eines Datawarehouses, in einen konsistenten und redundanzarmen Datenbestand überführt und dabei in der Lage ist, große Datenmengen effizient zu verarbeiten. Deswegen liegen die Aufgaben eines

solchen Frameworks einerseits in der Sicherung eines konsistenten Datenbestands und andererseits in der Datenaggregation und somit in der Reduktion des Datenvolumens

Ziel dieses Beitrags ist es, unsere Entwicklung vorzustellen und die Praxistauglichkeit dieses Konzepts der intelligenten Verarbeitung von Telematik-Daten herauszustellen. Dazu wird zunächst im zweiten Abschnitt die Konzeption des generischen Systems T.Dienst, das diese Auswertungen, so genannte Telematik-Services[2], bereitstellen soll, erläutert. Darauf aufbauend werden das Vorgehen bei der Verarbeitung von „rohen" Telematik-Daten und eine neue Sprache zur Definition von Konsistenzregeln vorgestellt und anhand eines konkreten Anwendungsbeispiels aus der Praxis erläutert. Der Beitrag endet mit einem Resümee und einem Ausblick auf weitere anstehende Forschungaufgaben.

2 Konzeption eines generischen Systems zur Verarbeitung und Auswertung von Telematik-Daten

Das anschließend beschriebene Konzept eines generischen Telematik-Service-Systems folgt dem Generatorkonzept. Das Generatorkonzept hat seinen Ursprung in der Konzeption von Entscheidungsunterstützungssystemen.[3] Ein DSS-Generator erzeugt aus einer Menge von für eine Problemdomäne entwickelter Modelltemplates durch Parametrisierung und Modellauswahl ein spezifisches, problemangepasstes Entscheidungsunterstützungssystem. Analog zu diesem Vorgehen wurde in unserem Projekt ein Generator für betriebswirtschaftliche Telematik-Services konzipiert und entwickelt.

Der Telematik-Service-Generator hält eine Menge modularer Telematik-Services für kleine und mittlere Unternehmen bereit und erlaubt es, betriebswirtschaftliche Telematik-Services individuell und bedarfsgerecht zusammenzustellen und diese zu parametrisieren. Am Ende dieser Konfiguration steht ein spezifisches, auf die Bedürfnisse eines bestimmten Unternehmens angepasstes Telematik-Service-System. Dabei soll der Benutzer des Generators iterativ vorgehen und neue Anforderungen sukzessive in die Wiederverwendungsplattform aufnehmen. Die Inhalte des Generators werden so ständig erweitert. Da ein genereller Ansatz von vornherein nicht allumfassend sein kann und zwangsweise später immer auf individuelle Bedürfnisse angepasst werden muss, kann dieser somit nur unterstützende Wirkung haben und nicht vollständig automatisiert werden.

Das generische Software-System wird mit der Zielsetzung konzipiert,

- für unterschiedliche mobile Ressourcen,

---

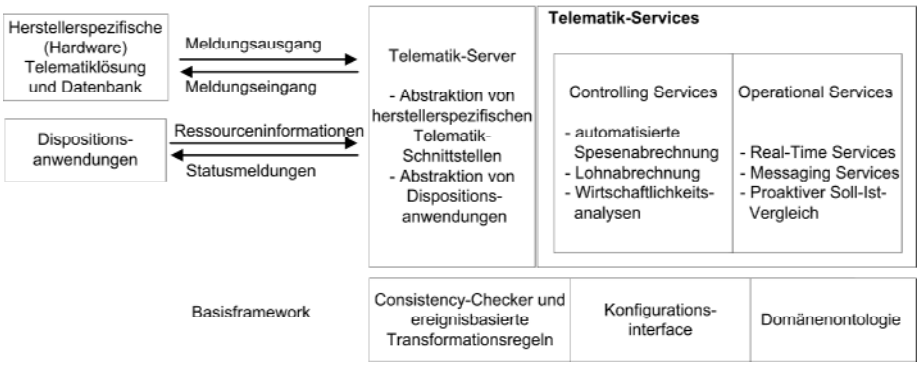
[2] Als Telematik-Services bezeichnen wir betriebswirtschaftliche Anwendungen der Telematik-Technologie, die in Unternehmen (oder für Kunden eines Unternehmens) einen Mehrwert schaffen. Beispiele bekannter Telematik-Services sind die Fahrzeugverfolgung bei Logistik-Dienstleistern oder die elektronische Übermittlung von Arbeitsanweisungen an Außendienstmitarbeiter.

[3] Vgl. [SpCa82]

- in verschiedenen Branchen,
- besonders in kleinen und mittleren Unternehmen,
- unter Berücksichtigung nicht standardisierter Telematik-Hardware und Dispositionssystemen,

flexibel das Management mobiler Ressourcen durch neue Telematik-Services zu unterstützen.

Das System muss in vier Dimensionen flexibel für diese unterschiedlichen Anforderungen anpassbar sein: Erstens muss eine Datenbeschreibung für die spezielle Branche auf der Basis der systeminhärenten allgemeingültigen Datenschemata in Form einer Domänenontologie bereitgestellt werden. Zweitens müssen die angestrebten Telematik-Services über ein Konfigurationsinterface parametrisierbar sein. Bei den Telematik-Services handelt es sich dabei sowohl um Operational-Services als auch um Controlling-Services. Drittens sollte über einen Telematik-Server eine Schnittstelle zu hierarchisch übergeordneten, bereits im Unternehmen implementierten betrieblichen Anwendungsystemen, wie z. B. einer Dispositionssoftware, geschaffen werden. Viertens muss eine Schnittstelle in Form eines Consistency-Checkers, der über Transformationsregeln verfügt, zu der eingesetzten herstellerspezifischen Telematik-Datenbank implementiert werden. Durch die Konfiguration in diesen vier Dimensionen entsteht ein spezielles System, dessen Architektur in der folgenden Abbildung verdeutlicht ist.



Komponenten und Schnittstellen des generischen Telematik-Service-Systems

Im folgenden Abschnitt werden die sprachliche Basis und die Funktionsweise unseres regelbasierten Consistency-Checkers vorgestellt.

3 Konzeption von regelbasierten Inferenzmechanismen

3.1 Struktur und Verarbeitung von Telematik-Daten

Gängige Telematik-Hardware versendet Nachrichten in der Regel über GSM-Netze unter Verwendung von SMS-Technologie[4]. Dementsprechend handelt es sich bei Telematik-Nachrichten meist um unformatierte Zeichenketten, in denen zu übermittelnde Informationen sequenziell aneinandergereiht stehen. Für die Verarbeitung und Auswertung solcher Nachrichten ergeben sich aufgrund der verwendeten Technologie zwei Problemstellungen: Zum Einen variieren Inhalt und Struktur der Botschaften in Abhängigkeit vom Hersteller der verwendeten Hardware. Hierdurch wird das Extrahieren einzelner Datenfragmente, z.B. Positionsdaten, Absender usw., erschwert. Zum Anderen bietet die SMS-Technologie keine Übertragungsgarantie, d.h. Nachrichten können mit bis zu mehreren Tagen Verzögerung oder auch überhaupt nicht zugestellt werden.[5] Daher treten regelmäßig Inkonsistenzen im Nachrichteneingang auf, weil Nachrichten fehlen oder in falscher Reihenfolge, d.h. nicht in Sendereihenfolge, empfangen wurden. Dieses Problem wird dann noch verstärkt, wenn der Nachrichtenversand nicht automatisch erfolgt, sondern von Mitarbeitern initiiert werden muss.

Ein generisches System zur Verarbeitung von Telematik-Daten bedarf aufgrund der geschilderten Sachverhalte zwingend einer intelligenten Parsingroutine, um in der Lage zu sein, einzelne Informationsfragmente aus den übermittelten Nachrichtentexten zu extrahieren. Außerdem wird ein Mechanismus zur weitgehenden Beseitigung von Inkonsistenzen in der Datenbasis (dem Nachrichteneingang) benötigt. Das ist insbesondere im Hinblick auf die Tatsache von Bedeutung, dass kleine Fehler in der Datenbasis große Auswirkungen auf nachfolgende Auswertungen sowie die Weiterverarbeitung der Daten haben können. Eine wichtige Voraussetzung für einen solchen Mechanismus ist die Möglichkeit, Inkonsistenzen, also pathologische Zustände der Datenbasis, sowie entsprechende Korrekturmaßnahmen zu beschreiben. Unser Konzept sieht für diesen Zweck eine Regelsprache vor, die im folgenden Abschnitt näher erläutert wird.

Eine einfach zu erkennende Inkonsistenz der Telematik-Datenbasis liegt etwa dann vor, wenn zwei Meldungen über „Pausenende“ ohne eine Meldung über einen zwischenzeitlichen „Pausenstart“ existieren. In diesem Fall ist in Absprache mit dem Anwender unter Berücksichtigung der Einsatzsitution evtl. nicht nur eine Erkennung sondern auch eine normierte und automatische Korrektur möglich. Eine vollständig automatische Sicherstellung der Konsistenz wird dabei nicht möglich sein, da eine formale Definition, die für alle Anwendungsfälle „sound and complete“ ist, nicht angegeben werden kann. Die in der Regelbasis mit Hilfe der Regelsprache erstellten Konsistenzregeln sind insofern notwendig aber u. U. nicht hinreichend, so dass stets

[4] SMS ist die Abkürzung für: „Short Message Service“. SMS ist in der Praxis zurzeit die favorisierte Kommunikationstechnologie, da bei GPRS/UMTS die Kommunikationskosten bei geringem Datenaufkommen noch wesentlich höher sind.

[5] Vgl. dazu [HäPeSt00]

eine zusätzliche manuelle Kontroll- und Korrekturmöglichkeit vorzusehen ist. Das Vorgehen ist insofern an der Pragmatik ausgerichtet. Das implementierte Regelsystem fungiert als Expertensystem und der gesamte Service ersetzt die bisher nur mögliche manuelle und daher oft zu aufwändige Rekonstruktion des Betriebsgeschehens auf Basis von gleichermaßen inkonsistenten Dokumenten. Final wird somit die Ausschöpfung von Erlös- und Kostensenkungspotenzialen möglich, im Beispiel des Systems zur Berechnung und verursachungsgerechten Zuordnung von Wartezeiten etwa die Möglichkeit der zusätzlichen Inrechnungstellung von Leistung an den Kunden, die zuvor nicht zuverlässig und zweifelsfrei ermittelt werden konnte.

3.2 Konzeption einer Regelsprache

Wie bereits erläutert, enthält der Bestand empfangener Telematiknachrichten aufgrund technischer und/oder menschlicher Fehler regelmäßig Inkonsistenzen, deren Beseitigung für die Auswertung und Weiterverarbeitung der Daten von großem Nutzen oder sogar unbedingt erforderlich sein kann. Da viele verschiedene Branchen und Betriebe Telematiksysteme einsetzen und für unterschiedliche Zwecke nutzen, ist es aber weder möglich pathologische Zustände einheitlich zu definieren noch können standardisierte Handlungsanweisungen zu deren Beseitigung angegeben werden. Stattdessen bedarf es im Einzelfall immer der individuellen Erstellung von Konsistenzregeln und der ständigen Evaluierung ihrer Auswirkungen im laufenden Betrieb, anhand derer die Regelbasis iterativ in einem evolutionären Prozess weiterentwickelt und an sich ändernde Rahmenbedingungen angepasst werden kann. Um solche individuellen Regelbasen erstellen zu können, haben wir eine Regelsprache entwickelt, die sich an das Konzept der ECA-Regeln aus dem Bereich aktiver Datenbanken anlehnt.[6] Gemäß des ECA-Regelkonzepts enthält unsere Regelsprache Konstrukte für die Angabe von Ereignissen, Bedingungen und Aktionen.[7]

Der Ereignis- und Bedingungsteil einer Regel dient der Beschreibung pathologischer Zustände, während im Aktionsteil die zugehörigen Korrekturmaßnahmen definiert werden. Regelereignisse gliedern sich grob in zwei Klassen. Zum Einen stellt jede Nachricht ein (Meldungs-) Ereignis dar, zum Anderen können Zeitereignisse definiert werden, die absoluten, relativen und/oder periodisch wiederkehrenden Zeitpunkten entsprechen und z.B. für Routineprüfungen wie: „Prüfe jeden Abend um 20 Uhr, ob sich alle Mitarbeiter abgemeldet haben“ genutzt werden können.

