

Ein Framework zur Nutzung situationsabhängiger mobiler Dienste

Thomas Diekmann, Nick Gehrke

Institut für Wirtschaftsinformatik, Abteilung II
Georg-August-Universität
Platz der Göttinger Sieben 5
37073 Göttingen
{tdiekma|ngehrke}@uni-goettingen.de

Abstract: In diesem Beitrag soll ein Framework vorgestellt werden, dass die Entwicklung von Anwendungen zur Nutzung situationsabhängiger mobiler Dienste unterstützt, den Kontext des Anwenders verwaltet und die Kommunikation zwischen Anwendung und mobilen Diensten abwickelt. Da Anwendungen, die sich implizit an die Nutzungssituation des Anwenders anpassen, wesentlich einfacher zu benutzen sind, stellt das Framework einen Schritt in Richtung der Vision des „Ubiquitous Computing“ dar.

1. Einleitung

Auch wenn einige Autoren konstatieren, die Weiser'sche Vision [We91] sei - zumindest technologisch gesehen - greifbar nah, ist im Bereich des Ubiquitous Computing (UC) und somit auch im Bereich der kontextadaptiven bzw. situationsabhängigen Dienste nach einem Hype eine allgemeine Ernüchterung eingetreten. Die Mobilfunkbranche als ehemals treibende Kraft des UC ist aufgrund der zum Teil desaströs verlaufenden UMTS-Einführung stark verunsichert und deswegen wenig daran interessiert in Visionen zu investieren. Vielmehr müssen Anwendungen gefunden werden, die der Vision des UC näher kommen, aber technisch ohne großen Aufwand schon heute realisierbar sind. In diesem Beitrag soll das Konzept eines Frameworks für Anwendungen, die situationsabhängige (insbesondere ortsbezogene) Dienste nutzen, vorgestellt werden. Im Vergleich zu anderen Arbeiten im Bereich kontextadaptiver Dienstonutzung [Sa02], die aufgrund der hohen Komplexität der Modellierung des gesamten Kontexts ein hohes Abstraktionsniveau wählen ist der Betrachtungsfokus der vorliegenden Arbeit wesentlich konkreter.

2. Grundlagen

2.1 Situationsabhängige Dienste

Situationsabhängige Dienste sind Dienste, die die Situation des Anwenders berücksichtigen und situationsadäquate Dienstleistungen (z.B. die Lieferung von Inhalten) erbringen. Durch die Berücksichtigung der Situation haben diese gegenüber

nicht-individuellen Diensten einen Mehrwert für den Anwender [AW02]. Die Situation eines Anwenders, auch als Anwenderkontext bezeichnet, wird durch die Zeit, den Ort und das Profil des Anwenders (persönliche Präferenzen, demographische Daten etc.) determiniert. Im Bereich des stationären Internets sind der Ort und das Profil des Anwenders in der Regel statisch. Bei mobilen Anwendungen ändert sich aber insbesondere der Ort des Anwenders in unregelmäßigen Abständen.

2.2 Technologien zur Ortsbestimmung mobiler Geräte

Es gibt zahlreiche Technologien zur Bestimmung des Ortes eines mobilen Geräts und somit des Benutzers. Die Technologien können in drei Kategorien zusammengefasst werden, die in Tabelle 2.2.1 beschrieben werden [AW02].

IP-basierte Ortsbestimmung	Fest zugeordnete IPs lassen sich oft einem bestimmten Adressraum zuordnen. Die Adressräume lassen sich wiederum z. T. Regionen zuordnen. Dynamisch vergebenen IPs können Providern zugeordnet werden, die z. T. nur regional tätig sind.
Terminal-basierte Ortsbestimmung	Die terminal-basierten Ortsbestimmung erfolgt durch das Gerät selbst. Es nutzt dazu z. B. die Daten des Global Positioning Systems (GPS), das eine satellitengestützte Ortsbestimmung ermöglicht, aus.
Netzbasierte Positionsbestimmung	Anhand der durch das Gerät genutzten Basisstation(en) können Provider die Position des Geräts ermitteln.

