

Kontext-Adaption – Stand der Technik

Bernhard Ehringer
Fachhochschule Oberösterreich – Fakultät für Informatik/Kommunikation/Medien
Masterstudiengang Mobile Computing
Softwarepark 11, 4232 Hagenberg, Österreich
bernhard.ehringer@fh-hagenberg.at

KURZFASSUNG

Systeme, welche den Kontext – also die aktuelle Situation des Benutzers – in Darstellungsformen, Verhaltensmuster oder dargestellten Informationsgehalt miteinbeziehen werden immer beliebter. Dies hat einerseits den Grund, dass es bereits ein breites Spektrum an Anwendungen gibt, welches etwa die Positionsinformation nutzt, um diverse Dienste kontextsensitiv anzubieten. Andererseits wird es Softwareentwicklern durch das Bereitstellen verschiedener Frameworks immer leichter gemacht, innovative situationsabhängige Anwendungen zu entwickeln.

Ein wichtiger Aspekt dieser Systeme stellt die Adaption der Anwendung dar – also die Art und Weise wie sich die Anwendung an die persönliche Situation des Benutzers anpasst. Dabei können etwa statisch definierte Bedingungen, dynamische Regel-Mechanismen oder Ontologien verwendet werden. Dieser Artikel präsentiert eine Vielzahl an existierenden kontextsensitiven Systemen und beschreibt die Konzepte, welche verwendet werden um die Anwendung – bezogen auf den teilweise oft wechselnden Kontext – zu adaptieren. Außerdem wird untersucht, welche Teile der Anwendung angepasst werden (z. B. Darstellung, Informationsgehalt, Verhalten etc.) und eine Auswertung hinsichtlich der verwendeten Konzepte vorgenommen.

1. EINLEITUNG

In viele Bereichen des täglichen Lebens können kontext-sensitive Systeme dem Menschen helfen den Tagesablauf zu planen, wichtige Entscheidungen richtig zu treffen und keine Termine zu versäumen. Solche Systeme sind Anwendungen, die sich an die gerade herrschende Situation (Ort, beteiligte Personen, vorhandene Ressourcen etc.) und an die kontinuierliche Veränderung des Kontexts anpassen [25]. Dabei gibt es die unterschiedlichsten Ausprägungen – jedoch haben alle Anwendungen dieser Art etwas gemeinsam: Das Benutzen von Informationen der aktuellen Situation – sei es die Position, die Temperatur, vorhandene Umgebungsgeräusche oder die aktuelle Tageszeit.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

Bevor solche Systeme die Informationen der Umgebung für sich nutzen können, ist es erforderlich, ein geeignetes Datenmodell dafür zu finden und die Kontextinformationen zu verwalten. Letztere Aufgabe übernimmt das Kontextmanagement – dazu gehört auch das Erfassen der Daten von Sensoren sowie das Verteilen der Informationen an interessierende Anwendungsmodule. Dadurch haben diese Komponenten die Möglichkeit, ihr Verhalten oder Aussehen der aktuell herrschenden Kontextsituation anzupassen. Für eine solche Adaption werden meist Regeln verwendet, welche beschreiben, wie sich die Anwendung an spezielle Situationen anpassen soll.

Dieser Artikel beschreibt in Kapitel 2 wie Anwendungen mithilfe von Kontextinformationen adaptiert werden können. Vor dem Resümee werden in Kapitel 3 die Adaptionskonzepte der vorgestellten Projekte gegenübergestellt sowie die Rechercheergebnisse tabellarisch verglichen.

2. KONTEXT-ADAPTION: STAND DER TECHNIK

Die aktuelle Kontextsituation kann in einer Anwendung – etwa zur Änderung der Darstellungsform – berücksichtigt werden. Die Adaption von Informationen, Visualisierung oder Architektur kann mit Hilfe verschiedener Mechanismen erfolgen. Dabei ist der Einsatz von einfachen *if*- bzw. *case*-Statements, welche direkt in den Source-Code eingebettet sind, ebenso möglich wie die Verwendung umfangreicher und flexibler Regelmechanismen. Im folgenden Abschnitt werden bereits bestehende Konzepte und Projekte vorgestellt. Es wird außerdem untersucht, wie Kontext-Adaption im Rahmen dieser Projekte umgesetzt wird und welche Teile der Anwendung davon betroffen sind (z. B. Darstellung, Informationsgehalt etc.).