Der Bedingungsteil einer Regel entspricht einem Bool’schen Ausdruck, der in der Regel Vergleichsbedingungen mit Parametern empfangener Nachrichten enthält. Für den Zugriff auf empfangene Nachrichten existieren die Funktionen FIRST und LAST, die von einem anzugebenden Referenzzeitpunkt oder einer Referenznachricht aus die erste bzw. letzte Nachricht mit definierbaren Eigenschaften zurückgeben.

---

[6] ECA steht für „Event Condition Action“. Eine umfassende Erläuterung des ECA-Regelkonzepts und möglicher Anwendungen ist u. a. in [DiGa00] zu finden.

[7] Eine Beschreibung der Syntax befindet sich im Anhang.

Möchte man z.B. feststellen, ob vor der Meldung „Pausenende“ auch der Pausenanfang von einem Mitarbeiter gemeldet wurde, wird die Funktion LAST mit dem gemeldeten „Pausenende“ als Referenznachricht aufgerufen und die Bedingung hinzugefügt, dass der gesuchte Pausenanfang zum gleichen Mitarbeiter gehören soll wie das Pausenende der Referenznachricht. Um den Suchzeitraum zu begrenzen, kann außerdem noch ein Zeitpunkt als untere Schranke (bzw. obere Schranke bei der Verwendung von FIRST) angegeben werden. Weiterhin können Aufrufe von FIRST und LAST beliebig tief geschachtelt werden:

```
IF FIRST(Pausenende PE, LAST(Pausenanfang PA, THIS, PA.PersNr =
THIS.PersNr), PE.PersNr = THIS.PersNr) = NULL
```

Mittels dieser verschachtelten Aufrufe von FIRST und LAST lässt sich beispielsweise feststellen, ob bei Eingang einer Pausenende-Meldung (diese ist im Beispiel in der Variable THIS enthalten) ein zugehöriger Pausenanfang gemeldet wurde, dem bisher noch keine Pausenende-Meldung gefolgt ist, d.h. dass das Pausenintervall offen ist. Durch eine solche Konstruktion lassen sich vor allem viele pathologische Zustände abgrenzen, die sich auf Intervalle beziehen. Insbesondere sind dies die Fälle, dass ein Intervallende gemeldet wird, obwohl kein offenes Intervall existiert (wie im Beispiel) oder der umgekehrte Fall, dass eine Intervallbeginnmeldung eintrifft, obwohl das letzte Intervall des gleichen Typs noch nicht abgeschlossen wurde. Weil solche Intervallprüfungen in der Praxis von hoher Relevanz sind und deshalb in vielen Fällen modelliert werden müssen, sieht unsere Regelsprache ein weiteres Konstrukt vor, durch das die umständliche Modellierung der Suche nach offenen Intervallen mit FIRST und LAST wegfällt. Mit Hilfe des OPEN-Befehls kann so, ausgehend von einer Referenzmeldung, geprüft werden, ob ein Intervall eines bestimmten Typs (z.B. Pause) für einen bestimmten Mitarbeiter und/oder eine bestimmte mobile Ressource (i.d.R. ein Fahrzeug oder eine Maschine) noch geöffnet, d.h. nicht abgeschlossen ist. Abgesehen von der einfacheren Modellierung ist die Verwendung von OPEN auch deutlich performanter als der Umweg über die FIRST/LAST-Konstruktion, da die Suchroutine speziell auf das Finden offener Intervalle optimiert ist und Intervalle nur auf Basis von Mitarbeiter- und Maschinen-Ids identifizieren kann. Letzteres schränkt wiederum die Flexibilität der Suche ein, wobei Erfahrungen bei der Modellierung von Regeln in Praxisprojekten gezeigt haben, dass Intervalle in den meisten Fällen durch Mitarbeiter oder Maschinen identifiziert werden können, so dass die Verwendung des OPEN-Konstrukts möglich ist.

Der Aktionsteil einer Regel enthält schließlich die Handlungsanweisungen zur Eliminierung der durch Ereignis und Bedingung spezifizierten pathologischen Zustände. Zu diesem Zweck existieren Konstrukte zum Einfügen neuer und zum Ändern oder Entfernen vorhandener Nachrichten. Meldet z.B. ein Mitarbeiter den Arbeitsbeginn an einer bestimmten Maschine und es wird festgestellt, dass der Mitarbeiter sich noch nicht von seiner letzten Pause wieder zurückgemeldet hat (und sich demnach noch in der Pause befinden müsste), so kann man eine Regel formulieren, die in diesem Fall automatisch eine passende Pausenendemeldung für den Mitarbeiter generiert und dem Meldungspool hinzufügt.

Zur Veranschaulichung der Regelformulierung folgt ein Regelbeispiel aus einem Praxisprojekt zur Entwicklung einer individuellen Telematiklösung für einen Entsorgungsdienstleister:

```
ON      Stillstandsende
IF      OPEN[Reparatur R,  'TRUE', 'FALSE'] OR
        OPEN[Pause P,  'FALSE', 'TRUE'] OR
        OPEN[Arbeitszeit A,  'FALSE', 'TRUE']
DO      ADDBEFORE[THIS, R, MessageType := Reparaturende];
        ADDBEFORE[THIS, P, MessageType := Pausenende];
        MODIFY[A, Time := THIS.Time];
```

Das hier betrachtete Regelereignis ist die Meldung „Stillstandsende“. Eine Stillstandsende-Meldung bedeutet, dass ein Fahrzeug durch einen Mitarbeiter in Bewegung gesetzt wurde. Bei Eintreffen einer solchen Meldung wird mit Hilfe des OPEN-Konstrukts überprüft, ob ein offenes Reparaturintervall für das Fahrzeug und/oder offene Pause- und Arbeitszeitintervalle für den zugehörigen Mitarbeiter existieren.[8] Wenn dies der Fall ist, liegen pathologische Zustände vor, weil das in Bewegung befindliche Fahrzeug nicht in der Reparatur, und der das Fahrzeug steuernde Mitarbeiter nicht in der Pause sein kann. Um diese Inkonsistenzen zu beseitigen, werden die zugehörigen Intervalle durch das Einfügen passender Intervallende-Meldungen (Reparaturende bei Reparatur und Pausenende bei Pause) im Aktionsteil zum Abschluss gebracht. Die Existenz eines offenen Arbeitszeitintervalls für den zum Meldungsereignis gehörenden Mitarbeiter ist im Prinzip korrekt, wird in dieser Regel aber dennoch abgefangen, um den Beginn des Arbeitsintervalls an den Beginn der Fahrzeugnutzung anpassen zu können. Hierdurch soll sichergestellt werden, dass Arbeitszeitintervalle von Mitarbeitern, die mit der Bewegung von Fahrzeugen verbunden sind, zeitlich mit den zugehörigen Fahrzeugintervallen übereinstimmen. Über den MODIFY-Befehl wird daher der Zeitpunkt der letzten Arbeitsbeginn-Meldung des Mitarbeiters auf den Zeitpunkt der aktuellen Stillstandsendemeldung (gespeichert in der THIS-Variable) gesetzt.

4 Resümee und Ausblick

Der vorliegende Beitrag stellt ein Framework zur Verarbeitung von Telematik-Daten von mobilen Objekten vor und zeigt die Anwendung an einem Beispiel aus dem Entsorgungsgewerbe. Das zentrale Konzept zur Konsistenzsicherung ist dabei die Sprache zur Abbildung von Konsistenzregeln.

[8] Der OPEN-Befehl enthält als zweiten und dritten Parameter bool’sche Werte, mit deren Hilfe die Suche nach offenen Intervallen des im ersten Parameter angegebenen Typs auf solche beschränkt werden kann, die zu dem gleichen Fahrzeug (2. Parameter = TRUE) bzw. zu dem gleichen Mitarbeiter (3. Parameter = TRUE) gehören wie das Regelereignis.

Es hat sich gezeigt, dass mit diesem Framework die Anwendungen unterschiedlicher Unternehmen gut gehandhabt werden konnten. Durch die flexible Definition der Semantik von Telematik-Daten können Sonderfälle und unternehmensindividuelle Anforderungen leicht abgebildet werden. Dies soll in zukünftigen Arbeiten näher untersucht werden. Darüber hinaus ergeben sich vielfältige weitere Untersuchungsmöglichkeiten: Mit der hier vorgestellten Sprache lassen sich Konsistenzregeln zwar flexibel abbilden, allerdings nicht in einer für Endbenutzer leicht verständlichen Form. Es bietet sich an, dem Benutzer ein mächtigeres Werkzeug an die Hand zu geben, mithilfe dessen er Regeln intuitiv an Beispielen, ähnlich Query By Example, definieren kann. Diese Werkzeuge zur Unterstützung der Regeldefinition befinden sich in der Entwicklung. Der Schwerpunkt zukünftiger Arbeiten stellt jedoch die Schaffung einer Entwicklungsmethodologie für individuelle betriebswirtschaftliche Telematik-Anwendungen dar.

Es ist zu erwarten, dass die Schaffung dieses generischen Systems das Potenzial besitzt, einen Beitrag zur Lösung der bestehenden Herausforderungen zu leisten und so die Nutzbarmachung von Telematik-Daten ein Stück voranbringt.

Appendix: Syntax der Regelsprache

Reservierte Worte

ADD	Meldung hinzufügen
ADDBEFORE	Meldung direkt vor anderer Meldung hinzufügen
ADDAFTER	Meldung direkt nach anderer Meldung hinzufügen
DO	Definitionsbeginn Regelaktion
EVENT	Regelereignis
FALSE	Boolescher Wert (falsch)
FIRST	Nachrichtenaccessor
FOREACH	Schleife (und Nachrichtenaccessor)
IF	Definitionsbeginn Regelbedingung
LAST	Nachrichtenaccessor
MODIFY	Meldung ändern
NEW	Nachrichtenkonstruktor
NULL	<nicht definierter Wert>
ON	Definitionsbeginn
OPEN	Nachrichtenaccessor
REMOVE	Meldung entfernen
ROUND	kaufmännisches runden (Zahlen und Datum)
SQL	Beginn SQL-Anweisung
THIS	Meldung aus Regelereignis (falls vorhanden)
TRUE	Boolescher Wert (wahr)

Operatoren

( )	Klammeroperator für Ausdrücke
[ ]	Klammer für Parameterliste
'	Kennzeichnung von Wertoperanden
NOT, AND, OR	Logische Operatoren
<, >, <=, >=, =	Vergleichsoperatoren
+, -	Arithmetische Operatoren
.	Zugriffsoperator
;	Anweisungstrenner (für Aktionsteil)
,	Parametertrenner
:=	Zuweisungsoperator
//	Kommentar (alles bis Zeilenende)

Syntax

Definition von Regeln

ON <Ereignisbezeichner> [,…]
[IF <Regelbedingung>]
DO <Regelaktion>

Definition von Regelbedingungen:

<Regelbedingung>: <Boolescher Ausdruck>
<Boolescher Ausdruck>: <Ausdruck> mit Ergebnis aus [TRUE, FALSE]
<Ausdruck>: Algebraischer Ausdruck aus <Operanden> und <Ausdrucksoperatoren>
<Ausdrucksoperatoren>: +, -, (,), NOT, AND, OR, <, >, <=, >=, =
<Operand>: <Wert-Operand> | <Attribut-Operand> | <Meldung> | <SQL-Operand>|
<ROUND-Operand>
<Wert- Operand >: '<Zeitpunkt>' | '<Zeitspanne>' | '<Zahl>' | '<Zeichenkette>' | 'TRUE'
|'FALSE' | 'NULL' | <Meldungstyp>
<Attribut-Operand >: <Meldung>.<Attributname> | EVENT.<Attributname>
<Meldung>: <Meldungsaccessor> | THIS | NEW(<Meldungstyp>) | <Instanzbezeichner>
<Meldungsaccessor>: {FIRST | LAST | OPEN}[[<Meldungs- od. Intervalltyp>]
<Instanzbezeichner>, <Intervallbeginn>, <Intervallende>, [<Regelbedingung>]]
Schnelles OPEN: [[<Meldungs- od. Intervalltyp>] <Instanzbezeichner>, <Zahl: vehicleId>,
<Zahl: driverId>]
<Intervallbeginn / -ende>: <Meldung> | <Zeit-Ausdruck>
<Zeit-Ausdruck>: <Ausdruck> mit Ergebnis: <Zeitpunkt>
<SQL-Operand>: SQL [<SQL-Statement>]
<ROUND-Operand>: ROUND[{<Zeit-Ausdruck>, <Zeitspanne>} | {<Zahl-Ausdruck>,
<Ganzzahl>}]
<Zahl-Ausdruck>: <Ausdruck> mit Ergebnis: <Zahl>

Definition von Regelaktionen:

<Regelaktion>: {{<FOREACH-Schleife> <Meldungsoperation> […]} |
<Meldungsoperation>} [;…]

<FOREACH-Schleife>: FOREACH[[<Meldungstyp>] <Instanzbezeichner>,
<Intervallbeginn>, <Intervallende>, [<Regelbedingung>]]
<Meldungsoperation>: <Meldung hinzufügen> | <Meldung ändern> | <Meldung entfernen>
<Meldung hinzufügen>: {ADD [<Meldung>, {<Attributname> := <Ausdruck>} [,…]]} |
{ADDBEFORE | ADDAFTER [<(Referenz-)Meldung>, <(Einfüge-)Meldung>,
{<Attributname> := <Ausdruck>} [,…]]}
<Meldung ändern>: MODIFY [<Meldung>, {<Attributname> := < Ausdruck >} [,…]]
<Meldung entfernen>: REMOVE [<Meldung>]

Literaturverzeichnis

[An04] *Andres, M.:* FTK Forschungsinstitut für Telekommunikation, Telematiksysteme für die eLogistik, Anwendungsbereiche, Lösungen, Marktüberblick, Dortmund: 2004

[DiGa00] *Dittrich, K. R. und Gatziu, S.:* Aktive Datenbanksysteme: Konzepte und Mechanismen, Heidelberg: dpunkt.verlag, 2000.