Tabelle 2.2.1: Technologien zur Ortsbestimmung mobiler Geräte

Da es sich bei Informationen über den Standort einer bestimmten Person um personenbezogene Daten handelt, unterliegen sie den zum Teil sehr restriktiven Bestimmungen der Datenschutzgesetzgebung [RZ02]. Um Verstöße gegen die Gesetzgebung zu verhindern, müssen Mobilfunkanbieter (aufwendig) sicherstellen, dass entsprechende Daten nur an Dritte übermittelt werden, die zur Nutzung berechtigt sind. Der Anwender hat nur einen sehr kleinen Einfluss darauf, an welche Parteien seine Informationen weitergeleitet werden. Werden die Daten allerdings durch ein eigenes Gerät (Handy, PDA, Notebook etc.) ermittelt, hat er volle Verfügungsgewalt über diese Daten und kann selbst entscheiden, wer welche Daten bekommt. Das in diesem Beitrag abgehandelte Framework verwendet aus diesem Grund GPS-Daten, die durch einen handelsübliche GPS-Antenne empfangen werden.

3. Konzept

3.1 Fachliche Konzeption

Das Framework soll die Entwicklung von Anwendungen zur Nutzung situationsabhängiger mobiler Dienste unterstützen, Kontextinformationen verwalten und die Kommunikation zwischen Anwendung und Diensten abwickeln. Das Framework ist insbesondere auf mobilen Endgeräte ausgerichtet, die zum einem über ausreichend Rechenkapazität und zum anderen über einen – idealerweise – permanenten Zugang zum Internet verfügen. Denkbar wären hierbei beispielsweise PDAs mit UMTS- bzw. WLAN- Zugang oder in Automobilen installierte Geräte. Das Framework soll in der Lage sein die durch die Zeit, den durch ein GPS-Gerät ermittelten Ort und dem Profil

des Anwenders determinierte Situation zu verwalten und situationsabhängigen Diensten zur Verfügung zu stellen, damit diese situationsadäquate Inhalte liefern können. Die gelieferten Inhalte sollen mittels eines geeigneten Graphical User Interface (GUI) durch die Anwendung dargestellt werden.

Für die Interaktion zwischen den beschriebenen Komponenten des Frameworks lassen sich grundsätzlich zwei verschiedene Interaktionsmodelle einsetzen [Fi02]:

1. **Pull-Modell:** Der Anwender greift über die Anwendung gezielt auf einen Dienst zu. Die Anwendung reicht die Anfrage mit Informationen zur Situation an und bereitet die von dem Dienst gelieferten Inhalte grafisch für den Anwender auf. In diesem Interaktionsmuster muss der Anwender entscheiden, ob eine Anforderung bzw. Neuanforderung von Inhalten notwendig ist. Dieses Interaktionsmuster entspricht dem Thin-Client-Konzept, in dem der Client keinerlei Logik verarbeitet.
2. **Push-Modell:** In dem Push-Modell übernimmt der Dienst die Initiative und sendet dem Anwender unaufgefordert Inhalte. Im Fall situationsabhängiger Dienste setzt dies voraus, dass der Dienst über die Situation des Benutzers informiert ist.

Das Framework setzt eine Hybridform aus beiden Interaktionsmodellen ein. Der Nutzer ergreift zwar anfangs die Initiative und entscheidet sich für eine bestimmte Dienstnutzung, über eine Neuanforderung entscheidet aber das Framework. Die Entscheidungen des Frameworks werden anhand eingetretener Ereignisse (Events) getroffen, wie z. B. die Veränderung der geografischen Position des Anwenders. Welche Events von Relevanz sind, muss vor der Nutzung des Dienstes durch den Dienst oder durch den Anwender festgelegt werden. Die von dem Framework unterstützten Events lassen sich in positionsbezogene Events (z.B. „der Anwender hat sich X Längeneinheiten von einer Position Y entfernt“) und zeitbezogene Events (z.B. „ein Zeitpunkt X ist erreicht“ oder „ein zyklischer Termin X ist erreicht“) untergliedern. In Abbildung 3.1.1 wird der Vorgang der Dienstnutzung in Form eines Interaktionsdiagramms dargestellt.