2.1 ParcTab (1995)

Als eines der ersten kontextbezogenen Systeme gilt das von Want u. a. entwickelte *ParcTab* [32], das mit Hilfe von *Tabs* (Vorfahren der heutigen PDAs) und einem Infrarot-Netzwerk Nachrichten austauscht und Objekte lokalisiert. Mittels stationärer *Device Agents* können an die Mobilgeräte verschiedene Anwendungen angebunden werden – etwa ein kontextsensitiver *Datei-Browser*. Dabei werden – abhängig von der aktuellen Position – nur jene Dateien des Dateisystems visualisiert, welche einen Bezug zum aktuellen Raum besitzen. Diese Relationen müssen bei Erstellung der Datei angegeben werden, um die Anwendung korrekt arbeiten zu lassen.

2.2 C-MAP (1998)

Fels u. a. beschreiben in ihrem Artikel das Project *C-MAP* [13], einen kontextsensitiven Tour-Guide für Ausstellungen, Museen oder Messen. Die Anwendung nutzt zuvor angegebene Interessen des Benutzers (Kontextinformationen), um eine Tour durch eine Messe zu planen. Der Benutzer soll somit den Weg zu den für ihn interessanten Ausstellungspunkten finden.

Mit Hilfe einer physischen Karte wird der Benutzer auf dem Messegelände positioniert – eine semantische Kartendarstellung zeigt dem Anwender jene Aussteller, die semantisch nahe bei seinen eigenen Interessen liegen.

2.3 Conference Assistant (1999)

Um Teilnehmer an Konferenzen mit Hilfe von Kontextinformationen zu unterstützen, wurde von Dey u. a. der *Conference Assistant* [10] entwickelt. Diese Anwendung baut auf das *Context Toolkit* [9] auf und versucht etwa – nach der Angabe von persönlichen Präferenzen mittels Benutzerprofil – für die jeweilige Person interessante Vorträge zu finden. Diese werden dem Benutzer mitsamt Detailinformationen präsentiert – es erfolgt also eine Adaption hinsichtlich Informationsgehalt.

Das Finden von geeigneten Vorträgen übernimmt ein so genannter Kontext-Interpreter, welcher die Kontextinformationen (persönlichen Präferenzen sowie aktive Präsentationen) heranzieht und passende Themen herausfiltert.

2.4 GUIDE (1999)

Das für die Stadt Lancaster entwickelte System *GUIDE* [5] stellt einen mobilen, kontextsensitiven Stadt-Führer dar. Die web-basierte Anwendung verwendet als einzige Kontextinformation die aktuelle Position des Benutzers, um Informationen über Sehenswürdigkeiten in der näheren Umgebung anzuzeigen. Dabei wird die aktuelle Position in einen Anfrage-URL kodiert, welche an einen Server geschickt wird. Die eigentliche Adaption der Informationen passiert innerhalb des Server-Dienstes, welcher die zur Position passenden Objekte identifiziert und an das Gerät retourniert.

2.5 News Dude (1999)

Billsus und Pazzani entwickelten mit *News Dude* [14] einen persönlichen Nachrichten-Agenten, welcher seinem Benutzer interessante Neuigkeiten bzw. Nachrichten präsentiert und vorliest. Anhand der Rückmeldung des Anwenders werden dessen Präferenzen und Interessen zu bestimmten Themenbereichen verwendet, um zukünftige Nachrichten nach eben diesen Interessen zu filtern. Die Rückmeldung erfolgt dabei verbal – etwa kann der User den News Dude beim Vorlesen eines Artikels unterbrechen und somit sein Desinteresse signalisieren. Des Weiteren kann der Benutzer nach Abhören einer Nachricht seine Einstellung zu diesem Thema signalisieren – etwa wird ein vom User abgegebenes „interesting“ von der Spracherkennung des News Dude erkannt und weiterverwendet. Neue Artikel werden mit Hilfe dieses Wissens und der Artikel-Kategorie in einem eigenen Modul bewertet und nach Relevanz sortiert. Somit werden – nach einer Lernphase der Applikation – nur die für den Benutzer interessanten Themengebiete angesprochen.

2.6 CybreMinder (2000)

Dey und Abowd beschreiben in ihrem Artikel [8] ein kontextsensitives Tool, welches den Empfang bzw. die Erinne-

rung an Aufgaben oder Notizen situationsgerecht steuert. Dabei werden vom Benutzer bei Erstellung der Notiz Situationen mit Hilfe von Kontextattributen (z. B. „location“) definiert, in denen die zugehörige Erinnerung erscheinen soll. Diese Regeln können weitere kontextbezogene Daten berücksichtigen – etwa die Zeit, die aktuelle Temperatur und den Namen des Benutzers.

Das Projekt verwendet die Infrastruktur des *Context Toolkits* [9] und ermöglicht es, Erinnerungen an Aufgaben per SMS, E-Mail, Display-Anzeige (PDA, CRT etc.) oder in Papierform (Drucker-Ausdruck) zu versenden.