[HäPeSt00] *Häckelmann, H.; Petzold, H. J.; Strahringer, S.:* Kommunikationssysteme: Technik und Anwendungen, Berlin: Springer, 2000

[SpCa82] *Sprague, R. H. and Carlson, E. D:* Building Effective Decision Support Systems, Englewood Clifts, N.J.: Prentice-Hall, Inc., 1982

[TuPo04] *Turowski, K, Pousttchi, K.:* Mobile Commerce: Grundlagen und Techniken, Berlin: Springer 2004

Towards Scalable and Efficient Processing of Probabilistic Spatial Queries in Mobile Ad Hoc and Sensor Networks

Dominique Dudkowski*, Tobias Drosdol*, and Pedro José Marrón
Universität Stuttgart, Institut für Parallele und Verteilte Systeme (IPVS)
{dudkowski|drosdol|marron}@informatik.uni-stuttgart.de

Abstract: With the proliferation of sensor technology and advances in wireless communication, gathering, processing, and querying context information in mobile ad hoc and sensor networks becomes attractive and feasible. To support context-aware applications in such networks, efficient processing techniques for frequently used functionality, such as query and event management, are highly beneficial. In this paper, we discuss the challenges in efficient processing of spatial queries in mobile ad hoc and sensor networks. These comprise advanced query semantics based on inaccurate position information, efficient protocols and algorithms for data storage and query resolution, scalability with respect to network size, and support for mobile network nodes. We outline our concepts for solving these issues and show how they can be implemented in a suitable software architecture. Using current evaluation results, we show that our prototype implementation fulfills the stated challenges.

1 Introduction

The integration of sensor technology into mobile communication devices allows capturing the dynamic state of objects and phenomena in the physical world. Large amounts of context information become available allowing the realization of context-aware applications. Because various functionality, such as query processing, event management, or road navigation, is often used by such applications, their repeated implementation becomes a tedious task. An underlying data management facility that provides such fundamental services greatly simplifies application design. For location-aware applications in particular, *spatial queries*, such as range and k-nearest neighbor queries, are frequently used to retrieve data items by referencing to their location. They may be used, for example, to retrieve all taxis in a particular region or the nearest tools to one's own position in a factory building. While our Nexus platform ([Gr05]) already supports spatial queries on a large scale for infrastructure-based networks, an adequate query management facility does not exist for mobile ad hoc and sensor networks (MASNs). In such networks, nodes have integrated sensing, communication, and storage capabilities, but have to manage acquired data autonomously in a highly distributed manner. Due to limited resources of network nodes concerning energy, computing power, memory, and communication bandwidth, designing efficient strategies for query processing poses new challenges.

*funded by the German Research Foundation (DFG) within the Center of Excellence (SFB) 627.

The focus of this paper is to discuss the major challenges in supporting spatial queries in MASNs and to give an overview of our concepts for solving these issues. We do not intend to describe algorithmic details, which we address in other work. Rather than that, our aim is to give new impulses to context management issues of comparable complexity in related fields of context-aware systems for mobile networks.

The rest of our paper is structured as follows: In Sec. 2 we describe the system model that we use in the following discussions. Sec. 3 identifies the challenges related to query management in MASNs and Sec. 4 discusses related work. Our approach to tackle the stated problems and the integration of our concepts into a software architecture are described in Sec. 5 and 6, respectively. We discuss current evaluation results in Sec. 7 and conclude our paper in Sec. 8 with a summary and implications for future work.

2 System Model

The system model comprises a MASN operating in a service area that we denote by A. Two types of entities are located inside of A. *Perceivable objects* (PO_j) are an abstraction for physical objects, persons, or services that may be observed by sensor nodes. They may move by self-propulsion (persons, robots) or by being attached to carriers (persons, carts, conveyors). Mobile *sensor nodes* (SN_i) are part of the MASN and they may assume different roles (observers, data servers and clients), possibly multiple ones at a time. *Observers* are sensor nodes that capture the state of perceivable objects in their vicinity. They assemble a corresponding data object o_j consisting of the object identifier, object type, and observed location of the corresponding perceivable object, as well as the observation time. The observed location may be estimated, for example, from the observer's own position and the interpolated distance between the observer and the perceivable object. *Data servers* (DS_k) store a subset of data objects. Finally, *clients* provide access to the observed data by issuing spatial queries. Our goal is to implement spatial queries based on the given system model using only ad hoc communication between sensor nodes. Next, we discuss the challenges that arise from sensor data acquisition to the assembly of the final query result.

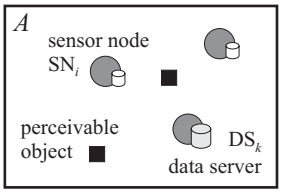


Figure 1: System model.

3 Challenges

The characteristics of MASNs lead to three principal challenges that must be solved with respect to the realization of a general spatial query management facility. First, *algorithm scalability* is essential to assure operation in networks with increasing size. Second, *node mobility* leads to the effect of moving data servers that must be compensated to retain the integrity of the data store. That is, it must be guaranteed that all objects relevant for query

resolution can still be found. Third, we must support advanced *query semantics* to account for inaccurate position information obtained from sensor observations.

With increasing size of the operated service area and number of sensor nodes, maintaining efficient operation of the overall query management facility is essential. Appropriate update propagation and storage protocols for acquired sensor data and strategies for aggregating query results from a highly distributed object store are required. Spatial queries are inherently suitable in that they support the design of a scalable solution. Because they access data based on location information, a geometric index can be designed that has a natural spatial relation to the coordinate system of the service area. In the case of range queries, it is intuitively suitable to store data near the specified range. The same is true for k-nearest neighbor queries, where objects nearest to a particular position are requested. Storing data close to its origin consequently allows to confine the resolution of a spatial query to a limited geometric region and provides the necessary scalability. Different mechanisms that provide a tradeoff between update propagation and query aggregation costs as well as the granularity by which the data model is partitioned and replicated across sensor nodes must be evaluated.

The increasing relevance of mobility in wireless network scenarios is a fundamental new issue when conceiving a general query management platform. Because data from sensor observations is stored directly on particular sensor nodes, data travels along with them and eventually loses its spatial coherence. If no countermeasures are taken, the communication overhead for finding particular data items increases constantly and eventually leads to network congestions and to the depletion of the nodes' energy. Relocation algorithms are required to "keep data in place" with minimum overhead in terms of communication costs and latency to assure continuous data availability for the aggregation of query results.

Physical processes involved in every sensor measurement inevitably lead to the acquisition of inaccurate sensor data. In the case of spatial data, the exact position of mobile objects is generally unavailable. In addition, limited update rates and the communication skew on each hop in a MASN aggregate to even greater inaccuracies. As a consequence, expressing the position of an object solely by using point coordinates is an inadequate simplification. In the case of a range query, the inclusion relation used to determine if the object is inside of the specified range is limited to a point inclusion test. However, due to localization errors, the object may be found in a particular *location area* that might only partially overlap with the given range. A threshold decision may then be much more appropriate to decide whether the object is considered to be inside or outside of the range. If the position data of individual objects is too inaccurate, some applications may even want to completely exclude these objects from the query result. The impact of inaccuracy on the outcome of a query might be even stronger if inhomogeneous position distributions are taken into account. The natural fact of position inaccuracies must be addressed by advanced query semantics, leading to *probabilistic spatial queries* that combine location areas with corresponding probability density functions.

4 Related Work

We classify existing work in the field of spatial query management into three categories: location and query semantics, algorithms for spatial queries in *stationary* sensor networks, location management and content location in mobile ad hoc networks.

In the first category, different query semantics were proposed based on interval calculations. The authors of [CP03] and [Ch04] introduce the notion of probabilistic threshold queries. In [CKP03] a classification of probabilistic queries is given, including definitions of probabilistic range and nearest neighbor queries. While the authors do not define spatial queries in more than one dimension, their concepts constitute a theoretical basis for the definition of spatial query semantics. Our work supports advanced query semantics for range and k-nearest neighbor queries based on inaccurate locations that also allow for excluding objects from query evaluation that do not possess a minimum accuracy.

In the second category, the Distributed Index for Features in Sensor Networks (DIFS, [Gr03]) and the Distributed Index for Multidimensional Range Queries in Sensor Networks (DIM, [Li03]) implement range queries in sensor networks for one and more dimensions, respectively. Both approaches apply localized storage for the geometric domain, which is beneficial for network scalability and query efficiency. Further, the Peer-Tree proposed in [DF03] was developed based on a centralized R-Tree that is used, e.g., in [SR01]. It is suitable for different types of spatial queries, in particular, for nearest neighbor queries. Most of these approaches concentrate on a single query type, and only the Peer-Tree may potentially be used for several types of spatial queries. Further, mobility is not addressed by any of the work, since all approaches are tailored to stationary networks. Last, only point coordinates are supported in the resolution of a spatial query, which does not allow to use more complex query semantics based on inaccurate positions.

In the third category, work that explicitly considers mobile nodes can be subdivided according to two purposes: location management and content location. Location management contains various approaches to query the position of network nodes for the purpose of geometric routing. Representative work includes the Grid Location Service (GLS, [Li00]) and the dead-reckoning-based location service proposed in [KD04]. Recent work in content location in mobile ad hoc networks is provided in [SH04] and [TV04]. These approaches allow to locate content by ID and have no support for spatial queries. As a consequence, they require a very different type of index structure with no relation to one required for spatial queries. Their work can be viewed supplementary to ours to provide a full-featured location service supporting position queries in addition to spatial queries.

To the best of our knowledge, all previous contributions do not address the challenges discussed in Sec. 3 in their combination. While spatial queries are implemented in the second category, node mobility is not considered. The third category supports mobility, but does not implement spatial queries. Finally, all approaches use just point coordinates and do not consider probabilistic query semantics. We will now address each of the challenges and show how they are solved in our architectural framework in Sec. 6.

5 Approach

5.1 Data Placement

We begin by defining a data placement strategy that is the basis for localized storage and constitutes the primary element for network scalability. Next, we show how data storage makes explicit use of that strategy during the observation of perceivable objects.

We recall from Sec. 2 that an observer captures the state of a perceivable object PO_j in form of a data object o_j. Each o_j contains the object identifier $o_j.\mathrm{id}$, object type $o_j.\mathrm{type}$, observed location $o_j.\mathrm{loc}$, and observation time $o_j.\mathrm{time}$. The observed location is composed of a location area and location probability density function (location pdf). The object is known to be located somewhere within this location area, and the location pdf defines the probability of being located in any subset of that area. Next, we subdivide the service area A into disjunct data sectors (Figure 2.a) and define two associations. First, we associate one data server DS_k that is located inside of a data sector S_r with that sector. Second, we associate data object o_j with each data sector S_r that overlaps with $o_j.\mathrm{loc}$. Note that multiple associations for the same object are possible (Figure 2.b). Finally, we define that o_j is mapped to data server DS_k iff both DS_k and o_j are associated with S_r.