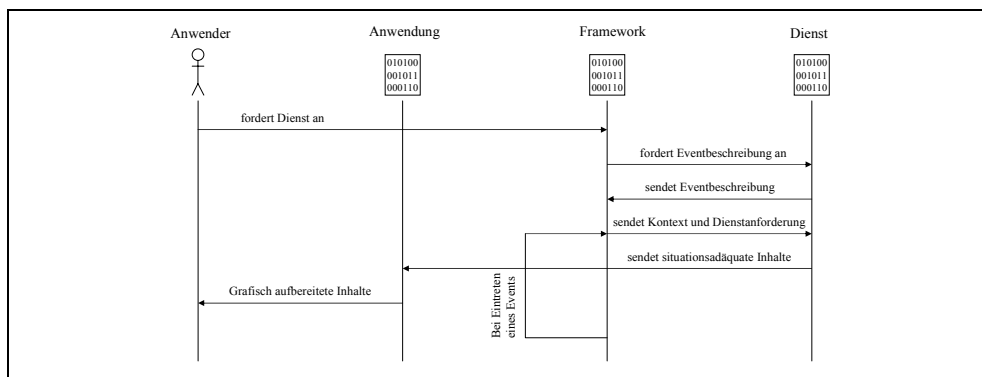


Abbildung 3.1.1: Interaktionsdiagramm einer Dienstnutzung

Der Anwender fordert durch die Anwendung einen Dienst an. Die Anwendung reicht die Anforderung an das Framework weiter, welches den angeforderten Dienst auffordert, eine Beschreibung der für Neuanforderungen relevanten Events zu senden. Das

Framework wertet die Beschreibung aus und initialisiert einen Eventhandler, der zyklisch überprüft, ob bestimmte Events eingetreten sind. Im Anschluss sendet das Framework den Kontext des Anwenders an den Dienst, der daraufhin situationsadäquate Inhalte liefert. Diese Inhalte werden an die Anwendung weitergeleitet, die diese grafisch für den Anwender aufbereitet.

3.2 Technische Umsetzung

Um die Machbarkeit und die praktische Einsatzfähigkeit des Konzepts zu evaluieren, wird das in Kapitel 3.1 erläuterte Konzept zurzeit prototypisch mit der Programmiersprache Java umgesetzt. Aufgrund der hohen Verfügbarkeit von Java Virtual Machines für verschiedene Endgeräte wird durch den Einsatz von Java zur Umsetzung eine plattformübergreifende Entwicklung ermöglicht. Zur Erfassung des Ortskontextes des Anwenders kommen GPS-Empfänger zum Einsatz, die den weit verbreiteten NMEA-Standard zur Kommunikation zwischen GPS-Gerät und anderen Geräten unterstützen. Entsprechende Geräte werden zu einem erschwinglichen Preis für eine breite Spanne von mobilen Endgeräten angeboten. Die Beschreibung der Events und des Kontextes erfolgt mit XML auf Basis speziell für das Framework spezifizierten Schemen.

Folgend wird ein Klassendiagramm skizziert, welches die Architektur des Frameworks verdeutlichen soll. Es werden nur die wichtigsten Klassen und Methoden vorgestellt, um nur das Wesentliche darzustellen.

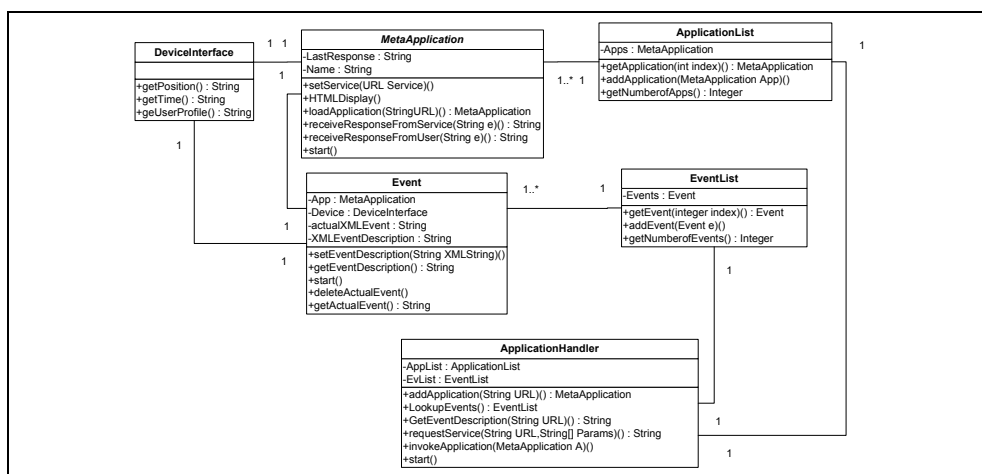


Abbildung 3.2.1: Klassendiagramm des Frameworks

Es folgt eine kurze Beschreibung der Klassen:

- **DeviceInterface** kapselt das Endgerät, welches Informationen der Umwelt feststellt (z.B. eine GPS-Antenne).
- **MetaApplication** ist eine abstrakte Klasse. Alle konkreten Anwendungen müssen von dieser Klasse abgeleitet werden. Das dynamische Beziehen der Klassenlogik durch einen Classloader ist durch die Methode `loadApplication` möglich. Lediglich die Methoden `receiveResponseFromService` und `receiveResponseFromUser` müssen

in den konkreten Klassen implementiert werden, da hier die individuelle Applikationslogik enthalten ist. Die Entwicklung einer Anwendung entspricht also somit der Implementierung einer abgeleiteten Klasse.

- **ApplicationList** enthält eine Liste von Applikationen, die im Framework koexistieren.
- Die Klasse **Event** kapselt ein Ereignis. Ist ein Ereignis relevant, wird es in *actualXMLEvent* abgelegt.
- **EventList** enthält eine Liste von Events, die im Framework koexistieren.
- Der **ApplicationHandler** ist ein Thread, der die „Kommunikationszentrale“ des Frameworks darstellt. Diese Klasse kommuniziert mit den Services im Internet (*requestService*) und leitet die Ergebnisse von Anfragen an die entsprechenden Applikationen weiter (*invokeApplication*). Auch horcht diese Klasse ständig nach neuen Events, um einen situationsadäquaten Aufruf eines Services zu initiieren.

4 Fazit und Ausblick

In diesem Beitrag wurde ein Framework vorgestellt, dass die Entwicklung von Anwendungen zur Nutzung situationsabhängiger mobiler Dienste unterstützt, den Kontext des Anwenders verwaltet und die Kommunikation zwischen Anwendung und mobilen Diensten abwickelt. Technisch ist die Umsetzung des Frameworks machbar. Lediglich der geforderte breitbandige Interzugangszugang ist noch nicht flächendeckend gegeben, was den Einsatz von situationsabhängigen Diensten auf wenige „kleine“ Anwendungen beschränkt. Es ist aber zu erwarten, dass die enormen Investitionen, die die Mobilfunkanbieter zurzeit in UMTS-Lizenzen und –infrastruktur tätigen, dieses Problem in mittlerer Zukunft relativieren. Um die Praktikabilität des Frameworks zu verifizieren, wird z. Z. angedacht diverse Applikationen z. B. im Bereich Touristik (interaktiver Fremdenführer) auf Basis des Frameworks in Kooperation mit Praxispartnern zu entwerfen und umzusetzen.

Literaturverzeichnis

- [AW02] Amberg, M.; Wehrmann, J.: A Framework for the Classification of Situation Dependent Services. Eighth Americas Conference on Information Systems Proceedings (AMCIS 2002), Dallas, University of Texas, USA 2002, 1838-1843
- [Fi02] Figge, S.: Die Open Mobile Architecture – Systemumgebung für mobile Dienste der nächsten Generation. In: Wirtschaftsinformatik 44 (2002) 4, S. 375-378
- [RZ02] Rayermann, M.; Zimmer, M.: Rechtliche Grundlagen des M-Commerce. In (Gora, W.; Röttger-Gerigk, S.): Handbuch Mobile-Commerce – Technische Grundlagen, Marktchancen und Einsatzmöglichkeiten. Springer-Verlag, Berlin et al., 2002; S. 91-113
- [Sa02] Samulowitz, M.: Kontextadaptive Dienstnutzung in Ubiquitous Computing Umgebungen, Dissertation, München 2002
- [We91] Weiser, M.: The Computer of the 21st Century. In: Scientific American, 265(3): S. 94-104, September 1991