2.7 ARREAL (2001)

Wahlster u. a. entwickelten im Rahmen des Projektes *RE-AL* [31] das ressourcenadaptierende Navigationssystem *ARREAL* – gedacht für Fußgänger im outdoor-Bereich. Die Anwendung stellt eine Karte dar, welche die aktuelle Umgebung des Benutzers visualisiert. Das Kartenmaterial wird dabei nicht nur mit Hilfe der Position, sondern auch mittels der gegenwärtigen Benutzergeschwindigkeit und der gerade vorhandenen Ortsgenauigkeit adaptiert. So zeigt die Anwendung etwa bei niedriger Geschwindigkeit eine genauere, detailreichere Darstellung – bei hoher Geschwindigkeit wird ein größerer Kartenausschnitt mit höherem Zoomlevel gewählt.

Die Adaption der Karte ist hierbei durch eine einfache Formel implementiert.

2.8 CRUMPET (2001)

Das von der EU initiierte Projekt *CRUMPET* [23] hat zum Ziel, benutzerfreundliche, mobile Dienste für den Tourismus zu entwickeln. Dabei werden vor allem der Position sowie der Benutzerkontext verwendet, um Informationen am Mobilgerät situationsgerecht anzupassen. Der Benutzerkontext wird dabei automatisch und dynamisch generiert – es erfolgt keine Angabe der Interessen durch den Anwender. Mit Hilfe der so gewonnenen Präferenzen werden etwa POIs gefiltert, und nur die für den Benutzer interessanten auf einer Karte visualisiert.

2.9 HyperAudio & HIPS (2001)

Petrelli u. a. stellen in ihrem Artikel [21] die Erfahrungen aus der Implementierung von zwei positionsabhängigen adaptiven Museum-Guides (*HyperAudio*, *HIPS*) gegenüber. Beide Systeme ermöglichen es, die aktuelle Kontextsituation in die Applikationslogik einzubeziehen und somit auf die Einflüsse von außen zu reagieren.

Beim Besuch eines Museums wird – ausgehend von der Position des Anwenders – eine Präsentation der Objekte im näheren Umfeld erstellt und visualisiert. Diese besteht aus einer Audio-Mitteilung, einem Bild sowie weiterführenden Links. Die Präsentation wird dabei passend zum aktuellen Kontext (Position des Benutzers) adaptiert, wobei Regel-Mechanismen eingesetzt werden. Nicht nur das Modul zur Erstellung von Präsentationen sondern auch ein *Input Analyser* verwendet Regeln, um das Verhalten der Anwendung zu steuern – etwa wird das Schließen der Präsentation bei Verlassen der jeweiligen Sehenswürdigkeit veranlasst.

2.10 PinPoint (2002)

Aufbauend auf der *PinPoint*-Infrastruktur wird von Roth ein kontextsensitiver Touristen-Führer [24] beschrieben, welcher neben dem Gerätekontext auch die Position des Benutzers verwendet, um Informationen situationsgerecht darzu-

stellen. Dabei werden die wichtigen Informationen (z. B. Koordinaten und Bildschirmauflösung) in einen URL kodiert – der serverseitige Dienst interpretiert diese Informationen und liefert das entsprechende Karten- und Informationsmaterial.

2.11 Chisel (2003)

Ein kontextsensitives, regelbasiertes Framework zur dynamischen Adaption von Applikationsverhalten [15] stellen Keeney und Cahill vor. Dieser Ansatz basiert auf der Verwendung von reflektiven Systemen – diese besitzen zusätzlich zur üblichen Programmlogik auch Metadaten (Zusatzinformationen) über sich selbst. Darüber hinaus werden Änderungen in der Anwendung auch in den Metadaten präsentiert – umgekehrt ist es möglich, durch das Adaptieren von Metadaten das Verhalten der Anwendung zu steuern. Diesen Ansatz nutzt Chisel, um die Anwendung – ohne direkt in diese eingreifen zu müssen – in ihrem Verhalten kontextgerecht zu adaptieren.

Die Adaption selbst erfolgt mittels Regeln, welche außerhalb der Anwendung textbasiert definiert werden. Die Regeln enthalten üblicherweise einen Auslöser (z. B. ein bestimmter Event), ein Zielobjekt und eventuell Bedingungen, um die Regel auf bestimmte Situationen einzuschränken. Bei einem Wechsel der Kontextsituation werden alle verfügbaren Regelsätze abgearbeitet und bei Bedarf die zugehörige Aktion ausgeführt bzw. die Änderung der Metadaten veranlasst.