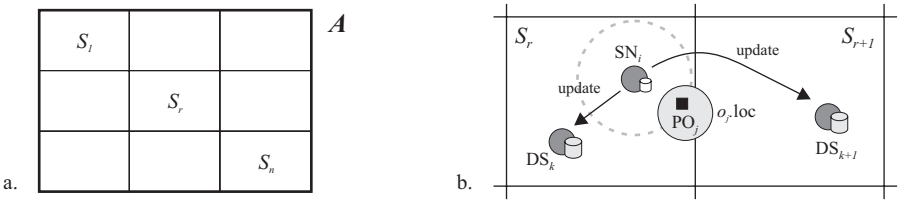


Figure 2: Subdivision of service area and placement of data objects.

In the first step of the data storage protocol, an observer constructs a data object o_j of the form $(o_j.\mathrm{id}, o_j.\mathrm{type}, o_j.\mathrm{loc}, o_j.\mathrm{time})$ that encapsulates the state of the corresponding perceivable object. Next, all data sectors are determined that overlap with the location area $o_j.\mathrm{loc}$. In Figure 2.b, $o_j.\mathrm{loc}$ overlaps with the data sectors S_r and S_{r+1}. According to the associations defined previously, copies of the data object are sent to data servers DS_k and DS_{k+1}. For that, a two-step routing is applied. In the first step, geometric routing, such as Greedy Perimeter Stateless Routing (GPSR, [KK00]), delivers the data object to a node inside of the respective data sector. Then, a local routing strategy is applied to finally reach the data server associated with that sector. At each data server, the object is inserted into the local database if a previous copy does not exist or its observation time is more current[1] than that of the existing copy. Note that each data object is annotated with a lifetime (counting from the observation time) and considered stale once its lifetime has elapsed. This way, copies that are not updated anymore (e.g. because the corresponding perceivable object is no longer detected by any observer) will eventually be deleted.

[1] We assume that sufficiently synchronized clocks exist, e.g., provided by GPS or a synchronization algorithm for mobile networks ([Röm01]).

5.2 Data Relocation

To maintain an efficient mapping between data servers and sectors, we use *data relocation* to move the data of an old server $\mathrm{DS}_{\mathrm{old}}$ to a new server $\mathrm{DS}_{\mathrm{new}}$. A *changeover condition* decides when relocation is initiated. A good strategy is to combine spatial and temporal predicates. If a server moves too far away from its associated data sector, the condition should trigger to limit the overhead for update and query routing. If a server remains nearly stationary inside of its associated sector, a timeout should trigger to avoid that single sensor nodes keep their server role for a long time and their energy is exhausted. In the second step, an election algorithm is used to determine a suitable sensor node that is eligible to become $\mathrm{DS}_{\mathrm{new}}$. For example, we may elect the node closest to the center of the corresponding sector, or the node that is expected to remain inside of that sector for a maximum time. In the final step, the contents of the database of $\mathrm{DS}_{\mathrm{old}}$ are transferred to $\mathrm{DS}_{\mathrm{new}}$. Our strategy transfers only those data objects that we do not consider stale, that is, whose lifetime has not yet elapsed. After relocation is complete, $\mathrm{DS}_{\mathrm{new}}$ takes over and thus guarantees storage locality.

5.3 Query Resolution Algorithms

We now describe two algorithms to resolve *probabilistic* range and k-nearest neighbor queries. Both algorithms involve an initial step in which the query is sent from the client to a *proxy node* that resolves the query on behalf of the client. For range and k-nearest neighbor queries, the node located closest to the center of the query range R and the query target position p_{NN} respectively, is an appropriate choice. Using GPSR, this node may be reached using greedy forwarding first, and then selecting the node visited twice during perimeter mode, which is guaranteed to be the closest node to the respective position.

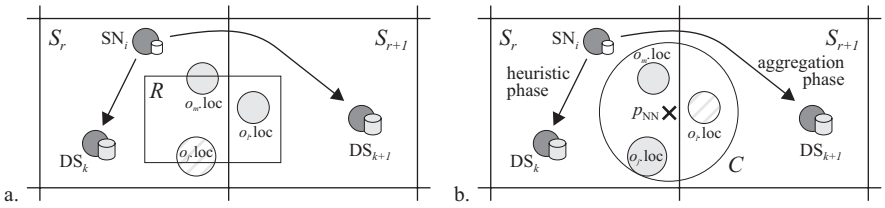


Figure 3: Query algorithms for range and k-nearest neighbor query resolution.

Probabilistic range queries (*p*RQs) for a geometric range R are resolved at the proxy node by aggregating the result from a number of *partial p*RQs. In the beginning, the proxy determines the set of data sectors overlapping with R (Figure 3.a). Each data server associated with one of these sectors is queried using a partial *p*RQ. Similar to data storage, routing to the data sectors is performed in two steps. At each data server, the partial *p*RQ is resolved locally. The algorithms for local resolution of partial *p*RQs depend on the precise query semantics and are out of the scope of this paper. Each partial result is returned to

the proxy that aggregates it into the final result. A timeout at the proxy terminates result aggregation if not all partial *p*RQs succeed. Finally, the proxy sends the result to the client.

The resolution of probabilistic k-nearest neighbor queries (*p*NQs) at the proxy node involves two phases: the *heuristic phase* finds initial nearest neighbor candidates, and the *aggregation phase* creates the final result. During the heuristic phase, the goal is to find k candidate objects that are close to the actual nearest neighbors. This is done by issuing a partial *p*NQ in the data sector that contains the query position p_{NN} (Figure 3.b). If no k candidates are returned, circumjacent data sectors are queried with increasing distance to p_{NN} until k candidates are available. From these candidates, our algorithm constructs the circle denoted by C with center p_{NN} that completely contains the location areas of the candidates. This property guarantees that all other objects that might be nearer to p_{NN} than any of the candidates are considered. During the aggregation phase, partial *p*NQs are sent to all data sectors that overlap C. At each data server, a partial *p*NQ is resolved that determines nearest neighbors from the objects stored in the server's local database. Up to k objects are returned to the proxy for each partial *p*NQ. The proxy aggregates received objects into the final query result. Again, a timeout terminates the aggregation if any partial result cannot be determined. Finally, the query result is returned to the client.

6 Software Architecture

Figure 4 shows the software architecture of our data management framework that implements the protocols and algorithms described in the previous section. The architecture is composed of two main parts: *routing* and *data management*. Routing comprises two layers with increasing level of abstraction with respect to the specification of routing goals. The *Routing Executive Layer* (REL) is responsible for carrying out low-level routing. It implements variations of geometric routing and extends GPSR ([KK00]). It is able to route a message to the node nearest to a geometric region (Location Routing Module) or to a specific node (ID Routing Module). It implements scoped flooding used to announce the location of database nodes, where the information is in turn used by the ID Routing Module. The *Routing Convergence Layer* (RCL) interfaces with the REL and data management and realizes complex routing goals. It provides the two-step routing used for data storage and query resolution by invoking the RCL multiple times.

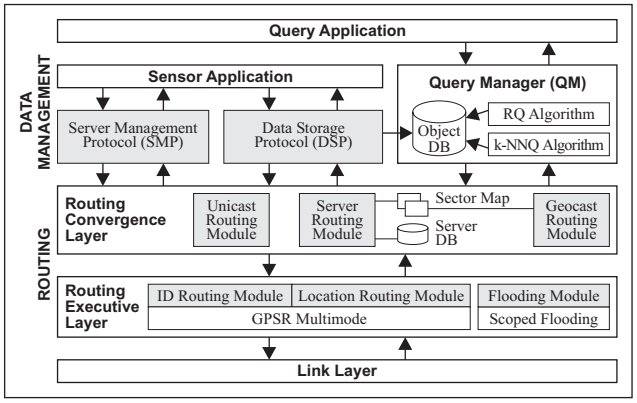


Figure 4: Software Architecture.

Data management implements the protocols and algorithms for data placement, data relocation, and query resolution. The *Data Storage Protocol* (DSP) implements the functionality described in Sec. 5.1. The DSP invokes the RCL possibly multiple times to route copies of data objects to the data servers to which the object maps, and updates each copy in the local database (Object DB) of these servers. The *Server Management Protocol* (SMP) performs data relocation as described in Sec. 5.2. Finally, the *Query Manager* (QM) integrates implementations of query algorithms, like the range and k-nearest neighbor query algorithm. It is modular and allows to add and remove implementations as needed. Each query algorithm can be invoked by applications and performs the resolution of a particular query instance by invoking the RCL each time a partial query is resolved.

7 Performance Analysis

In this section we state quantitative results to discuss the performance of our architecture. We have conducted our experiments in the ns-2 simulation environment using a service area of 400 $\times$ 400 m. We deployed 80 sensor nodes and 160 perceivable objects in the service area. Nodes and objects move according to the random waypoint mobility model with a fixed default speed of 1.5 m/s and 30 seconds of pause time. The transmission range of nodes is 150 m. We assume that a node is able to detect a perceivable object while its physical distance to the object is less than or equal to 20 m. Further, the location computed during an observation is assumed to be inaccurate by a disc with a radius of 10 m.

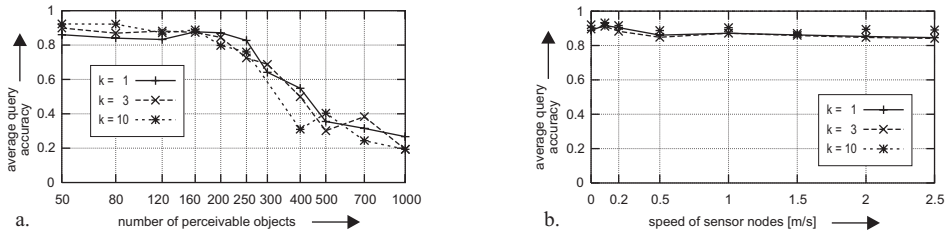


Figure 5: Average query accuracy for k-nearest neighbor queries and different values of k.

Figure 5.a shows the average *query accuracy* in relation to the number of perceivable objects for k-nearest neighbor queries. Query accuracy is the fraction of the objects returned by our algorithm that would also be returned from a global data structure containing the most recently observed location of all perceivable objects. With increasing number of these objects, the average distance between them decreases. Thus, there are more objects with a similar distance to the query reference position p_{NN}. As a consequence, ordering objects by that distance is much more influenced by inaccurate position information. This leads to a decrease in query accuracy with a growing number of objects, which may only be compensated by higher observation rates or additional location fusion strategies (which is out of the scope of this paper). Figure 5.b shows query accuracy as a function of node speed. The results confirm that mobility has virtually no effect on query accuracy for typical pedestrian speeds of up to 2.5 m/s.

Figure 6 shows results of various metrics as a function of the number of data sectors for k-nearest neighbor queries. Minimal values of the metrics, which indicate optimal performance of the algorithms, are found for small numbers of sectors. Average *query latency*, defined as the time necessary from issuing the query to returning the result to the client, is shown in Figure 6.a. It increases with the number of data sectors because more partial queries are required to resolve a nearest neighbor query. Since we do not send partial queries redundantly, the probability increases that at least one of them fails due to unresolved routing. This leads to an increasing number of cases where query timeouts occur at the proxy and thus, average query latency increases.

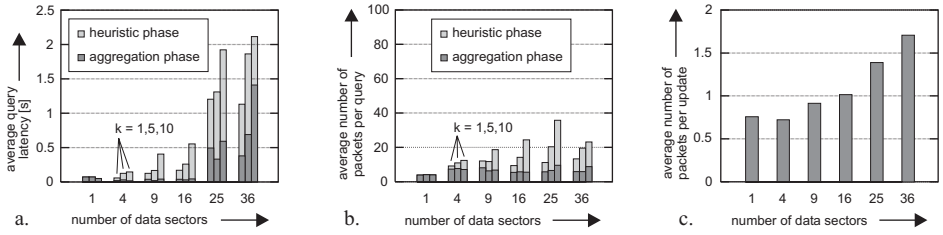


Figure 6: Average query latency, query costs, and update costs.

Figure 6.b shows the average *query costs*, defined as the total number of packets required to resolve a nearest neighbor query. Costs increases with the number of data sectors, because more partial queries are required for a larger number of data sectors. Note that with increasing k, query latency and query costs in Figure 6.a and 6.b increase because more partial queries are required. This is a direct consequence from the circle C that is determined according to Sec. 5.3, whose radius increases and which overlaps more data sectors with larger values of k. Figure 6.c shows the *update costs* for increasing number of data sectors, that is, the number of packets required to update a data object at potentially multiple servers. In the case of 4 sectors, update costs are optimal. For more data sectors, update costs increase because the location area computed by the data storage protocol according to Sec. 5.1 overlaps with more data sectors. Note that the average number of packets may be less than 1, since some observations are done by a data server that is responsible for storing the data object it has created.

8 Conclusion

In this paper, we discussed the challenges for efficient and scalable processing of spatial queries in mobile ad hoc and sensor networks. We have described our general approach that solves the problems of algorithm scalability, node mobility, and query semantics by exploiting localized data storage and efficient data relocation while taking into account general query semantics defined on location areas and probability density functions. We presented our software architecture that incorporates the algorithms into a general framework for spatial query processing in mobile ad hoc and sensor networks. Finally, we showed by quantitative analysis that our architecture achieves the stated goals.

Future work includes extension and optimization aspects. Extensions concern the inclusion of a topologic model to account for restricted movement of objects, for example, in urban areas. A more complex static data model requires partitioning approaches to consider the limited storage capabilities of sensor nodes. Further, the incorporation of position queries must be accomplished to provide all essential queries used in location-aware applications. Optimizations are to be done to packet aggregation on the update side, which we expect will tremendously decrease network traffic. On the query side, we see the potential for many optimizations when considering different timeout strategies together with partial query planning based on object densities in different parts of the service area. Further, redundant sending of partial queries may be used to increase query accuracy. Last, we have already designed a suitable cleanup protocol that increases consistency of multiple copies corresponding to the same perceivable object.

References

[CKP03] Cheng, Kalashnikov, Prabhakar: Evaluating Probabilistic Queries over Imprecise Data. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pp. 551–562, San Diego, USA, 2003.

[CP03] Cheng, Prabhakar: Managing Uncertainty in Sensor Databases. *ACM SIGMOD Record. Special Issue: Special Section on Sensor Network Technology & Sensor Data Managment*, 32(4):41–46, 2003.

[Ch04] Cheng et al.: Efficient Indexing Methods for Probabilistic Threshold Queries over Uncertain Data. In *Proc. 30th VLDB Conf.*, pp. 876–887, Toronto, Canada, 2004.

[DF03] Demirbas, Ferhatosmanoglu: Peer-to-Peer Spatial Queries in Sensor Networks. In *Proc. 3rd Int. Conf. on Peer-to-Peer Computing*, pp. 32–39, Linköping, Sweden, 2003.

[Gr03] Greenstein et al.: DIFS: A Distributed Index for Features in Sensor Networks. In *Proc. 1st IEEE Int. Workshop on Sensor Network Protocols & Applications*, pp. 163–173, Anchorage, USA, 2003.

[Gr05] Großmann et al.: Efficiently Managing Context Information for Large-scale Scenarios. In *Proc. 3rd IEEE Conf. on Pervasive Computing & Comm.*, Kauai Island, USA, 2005.

[KD04] Kumar, Das: Performance of Dead-Reckoning-Based Location Service for Mobile Ad Hoc Networks. *Wireless Comm. & Mobile Computing*, 4(2):189–202, 2004.

[KK00] Karp, Kung: GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proc. 6th Int. Conf. on Mobile Computing & Networking*, pp. 243–254, Boston, USA, 2000.

[Li00] Li et al.: A Scalable Location Service for Geographic Ad Hoc Routing. In *Proc. 6th Int. Conf. on Mobile Computing & Networking*, pp. 120–130, Boston, USA, 2000.

[Li03] Li et al.: Multi-Dimensional Range Queries in Sensor Networks. In *Proc. ACM Conf. on Embedded Networked Sensor Systems*, Los Angeles, USA, 2003.

[Röm01] Römer: Time Synchronization in Ad Hoc Networks. In *Proc. 2nd ACM Int. Symp. on Mobile Ad Hoc Networking & Computing*, pp. 173–182, Long Beach, USA, 2001.

[SH04] Seada, Helmy: Rendezvous Regions: A Scalable Architecture for Service Location and Data-Centric Storage in Large-Scale Wireless Networks. In *Proc. 18th Int. Parallel & Distributed Processing Symp.*, p. 218a, Santa Fe, USA, 2004.

[SR01] Song, Roussopoulos: K-Nearest Neighbor Search for Moving Query Point. In *Proc. 7th Int. Symp. on Advances in Spatial & Temporal Databases*, pp. 79–96, Redondo Beach, USA, 2001.

[TV04] Tchakarov, Vaidya: Efficient Content Location in Wireless Ad Hoc Networks. In *Proc. IEEE Int. Conf. on Mobile Data Management*, pp. 74–85, Berkeley, USA, 2004.

Ein Blick in die Zukunft: Datenbankunterstützung für mobile AR Systeme[1]

Martin Breunig[1], Wolfgang Bär[1], Andreas Thomsen[1],
Alexandre Hering Coelho[2], Guido Staub[2], Sven Wursthorn[2]

[1]Forschungszentrum für Geoinformatik und Fernerkundung (FZG)
Universität Osnabrück
Eichendorffweg 30, D-49377 Vechta
E-mail: {vorname.nachname}@uni-osnabrueck.de

[2]Institut für Photogrammetrie und Fernerkundung (IPF)
Universität Karlsruhe
Englerstr. 7, D-76128 Karlsruhe
E-Mail: {nachname}@ipf.uni-karlsruhe.de

Abstract: Die Entwicklung von Datenbanksystemen und von mobilen Augmented Reality Systemen verläuft bisher orthogonal zueinander. Aus Datenbanksicht betrachtet sind jedoch Techniken der Augmented Reality (AR) für neue Benutzeroberflächen zur Visualisierung komplexer 3D-Objekte in Anfrageergebnissen sehr interessant. Neben der Medizin und den Biowissenschaften scheinen uns die Geotechnologien mit ihren inhärent raum- und zeitabhängigen Anwendungen im Gelände besonders geeignete Szenarien für zukünftige mobile, datenbankgestützte AR Systeme aufzuzeigen. Bei der Arbeit im Gelände sollte der Geowissenschaftler auf alle bisher erfassten Geodaten zugreifen, diese online ergänzen bzw. mit existierenden Modellen vergleichen und dabei auch 3D Strukturen unter der Erdoberfläche einbeziehen können. In diesem Beitrag wagen wir einen Blick in die Zukunft und beleuchten die benötigte Funktionalität mobiler, datenbankgestützter AR Systeme, welche die physische Umgebung durch modellbasierte Szenarien überlagern können. Dies könnte in Zukunft der präventiven Analyse möglicher Naturkatastrophen wie Hochwasser und Hangrutschungen dienen.

1 Das diesem Artikel zugrundeliegende Vorhaben „Weiterentwicklung von Geodiensten“ (http://www.geoservices.uni-osnabrueck.de) wird mit Mitteln des Bundesministeriums für Bildung und Forschung unter den Förderkennzeichen 03F0373B et al. gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

1 Einführung: AR und VR

Während Virtual Reality (VR) die reale Umgebung z.B. durch eine photorealistische Darstellung vollständig ersetzt, sieht der Benutzer bei Augmented Reality (Erweiterte Realität, kurz AR) eine Kombination von virtueller Realität und tatsächlicher Realität. Unter AR wird die Überlagerung von virtueller Information mit der physischen Umgebung („Realität“) in Echtzeit durch Nutzung transparenter Head Mounted Displays verstanden. Dabei soll die Information möglichst am korrekten geometrischen Ort dargestellt werden. Im Gegensatz zu VR ergänzt AR also die Realität und ersetzt sie nicht.

2 Bedarfsanalyse

Seit Jahren besteht ein großer Bedarf an der Entwicklung und Einbindung neuer, visuell unterstützender Benutzerschnittstellen für Datenbanksysteme [Ab03]. Der Bedarf an mobiler datenbankgestützter Augmented Reality ist heute schon in allen raumbezogenen Wissenschaften wie der Medizin, den Bio-, Geo- und Umweltwissenschaften und den Ingenieurdisziplinen vorhanden [FOR03] [Mü01]. Allerdings können derzeit existierende mobile Datenbanken [MS03] [Tü03] auf kleinen mobilen Endgeräten wie PDAs hierzu noch wenig beitragen. Anwendungsfelder sind überall dort zu sehen, wo jetzt schon mobile Geräte - z.B. zur Unterstützung von Planungsvorhaben, der Analyse geologischer Strukturen, der Sichtbarmachung von Leitungen im Untergrund, etc. - eingesetzt werden. Hier kann die AR-gestützte vor-Ort-Analyse eine entscheidende Verbesserung zu den bisher angewandten Verfahren bringen [Br03]:

- In der *Medizin* können mobile AR Systeme für die Operationsplanung, die Operation selbst oder die medizinische Ausbildung genutzt werden. Hierbei geht es um die Überlagerung präoperativ gewonnener virtueller Daten (CT-, MRT- oder Ultraschallaufnahmen) mit dem zu behandelnden Körperteil des Patienten. Somit können komplexe anatomische Zusammenhänge dargestellt werden [Su02].

- Bei der präventiven *Gefahrenanalyse für evtl. auftretende Naturkatastrophen* wie dem Hochwasser [WCS04] dient der Abgleich von aktuellen Messwerten im Gelände mit entfernten Datenbanken zum einen einer direkten Nutzbarmachung für Anwender, zum anderen können neue Messwerte direkt weiterverarbeitet und die Ergebnisse dieser Datenauswertung für die Feldarbeit genutzt werden. Hoch aktuelle Daten werden künftig jederzeit zur Verfügung stehen, um direkt abgerufen und wieder abgespeichert zu werden. Die Ergebnisse können direkt in die weitere Messpunktwahl mit einfließen, ohne dass der Erfassungsvorgang unterbrochen werden müsste.

- In der *Geologie* kommt es oft vor, dass während der geologischen Aufnahme aufgrund einer äußerst komplexen Geländemorphologie oder fehlender Geländeaufschlüsse Schwierigkeiten bei der Erfassung der genauen Lage und Ausprägung einzelner geologischer Schichten bestehen. Es bietet sich daher an, dass der Fachmann im Bedarfsfall sich vor Ort das entsprechende 3D-Untergrundmodell einblenden läßt und mit den sichtbaren Strukturen auf der Geländeoberfläche vergleichen kann.

- Weitere interessante Anwendungsbereiche sind Planung und Ausführung im *Bauwesen* und *Bergbau.* Beispielsweise ist es interessant, vor Ort eine „integrierte Sicht" von schon existierenden mit geplanten Bauwerken bereitgestellt zu bekommen oder den voraussichtlichen weiteren Flözverlauf beim Gang durch eine Grube zu visualisieren.

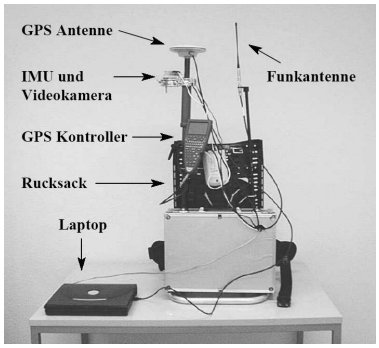
Den mobilen Datenbanken fehlt jedoch für den praktischen Einsatz in entsprechenden Anwendungsszenarien noch die notwendige Leistungsfähigkeit. Ausgehend von prognostizierten Entwicklungen wird daher im folgenden auf den Einsatz datenbankgestützter Augmented Reality in einem weiteren Zeithorizont von 10-20 Jahren eingegangen. Als Anwendungsbereich betrachten wir Geoanwendungen, da dort die Arbeit im Gelände den Einsatz leistungsfähiger mobiler datenbankgestützter AR Systeme rechtfertigt.

3 Typische Anwendungsszenarien

Die mobile Erfassung und das Update von Geodaten vor Ort, sowie die Visualisierung und Überlagerung von mit dem Auge nicht sichtbarer, von einem Datenbanksystem verwalteter Objekte im Untergrund spielen eine wesentliche Rolle in Geoanwendungsszenarien.

3.1 Anwendungsszenario „Hochwasser"

Hochwasser verursacht in städtischen Gebieten erhebliche Schäden. Die Hochwassergefahren können bereits im Vorfeld dadurch reduziert werden, dass à priori geeignete Techniken zur Visualisierung der Katastrophenszenarien und ihrer Analyse als ein Bestandteil eines Hochwassermanagement-Systems bereitgestellt werden. Als herkömmliche Visualisierungstechniken gelten Bilder, Karten, Geographische Informationssysteme und Virtual Reality. Diese Medien stellen die Wirklichkeit abstrakt dar. Im Gegensatz dazu bietet *Augmented Reality* die Möglichkeit, die natürliche Umgebung des Nutzers mit zusätzlichen, virtuellen Inhalten erweitern zu können, ohne vom Nutzer die Übertragung der virtuellen Information in seine reale Welt zu fordern.