2.12 Gulliver's Genie (2003)

Einen weiteren mobilen kontextbezogenen Reiseführer für Touristen präsentieren O'Grady und O'Hare [20]. Dabei liegt der Fokus auf der Adaption von Präsentationen, welche bei Annäherung an eine Sehenswürdigkeit von einem zentralen Dienst geladen und visualisiert werden. Die Adaption des Präsentationsinhalts erfolgt auf Basis eines persönlichen Benutzerprofils sowie kulturellen Interessen – beides wird vom Anwender im Vorhinein festgelegt. Mit Hilfe von Tags, welche den Elementen der Präsentation zugewiesen sind, kann der Adaptionsprozess für den User interessante Präsentationsinhalte selektieren, und dadurch eine kontextsensitive Präsentation erstellen.

2.13 LOL@ (2003)

Umlauft u. a. beschreiben mit dem Projekt *LOL@* [29] einen mobilen Touristenführer, entwickelt für die Stadt Wien. Basierend auf dem Mobilfunkstandard UMTS¹ wurde eine Anwendung entwickelt, welche es dem Benutzer u. a. erlaubt Informationen über Sehenswürdigkeiten abzurufen, sich durch die Stadt navigieren zu lassen sowie ein elektronisches Tour-Tagebuch zu führen. Als einzige Kontextinformation wird dabei die Position des Benutzers verwendet, um auf einer Übersichtskarte nahe gelegene POIs visuell hervorzuheben. Diese Funktion wird durch einen zentralen Dienst realisiert, welcher die aktuellen GPS-Koordinaten mit verfügbaren POIs vergleicht und der Anwendung einen Befehl zur besonderen Darstellung der nahe gelegenen Objekte sendet.

2.14 mobiDENK (2003)

Das Projekt *mobiDENK* [16] realisiert ein mobiles, ortsbezogenes Informationssystem und wurde von Krösche u. a.

¹Universal Mobile Telecommunication System

entwickelt. Die Anwendung basiert auf der *Niccimon Plattform* [1] und dient dazu, Touristen auf nahe gelegene, besonders wertvolle Denkmäler und historische Sehenswürdigkeiten hinzuweisen sowie ortsabhängige multimediale Informationen dazu zu liefern. Mit Hilfe eines eingebauten GPS-Empfängers visualisiert ein Mobilgerät eine Karte mit der aktuellen Umgebung des Benutzers. Neben der eigenen Position werden von diesem Modul auch POIs – diese repräsentieren die Denkmäler und Sehenswürdigkeiten – dargestellt. Durch die Selektion verschiedener Typen von Sehenswürdigkeiten kann der Anwender die Auswahl der angezeigten Objekte einschränken und auf seine eigenen Interessen begrenzen. Dadurch werden neben der Positionsinformation auch persönliche Interessen des Users in die berücksichtigten Kontextinformationen aufgenommen.

2.15 Sigtseeing4U (2003)

Boll beschreibt in ihrem Artikel *Sightseeing4U* [3], ein auf Web-Services basierender multimedialer Fremdenführer. Dabei wird primär der Benutzer- und Gerätekontext verwendet, um multimediale Inhalte situationsgerecht auszuwählen und als Präsentation an den Client zu übertragen. Der Anwender wählt seine Präferenzen anhand einer Eingabemaske beim Start der Applikation. Wird eine Anfrage an den serverseitigen Dienst gestellt (z. B. Sightseeing-Tour durch Wien), so werden mittels der Interessen des Benutzers die interessantesten Objekte aus einer internen Datenbank selektiert. Eine daraus verfasste personalisierte Präsentation wird dem User am mobilen Client visualisiert.

2.16 CASS (2004)

Das von Fahy und Clarke [12] entwickelte *CASS*-System (Context-Awareness Sub-Structure) ist eine Middleware für mobile kontextbezogene Anwendungen. Dabei wird ein Verfahren namens *Forward Chaining* eingesetzt, um Adaptionen in der Anwendung abzubilden. Dieses Verfahren nutzt Regeln, um von bekannten Fakten (z. B. Kontextinformationen) neue, auf höherer Ebene dargestellte Fakten (z. B. eine bestimmte Situation) abzuleiten. Darauf aufbauend können Anwendungs-Adaptionen definiert werden – etwa das Darstellen von aktuellen Kontextinformationen für den Benutzer oder das automatische Ausführen von Aktionen bzw. Diensten.