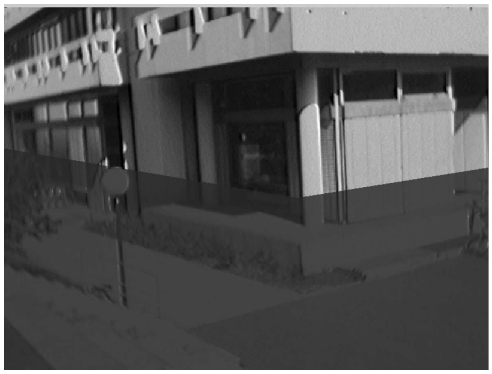
(a) Aufbau des tragbaren AR Systems

(b) Tragbares AR System im Feldversuch

Abbildung 1: Prototyp – Tragbarer Aufbau mit integrierten Sensoren

Abbildung 1 zeigt den Prototypen eines AR Systems für die Outdoornutzung [WCS04], mit dem virtuelle Wasserspiegel (Ergebnisse von Hochwassersimulationen des Instituts für Wasserwirtschaft und Kulturtechnik der Universität Karlsruhe) direkt in der „Realität“ visualisiert werden.

Der Benutzer des AR Systems kennzeichnet im Freien die Stellen, an denen mögliche Schäden im Fall einer realen Flut auftreten können. Die Szenen werden auf dem Monitor des Laptops sichtbar gemacht und nach Relevanz aufgenommen. In einer Datenbank werden für einzelne Gebäude die Aufnahmen zusammen mit Textinformation gespeichert. Beispiele von Szenen, die sich mit dem AR System generieren lassen, sind in Abbildung 2 dargestellt. Die virtuellen Wasserspiegel werden dabei durch transparente, blaue Flächen repräsentiert.



(a) Topographie aus Gebäudemodell – Horizontale Ebene als Wasserspiegel

(b) Topographie aus Laserscannerdaten – Simulierter Wasserspiegel

Abbildung 2: Beispiele von Aufnahmen – Wasserspiegel transparent dargestellt.

Abbildung 2a zeigt eine Szene, in der ein Wasserspiegel als grobe Näherung durch eine horizontale Ebene dargestellt wird. Ein größerer Realismus lässt sich durch die Anwendung eines Wasserspiegels aus einer Hochwassersimulationsrechnung erhalten (siehe Abbildung 2b). Eine statische Fläche als Wasserspiegel ist allerdings immer noch eine grobe und unrealistische Darstellung, da der Wasserspiegel bei einem realen Überschwemmungsszenario natürlich dynamisch ist. Die Gefahrenanalyse hängt sehr von dieser Dynamik ab. In der Zukunft sind zur Optimierung solcher Systeme dynamische Wasserspiegel zu nutzen. Die Visualisierung von Geschwindigkeitsvektoren wäre ebenfalls für eine Gefahrenanalyse von Bedeutung.

3.2 Anwendungsszenario „Hangrutschung“

Hangrutschungen können ebenfalls erhebliche Schäden verursachen. Auch wenn sie selbst nicht verhindert werden können, ist es möglich, Risikobereiche zu identifizieren und langfristig zu überwachen, um gegebenenfalls das gefährdete Gebiet zu räumen und zu sperren. Der Winkelgrat, ein markanter Vorsprung des Albtraufs südwestlich von Laufen im Zollernalbkreis (Baden-Württemberg) ist ein Beispiel für solch ein Risikogebiet. Hier werden seit vielen Jahren tiefreichende Felsbewegungen beobachtet, welche eine potentielle Gefährdung einer Kreisstraße darstellen [La02]. Seit einigen Jahren ist hier ein automatisches Überwachungssystem im Einsatz. Mit fünf hochempfindlichen Extensiometern (Abb. 3) werden die Bewegungen der Teilschollen kontinuierlich aufgezeichnet und an einen vor Ort installierten Rechner gemeldet. Dieser löst bei Überschreitung gewisser Grenzwerte einen Alarm aus, die Kreisstraße wird mittels zweier Ampeln gesperrt und der zuständige Geologe benachrichtigt. Nach Besichtigung vor Ort kann entweder die Sperrung aufgehoben werden, oder es wird entschieden, welche weiteren Maßnahmen zur Sicherung der Straße notwendig sind.

Neben den Extensiometermessungen sind weitere Daten, z.B. geodätische Festpunkte mit ihren Bewegungen, die Lage der Spalten und ein digitales Geländemodell [BV01] vorhanden. Aus dem Geländemodell sowie aus vom zuständigen Landesamt (LGRB) konstruierten Profilschnitten wird ein vereinfachtes 3D-Modell des Untergrundes erstellt (Abb. 4). Wegen der sehr unterschiedlichen Genauigkeit der Darstellung des weitmaschigen Geländemodells einerseits und der Extensiometermessungen andererseits werden die erfassten Bewegungen als diskretes zeitabhängiges Vektorfeld auf den Messpunkten dargestellt, aus welchem Lageänderungen der Messpunkte über ein längeres Zeitintervall abgeleitet werden können.

Abbildung 3: Extensiometer

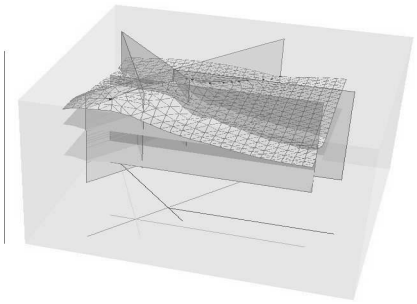


Abbildung 4: 3D-Modell des Untergrundes

Datenbankgestützte AR kann den Geowissenschaftler bei der Geländearbeit in vielfältiger Weise unterstützen. Durch die Einblendung der gespeicherten Daten können Messpunkte mit ihren Messwerten und verborgene Strukturen bereits früher erfasster Spalten sichtbar gemacht, sowie Ausbisslinien im Modell erfasster Verwerfungen an der Oberfläche angezeigt werden. Daneben steht dem Bearbeiter über die Datenbankanbindung die gesamte erfasste Information aus früheren Aufnahmen zur Verfügung. So kann AR eine Kartierhilfe bei Aufnahme neuer Elemente (Kleinmorphologie) darstellen. Der vor-Ort-Vergleich aktueller Informationen mit dem älteren Datenbestand kann zur Unterstützung der Entscheidungsfindung dienen. Darüber hinaus erlaubt AR auch den Vergleich von etwa mittels einer numerischen Modellrechnung erstellten Prognosen mit dem realen Ist-Zustand: „Wie sieht es aus, wenn sich die gemessenen Bewegungen über einen längeren Zeitraum fortsetzen?“.

4 Aus den Anwendungsszenarien abgeleitete Datenbank- und Dienstfunktionalität für künftige mobile AR Systeme

Im folgenden wird zuerst auf die zukünftige Funktionalität in AR Systemen aus der Sicht der Nutzer eingegangen (AR Clients). Diese Funktionalität muss dabei vollständig von Datenbanken unterstützt bzw. bereitgestellt werden. Auf die hierfür benötigte Datenbankfunktionalität wird anschließend eingegangen.

4.1 Funktionalität aus Nutzersicht[1]

Benutzersteuerung: Die Benutzersteuerung künftiger mobiler AR Systeme wird sich wesentlich ändern: Neben einer Sprachsteuerung für Datenbankanfragen wird die Gestik des Nutzers für die Steuerung verwendet werden können, beispielsweise wenn bestimmt werden soll, dass keine weiteren Szenen und somit wieder das Übersichtsbild eingeblendet werden soll. Außerdem wird in einem Minifenster zusätzlich zur Ansicht in der Vogelperspektive ein 2D Bild eingeblendet werden können, das wie eine Karte genutzt werden kann und bei Bedarf das Umschalten zwischen 2D und 3D Ansicht mit Nachladen der entsprechenden Daten aus der Datenbank ermöglicht. Eine für die Zukunft wichtige Eigenschaft von AR Systemen ist die ununterbrochene Nachführung der Szenen, die durch die Bewegungen des Nutzers und den damit verbundenen neuen Blickrichtungen entstehen (idealerweise in Echtzeit). Dabei wird beispielsweise die Position des Wasserspiegels im vorgestellten Hochwasserszenario kontinuierlich durch die Messungen der Sensoren für Positionierung und Lagebestimmung korrigiert. Diese Eigenschaft wird dann dem Benutzer des AR Systems Bewegungsfreiheit während der Arbeit erlauben. Er kann die Szenen aus den Blickwinkeln aus aufnehmen, die eine klare Darstellung der identifizierten Probleme ermöglichen.

Automatisches Einblenden von Daten: Eine wichtige Rolle in zukünftigen mobilen AR Systemen wird das Einblenden von „Metadaten“ zum gerade ins Visier genommenen Untersuchungsgebiet spielen. Hierunter sind vor allem Daten zur Begehungshistorie und zu Gefahrenpunkten gemeint. Von unschätzbarem Wert wird beispielsweise die eingeblendete Meldung sein: „Hier war Ihr Kollege *x* bereits vor drei Monaten“. Noch wichtiger im unwegsamen Gelände ist der *Hinweis auf Gefahrenbereiche* wie Felsspalten, die möglicherweise durch Laub oder sonstige Bodenbedeckung verdeckt sind. Weiterhin wird es Routine werden, dass für sich im Blickfeld befindliche Messeinrichtungen (Sensoren) deren *aktuelle Messwerte im Display* eingeblendet werden. Diese Einblendung muss je nach Nähe zur Messeinrichtung detaillierter erfolgen.

Einblendung von Strukturen: Der größte Mehrwert für den Experten im Gelände besteht darin, Untergrundstrukturen einzublenden, die Kontakt zur Erdoberfläche haben. Dadurch wird die sichtbare Geländeoberfläche ergänzt zu einem Gesamtbild aus Oberflächenstrukturen und unsichtbaren Strukturen des Untergrundes. Hier müssen auf jeden Fall auch Angaben zur *Qualität und Genauigkeit* der gerade eingeblendeten Daten berücksichtigt werden. Im Vordergrund einer Szene wird dabei eine weit höhere Genauigkeit erwartet als im Hintergrund. Vorstellbar ist in zukünftigen mobilen AR Systemen ebenfalls eine Umschaltung auf einen Virtual Reality (VR) Betrieb, in dem Untergrundstrukturen virtuell dargestellt werden, die mittels AR Techniken nicht vermittelbar sind.

[1] Die im folgenden aufgeführten Thesen wurden nach einer Befragung einiger potentieller Anwender des in Abbildung 1 vorgestellten AR Systems aufgestellt.

4.2 Datenbankfunktionalität für zukünftige mobile AR Systeme

Ähnlichkeitsbasierte Datenbanksuche: Die Aufnahme einzelner Punkte im Gelände mit dem Fadenkreuz wird der Vergangenheit angehören. Vielmehr werden aus der Struktur der Erdoberfläche automatisch Strukturen wie Bruchkanten, Grenzlinien oder Flächenabgrenzungen extrahiert. Für diese Strukturen wird nach Beziehungen zu bereits vorhandenen 3D Strukturen in der Datenbank gefahndet. Benötigt wird *eine Suche nach Ähnlichkeit von 3D Objekten* unabhängig von ihrer Skalierung bzw. dem verwendeten geometrischen Referenzsystem. Ein typisches Beispiel einer Datenbankanfrage ist: „Zu welcher Störung gehört die an der Oberfläche gefundene Bruchkante?“. Hier könnten lernbasierte Algorithmen, wie sie beispielsweise in der Robotik verwendet werden, eingesetzt werden. Wichtig ist es die Benutzereingabe möglichst zu minimieren, um den Geowissenschaftler im Gelände nicht unnötig abzulenken. Durch die Erkennung einer geometrischen Ähnlichkeit zwischen den extrahierten Strukturen der Oberfläche und den 3D Objekten des Untergrundes in der Datenbank kann der Geowissenschaftler die Qualität der 3D Modellbildung verbessern. Hierbei geht es insbesondere um die Konsistenz der 3D Geometrien und ihre topologischen Beziehungen untereinander. Die Modellbildung kann durch Referenzgeländeformen unterstützt werden, die in der Datenbank abgespeichert sind und abgefragt werden können. Außerdem können durch einen automatischen Vergleich räumlicher Strukturen auch Anomalien festgestellt werden, durch deren Erkennen und anschließender Korrektur Geowissenschaftler ihr Modell ebenfalls qualitativ verbessern können.

Verwaltung von Objekten in der Zeit: Der Geowissenschaftler interessiert sich natürlich nicht nur für den Ist-Zustand eines gegebenen Untersuchungsgebietes, sondern auch für dessen Historie und daraus ableitbarer Prognosen. Die Modellierung der Historie und die Verwaltung daraus resultierender, zeitabhängiger 3D Objekte des Untergrundes in Datenbanken vermitteln in Verbindung mit der Visualisierung durch AR Systeme ein besseres Verständnis geowissenschaftlicher Prozesse. Weiterhin geht es bei den zuständigen Landesämtern in hohem Maße um Einschätzungen der aktuellen und zukünftigen Lage, beispielsweise bei gefährdeten Hängen oder zu erwartenden Hangrutschungen. Hierfür können aufgrund numerischer bzw. statistischer Modell-rechnungen Prognosen in Form verschiedener Varianten *zeitabhängiger 3D-Objekte* erstellt und *von der Datenbank verwaltet* werden. Damit können komplexe Anfragen von der Datenbank beantwortet und visualisiert werden, wie zum Beispiel: „Wie verhält sich raum/zeitlich der obere Teil des Hanges, wenn der untere 50 m abgerutscht ist?“.

Unterstützung von Gruppenarbeit: In Zukunft werden AR Anwender sich im Gelände zu einem mobilen Ad-hoc Netzwerk (MANET) verbinden und in kooperierende Gruppen organisieren. Die *lokalen mobilen Datenbanken* müssen dazu in das *MANET integriert* sein und die lokalen Datenbestände den anderen Gruppenmitgliedern zugänglich machen. Die Interaktion zwischen den lokalen mobilen Datenbanken der AR Clients kann dabei zum einen als direkte Client/Server-Interaktion, vermittelt durch das MANET, erfolgen. Hierbei werden gezielte Anfragen auf der lokalen Datenbank eines anderen Clients ermöglicht. Andererseits sind automatisierte Mechanismen ohne eine Nutzerinteraktion möglich. Hierbei werden, geregelt z.B. über Publish/Subscribe-Verfahren, *Änderungen an lokalen Datenbeständen* eines Clients an alle anderen interessierten Mitglieder des MANETs *propagiert.*

Einbeziehung von Sensornetzwerken: Die derzeitige Entwicklungen im Bereich der Sensornetzwerke [HHM03] lassen einen häufigen Einsatz solcher kostengünstiger Monitoringsysteme für viele zukünftige Anwendungen in den Umweltwissenschaften erwarten. Bereits jetzt existieren vielversprechende Ansätze zur Integration von Datenbanktechniken zur Steuerung und Abfrage von Sensornetzen [De03][Ma04]. Zukünftige AR Systeme müssen in der Lage sein die im lokalen Umfeld erreichbaren *Sensornetzwerke* als Datenquelle *einbeziehen* zu können. Hierdurch kann die Verfügbarkeit aktuellster Daten ermöglicht werden, wie es selbst mit online Anfragen auf den zugehörigen Datenservern - die nur in Intervallen aktualisiert werden - nicht möglich ist. Werden Sensornetze beispielsweise zum Monitoring von Hangrutschungen eingesetzt, ergeben sich auch Änderungen der Lage der Sensoren und damit der Topologie des Sensornetzwerkes. Hierdurch können sich auch Messwerte und Beziehungen zwischen einst benachbarten Sensoren ändern. Solche *Topologieänderungen in Sensornetzwerken* müssen in Zukunft ebenfalls bei der Integration von Datenbanktechniken zur Abfrage von Sensornetzen berücksichtigt werden (Ablaufplan bzw. Routing von DB-Anfragen). Derzeitige Ansätze [De03] gehen meist von nicht ortsveränderlichen Sensoren aus.

5 Datenmanagement für künftige mobile AR Systeme

Zukünftige mobile AR Systeme werden für ihre Datenverwaltung vollständig auf Datenbanksysteme aufbauen. Derzeitige AR Systeme basieren dagegen größtenteils noch auf der Datenverwaltung in Dateisystemen oder liefern komplette, abgeschlossene Szenengraphen als Ergebnis auf Datenbankanfragen [RS03]. Bei zukünftigen direkten Koppelungen von AR Systemen und DBS müssen die einzelnen Anfragen in die bereits vorhandenen Datenbestände des AR Systems integriert werden können, ohne eine neue AR Szene aufzubauen. Das Datenmanagement für zukünftige mobile AR Systeme muss dabei eine Vielzahl von Datenquellen integrieren können. Wir werden zuerst auf das bisherige Datenmanagement, wie es in bestehenden AR Systemen verstanden wird [RS03][Br03], eingehen um anschließend auf die Zusammenführung dieser Strategien zu einer Datenmanagementstrategie für zukünftige AR Systeme zu kommen.

5.1 Bisheriges Datenmanagement

Bisher wurden AR Systeme entweder mit entfernten Datenbanksystemen gekoppelt (Online Datenmanagement) oder direkt ohne eine Netzverbindung im offline Modus mit lokalen Datenbanksystemen (Offline Datenmanagement) oder Dateisystemen betrieben. Neben diesen beiden Möglichkeiten des Datenmanagements besteht weiterhin die Möglichkeit zwischen kooperierenden AR Clients im Gelände mobile Ad-hoc Netzwerke aufzubauen und hierauf den Austausch der Datenbestände untereinander zu ermöglichen (MANET Datenmanagement).

Online Datenmanagement: Beim online Datenmanagement besteht zwischen dem AR System als mobiler Client und einem Datenbanksystem im Internet eine Client/Server-

Verbindung. Diese Verbindung muss für die gesamte transaktionelle Interaktion aufrechterhalten werden. Diese Kategorie stellt den derzeitigen Status-Quo bei der Verbindung von AR Systemen und DBS dar. Der Zugriff auf das Datenbanksystem erfolgt dabei in der Regel nicht direkt, sondern über spezielle Middleware-Dienste, welche die existierende Datenbasis für AR Systeme vorprozessieren bzw. höherwertige Funktionalität aufbauend auf dem DBS zur Verfügung stellen [BBT04]. Auf diese Weise ist eine Speicherung der Datenbasis in einem allgemeinerem Datenmodell möglich, während die Datenauslieferung in einem spezifischen AR Datenmodell erfolgen kann. Diese Art des Datenmanagements wird hauptsächlich für die Abfrage von Daten benutzt. Das Update von Datenbeständen in AR Systemen mit online Datenmanagement führt dagegen sehr schnell zum „Long Transaction Problem".

Grundlegender Vorteil von online Datenmanagement ist die Verfügbarkeit hochaktueller Datenbestände und der Zugriff auf beliebige Datenquellen im Internet. Nachteile ergeben sich vor allem aus der benötigten Netzverbindung. Probleme hierbei sind beispielsweise ungenügende Bandbreite/Netzabdeckung oder die immer mehr in Vordergrund tretenden Kosten und hoher Energieverbrauch.

Offline Datenmanagement: Begrenzte Energiereserven in mobilen Systemen, verbunden mit hohen Kosten und hohem Energieverbrauch drahtloser Verbindungen, bedingen die Unterstützung von offline Datenmanagement auch für AR Systeme. Dieses derzeit auf Dateisystemen basierende offline Datenmanagement wird in Zukunft durch lokale mobile Datenbanken übernommen. Hierfür werden mobile Datenbanksysteme benötigt, die einerseits komplexe 3D Geoobjekte, wie sie in der AR zwangsläufig benötigt werden, verarbeiten können und andererseits auch mit den jeweiligen Server-Pendants integriert sind. Die derzeit existierenden mobilen Datenbanksysteme [MS03] sind jedoch weit von der Eignung für AR Systeme entfernt. Ihnen fehlt die Erweiterbarkeit um nutzerdefinierte Datentypen und Indexstrukturen. Weiterhin sind die derzeitigen Konfliktauflösungsstrategien nicht für ein hohes Konfliktpotential geeignet [Go03], wie es gerade im geowissenschaftlichen Bereich vorherrscht. Hier scheint eine nutzerorientierte Sichtweise der Konfliktauflösung (Kooperation) die bessere Alternative zu sein.

Offline Datenmanagement bietet als Vorteil die Unabhängigkeit von Netzverbindungen und einen direkten, effizienten Zugriff auf die vorhandenen lokal replizierten Daten. Nachteile ergeben sich daraus, dass die replizierten Daten nicht auf dem aktuellsten Stand sein müssen und nicht replizierte Datenbestände in diesem Modus nicht verfügbar sind. Es besteht natürlich die (kostenintensive) Möglichkeit über GPRS/UMTS nicht replizierte Datenbestände verfügbar zu machen. Dies würde jedoch einem kombinierten Offline/Online Datenmanagement entsprechen.

MANET Datenmanagement: Neben alleinstehenden AR Systemen sind in Zukunft auch die Interaktionen zwischen mobilen AR Systemen zu unterstützen. Hierfür bauen die AR Clients im Gelände ein mobiles Ad-hoc Netzwerk (MANET) auf und schließen sich innerhalb des MANETs in kooperierende Gruppen zusammen. Das MANET muss dabei zum einen die Koordination der AR Clients in ihrer Aufgabenstellung unterstützten und zum anderen eine Integration der lokalen mobilen Datenbanken ermöglichen. Das Spektrum der Integration reicht hier von der Beantwortung expliziter Anfragen an

spezifische Clients bis zur automatischen Propagierung von Änderungen an den Datenbeständen an die kooperierenden Clients.

5.2 Zusammenführung des Datenmanagements

Wir gehen davon aus, dass ein zukünftiges Datenmanagement die bisherigen einzelnen Datenmanagementstrategien zusammenbringen muss. Das zentrale Datenbanksystem für AR Systeme wird das lokale Datenbanksystem darstellen. Dieses lokale Datenbanksystem (LDBS) fungiert als Integrationsdatenbank für eine beliebige Menge von Datenquellen. Diese Datenquellen können dabei entweder offline oder online zur Verfügung stehen, Datenbanken in einem MANET oder Datenbankschnittstellen zu lokalen Sensornetzwerken darstellen. Das LDBS ist hierbei nicht als ein Multidatenbanksystem für heterogene Datenbestände anzusehen, vielmehr als Metadatenbank über - für das Projekt - existierende Datenquellen mit dem Wissen, wie diese zur Verfügung gestellt werden. Für ein bestimmtes Gebiet wird demzufolge ein LDBS aufgebaut und die existierenden online Datenquellen, MANETs und Sensorschnittstellen registriert. Weiterhin können Teile der online Datenquellen lokal im Datenbanksystem repliziert werden, um häufige online Zugriffe zu vermeiden. Werden Daten aus online Datenquellen benötigt, so werden diese beim Abruf auch im LDBS, für spätere weitere offline Zugriffe, gespeichert.

Zentraler Punkt zukünftiger mobiler AR Systeme wird auch die Datenerfassung bzw. das Datenupdate sein. Hierfür muss das LDBS auf lokal replizierten Daten Update-Operationen unterstützen und eine Wiedereinbringung in den Serverdatenbestand ermöglichen. Versionsmanagement bietet hier die Möglichkeit kooperative Konfliktauflösungswege anzubieten. Hierbei werden Änderungen in den lokalen Datenbanken nicht transaktionell eingebracht, sondern als Versionen im Server-DBS vorgehalten. Bevor eine Version in die Datenbasis eingebracht werden kann, müssen etwaige Konflikte mit konkurrierenden Versionen aufgelöst werden. Diese Art der Konfliktauflösung sehen wir in AR Systemen als praktikabel an, da hier im Gegensatz zu Standardanwendungen keine große Anzahl an konkurrierenden Zugriffen zu erwarten ist. Vielmehr liegen die Anwendungsgebiete der AR es nahe, dass vornehmlich kooperative Anwendungen entwickelt werden.

6 Fazit

Erweiterte Realität (augmented reality) wird in der Zukunft als Benutzerschnittstelle für Datenbanksysteme eine größere Rolle spielen. Der zukünftige Bedarf an datenbankgestützten AR Anwendungen wurde anhand zweier Szenarien „Hochwasser“ und „Hangrutschung“ aus dem Umweltbereich gezeigt. Hieraus wurde in einer ersten Annäherung die benötigte Funktionalität für Datenbanksysteme und spezifische Middleware-Dienste abgeleitet. Ferner wurde die bisherige Verwendung von Datenbanken in AR Systemen in die drei Kategorien „Online Datenmanagement“, „Offline Datenmanagement“ und „MANET Datenmanagement“ eingeteilt und deren

Möglichkeiten aufgeführt. Für zukünftige mobile AR Systeme wird vorgeschlagen, diese Datenmanagementstrategien in eine AR Datenmanagementstrategie auf Basis eines „integrierenden“ lokalen, mobilen AR Datenbanksystems zu vereinen.

Danksagung

Dies ist Publikation Nr. GEOTECH-113 des GEOTECHNOLOGIEN Programms, gefördert von BMBF und DFG, Förderkennzeichen 03F0373B et al.

Literaturverzeichnis

[Ab03] Abiteboul, S.; Agrawal, R.; Bernstein, P. et al.: The Lowell Database Research Self Assessment, Microsoft Research Report, MSR-TR-2003-69, 2003.

[BeVo01] Bernhard, F.; Votteler, M.: Topographische Aufnahme des Rutschhanges Winkelgrat. Diplomarbeit, FH Stuttgart, FB Vermessung u. Geoinformatik, 2001, 41S.+Anhang.

[BBT04] Breunig, M.; Bär, W.; Thomsen, A.: Usage of Spatial Data Stores for Geo-Services. Proc. of the 7th AGILE Conf., Heraklion, Greece, 2004, 687-696.

[Br03] Breunig, M.; Malaka, R.; Reinhardt, W.; Wiesel, J.: Vision mobiler Geodienste. In: [Tü03], 6 S.

[De03] Demers, A.; Gehrke, J.; Rajaraman, R.; Trigoni, N.; Yao, Y.: The Cougar Project: A Work-In-Progress Report, SIGMOD Record, Vol.32, No. 4, 2003, 53-59.

[FOR03] Fornefeld, M.; Oefinger, P.; Rausch, U.: The market for geospatial information: potentials for employment, innovation and value addes. Commissioned by BMWA. Basis einer Parlamentarischen Anfrage zum Thema Geoinformationspolitik in Deutschland, MICUS Management Consulting GmbH Düsseldorf, 2003, 171S.

[Go03] Gollmick, Ch.: Client-Oriented Replication in Mobile Database Environments. Jenaer Schriften zur Mathematik und Informatik, Math/Inf/08/03, Universität Jena, 2003.

[HHM03] Hellerstein, J.M.; Hong, W.; Madden, S.R.: The Sensor Spectrum: Technology, Trends, and Requirements, SIGMOD Record, Vol.32, No. 4, 22-27, 2003.

[La02] Landesamt f. Geologie, Rohstoffe u. Bergbau Baden Württemberg: Georisiken: Aktive Massenbewegungen am Albtrauf. LGRB-Nachrichten Nr.8/2002.

[Mü01] Müller S.: Virtual Reality - Augmented Reality. INI-Graphics-Net. Brochure, Fraunhofer - IGD, Darmstadt, 2001.

[MS03] Mutschler, B.; Specht, G.: Implementationskonzepte und Anwendungsentwicklung kommerzieller mobiler Datenbanksysteme. In: [Tü03], S. 67-76.

[Ma04] Madden, S.; Hong, W.; Hellerstein, J.M.; Franklin, M.: TinyDB web page – http://telegraph.cs.berkeley.edu/tinydb

[RS03] Reitmayr, G.; Schmalstieg, D.: Data management strategies for mobile augmented reality. Proceedings STARS 2003, Japan, 2003.

[Su02] Suthau, T.; Vetter, M.; Hassenpflug, P.; Meinzer, H.-P.; Hellwich, O.: Konzeption und Einsatz von Augmented Reality in der Leberchirugie, DGPF 2002, 12 S.

[Tü03] Türker, C. (Hrsg.): Proceedings Workshop “Mobilität und Informationssysteme” des GI-Arbeitskreises “Mobile Datenbanken und Informationssysteme”, Techn. Report No. 422, Departement Informatik, ETH Zürich, 2003.

[WCS04] Wursthorn, S.; Coelho, A.H.; Staub, G.: Applications for Mixed Reality. In: XXth ISPRS Congress, Istanbul, Turkey, 12-23 July 2004.

Service Offer and Request Descriptions in Mobile Environments*

– A Position Paper –

Johannes Grünbauer[1] and Michael Klein[2]

[1]Institut für Informatik, Technische Universität München,
85748 Garching, Germany, gruenbau@in.tum.de
[2]Institute for Program Structures and Data Organization, Universität Karlsruhe,
76128 Karlsruhe, Germany, kleinm@ipd.uni-karlsruhe.de

Abstract. In order to obtain robust and context-aware applications in mobile environments, service oriented computing can be a very promising programming paradigm. Here, the participants cooperate in a loosely coupled manner by invoking context-depend services on demand. However, the approach demands for an expressive description of the offered and requested services. This paper aims at giving a basis for discussion on the requirements and approaches when describing services for mobile environments.

1 Introduction

Service oriented computing is a new paradigm that is especially interesting in mobile environments. As a characteristics, functionality is hidden behind an interface and described as a black box with the help of a service description language. This enables participants of the network to enlarge the limited capabilities of their devices by using services provided by others. As service requestors and providers are not fixedly tied together but are dynamically matched and bound, this architecture is especially advantageous in mobile environments and their constantly changing situation.

Typically a service usage follows the so called *service triangle* (see Figure 1): The service provider (which can be mobile) wants to offer a certain functionality. He describes it as service using the service description language (Step 1) and publishes this description at the service repository (Step 2). This repository can be central or distributed. If a requestor (which can be mobile, too) want to use a certain functionality, he described his requirements as service request (Step 3) and sends it to the repository (Step 4). Here, a matchmaking of the request and the offers takes place (Step 5). Matching offers are returned to the requestor, who selects one of them (Step 6), configures it according to his requirements

* The ideas for this paper have been developed at the Dagstuhl Seminar 04441 on *Mobile Information Management* in October 2004 together with Georgia Koloniari, George Samaras, and Can Türker.

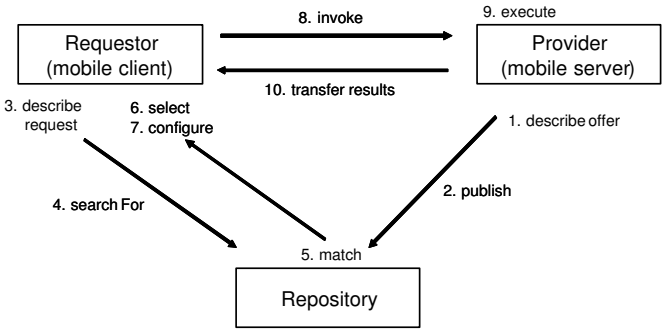


Fig. 1. The service triangle.

(Step 7), and invokes the corresponding service provider directly (Step 8). The provider executes the service with the given configuration (Step 9) and – if there are any results – returns them to the requestor (Step 10).

In the following, we want to give a basis for a discussion for two major topics from this process with regard to their characteristics with regard to mobile environments: Service offer descriptions (Section 2) and service request descriptions (Section 3). We conclude the paper by taking a look at some challenges in this area.

2 Service Offer Descriptions in Mobile Environments

First, we give a classification of services in mobile environments. This can be done by splitting the space along two dimensions (see Figure 2):

– *Mobility.* Whether the service itself is mobile or not.

	mobile	**non-mobile**
location dependent	Car Taxi (location affects functionality, context - related)	Restaurant
not location dependent	Tools on mobile devices (functionality is independent from the location)	Flight booking

Fig. 2. Classification of services in mobile environments.

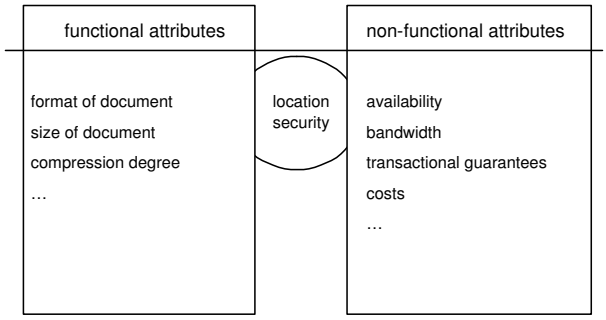


Fig. 3. Important functional and non-functional attributes for services in mobile environments.

– *Location Dependency.* Whether the content of the service is dependent from the location where it is used.

The simplest services are non-mobile, non location dependent services like a flight booking service in the internet. It runs on a central server and provides a functionality that is not bound to a specific location. In contrast, a restaurant booking service could enable a client to reserve a place in a restaurant in a certain area. Therefore, this is a location dependent service. Examples for service where the providing device itself is mobile could be a taxi reservation service (location dependent) or tools like a dictionary on a mobile device (not location dependent).

When describing these services, it is necessary to give information about certain attributes that are of special importance in mobile environments. They can be divided in functional and non-functional attributes (see Figure 3). Important functional attributes could be the format, size, or compression degree of a file as mobile devices typically have limited capabilities in dealing with all file types. With regard to non-functional attributes, availability, bandwidth requirements, transactional guarantees, and costs could be relevant when checking whether a given service is appropriate in a mobile situation.

However, filling the attributes of a service description leads to problems in a mobile environment: Changes in the environment can also lead to changes in the description, so frequent updates of the description would be necessary. To avoid this, the description could be split up into two parts: a static part containing the regular service description and a dynamic part, which captures the current context of the service provider like his current location, the times of availability and so on. This division could be done logically only or lead to two different documents which could be stored in two different repositories. With our classification from above, mobile services would have a large dynamic part whereas non-mobile service would only have a small one.

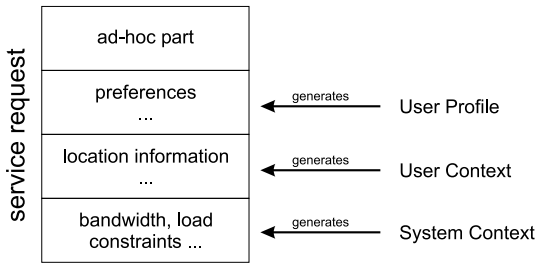


Fig. 4. Generating requests.

3 Service Request Descriptions in Mobile Environments

Requesting services is quite similar to performing a database query. The user or a program needs to have a certain query language to get information from a service. However, we neither create a new query language nor relate to a existing one in this section, but we present some ideas for request descriptions in mobile environments.

In mobile environments there exists additional information which has to be given to the service—the context information (cf. Sect.2). The context adds additional constraints to the request. We call these constraints *implicit information*.

As shown in Fig.4, a mobile request is assembled by different parts: The *ad-hoc part*, the *User Profile*, the *User Context* and the *System Context*.

The implicit information is generated by different aspects:

- The *User Profile* generates information like preferences of the user. This profile can be set up manually or be generated by the user's behaviour over a period using an AI-system. The preferences could contain information like "user prefers Chinese food" or "user likes musical shows".
- The *User Context* generates information like location information. This makes sense when ordering a taxi or in case of using a PDA with a travel guide system which can e.g. offer information about sights nearby the user's current position.
- Furthermore, the *System Context* generates information like bandwidth or load constraints. To continue the travel guide example, we assume that the user wants to watch an information movie on his PDA about a sight he is currently standing in front of. If the bandwidth is low, the system should provide the user a movie with a low resolution or sound with a low quality.

The *ad-hoc*-Part is the request generated by the user. This is the explicit information the user gives to the service ("I need a taxi in 20 minutes" or "I want to reserve a table in a restaurant."). It must be possible to overwrite the generated implicit information by the ad-hoc part.

Example 1 (Taxi Service). A person queries a service to get a taxi ("I need a taxi in 20 minutes."). Here we have to distinguish between implicit and explicit

information. The explicit information is, that a taxi is needed in 20 minutes. But this information is completely worthless if the service doesn't know *where* the taxi should be in 20 minutes. So, here the implicit information (user context) is the location of the user.

If the query contains explicit location information ("I need a taxi in 20 minutes *at the Plaza Hotel*"), the information generated by the user context will be overwritten. □

Example 2 (Reservation Service). Another example is a service, which can be used to reserve a table in a restaurant ("I want to reserve a table in a restaurant."). The implicit information for the service request could be:

- the restaurant should be nearby (user context)
- the user prefers Chinese over Italian food (user profile)
- advertising movie about the restaurant in a certain quality available (system context)

Again, the implicit information can be overridden with queries like: "I want to reserve a table in a Greek restaurant in Munich". □

4 Summary and Further Challenges

In this paper, we started the discussion on requirements and approaches for describing and matching services especially in mobile environments. We presented a classification of services in these environments and showed which attributes in the description are necessary here. Moreover, a first approach to complete ad-hoc queries with profile and context information was sketched.

However, as this is a rather new research topic, a number of open question remain. On the one hand, it has to be analyzed how detailed service descriptions should be, i.e. which constructs are needed to build and process them. On the other hand, it should be researched how such descriptions can be compared. Moreover, questions about appropriate similarity measures arise.