2.17 COMPASS (2004)

Der von van Setten u. a. beschriebene kontextsensitive persönliche Assistent (*Context-aware Mobile Personal Assistant*) [30] bietet Informationen und Dienste speziell für touristische Benutzer. Dabei werden sowohl der aktuelle Benutzerkontext als auch diverse Präferenzen des Anwenders berücksichtigt, um bei einer Anfrage (z. B. „Finde eine Übernachtungsmöglichkeit“) für den Anwender interessante Objekte (POIs) zu identifizieren. Die Adaption der gelieferten Ergebnismenge übernimmt die *Recommendation Engine*, welche wiederum – mit Hilfe der Kontextdaten und der Benutzerpräferenzen – verschiedene Strategien zur Bewertung der POIs verwendet. Die gelieferten Ergebnisse werden dem Benutzer in Form einer Liste (mitsamt Beurteilung anderer User) visualisiert.

2.18 m-ToGuide (2005)

Schwinger u. a. beschreiben in ihrem Artikel eine Menge mobiler Touristen-Guides, darunter auch *m-ToGuide* [26].

Dabei werden Benutzer-Präferenzen in Form von Profilen verwendet, um die für den Anwender interessanten POIs zu finden. Nicht nur die POIs selbst, sondern auch deren Beschreibungen werden kontextsensitiv adaptiert. Um die Profile aktuell zu halten, wird das Verhalten des Benutzers untersucht und dessen Präferenzen automatisch aktualisiert. Neben dem Benutzerkontext spielt bei m-ToGuide auch der Zeit-Kontext eine Rolle – etwa werden Öffnungszeiten der POIs (z. B. Museen) berücksichtigt, um auf besuchbare Sehenswürdigkeiten hinzuweisen.

2.19 xPOI (2005)

Krösche und Boll verwenden in ihrem Projekt *xPOI* [17] Regel-Mechanismen (ECA-Regeln), um kontextsensitive Informationen in die Anwendung zu integrieren. Dabei werden georeferenzierte Interessenspunkte (Points-Of-Interest, POIs) in Form von Icons auf der Landkarte an die aktuelle Situation des Benutzers angepasst. Beispielsweise könnte die Darstellung von U-Bahn-POIs – also Teile des User Interfaces – aufgrund länderspezifischer Eigenheiten adaptiert werden.

2.20 Chameleon (2006)

Basierend auf dem URICA-Framework wurde von Mohamed u. a. Chameleon [18] entwickelt – ein ressourcenadaptierendes System, das Bilder von Webseiten durch Adaption der Qualität an bandbreitenlimitierte Verbindungen anpasst. Dabei wird auf Client-Seite ein adaptiver Browser verwendet, welcher Rückmeldung bzw. Veränderung der Qualität von jedem Bild durch den Benutzer zulässt. Serverseitig wird ein adaptierender Proxy eingesetzt, der die Webinhalte vom ursprünglichen Server lädt, adaptiert und an den Benutzer weiterleitet.

Die Bilder der angeforderten Seite werden zu Beginn optimal auf die langsame Verbindung angepasst – es erfolgt also eine relativ drastische Reduktion der Qualität. Mit Hilfe des adaptiven Browsers können die Anwender andere Qualitätsstufen jedes einzelnen Bildes vom Proxy anfordern. Diese Änderungen bezieht das System in neue Anfragen mit ein – Prädiktions-Algorithmen bestimmen die optimale Qualitätsstufe des zu liefernden Bildes, welche ein Kompromiss aus Ladezeit und Detailgehalt darstellt.

2.21 ContextL/ContextJ (2006)

Costanza u. a. stellen in ihren Artikeln [6, 7] einen auf Layern basierenden Ansatz (*ContextL*) zur Steuerung von Anwendungs-Adaption vor. Abbildung 1 illustriert diese Idee – von einem bestimmten Kontext (*foo*, *bar*) aus werden Methodenaufrufe getätigt. Der jeweils zuständige bzw. aufgrund der Kontextsituation aktive Layer übernimmt die Anfrage und führt die zugehörigen Aktionen aus.

ContextL ist im Kern eine Erweiterung der Programmiersprache CLOS (Common Lisp Object System), welche wiederum auf der Verwendung von generischen Funktionen basiert. Das für Java verfügbare Pendant *ContextJ* ist ebenso wie *ContextL* dazu gedacht, kontext-orientierte Programmierung zu unterstützen und zu vereinfachen. Dies geschieht mit Hilfe von Layern bzw. Schichten, wobei ein Root-Layer existiert und beliebig viele weitere Layer definiert werden können. Jeder dieser Layer kann nun dynamisch von der Anwendung aktiviert bzw. deaktiviert werden. Um dem Layer Leben zu verleihen, können Klassen mit einem spezifischen Layer assoziiert werden. Dies bedeutet, es wird innerhalb des

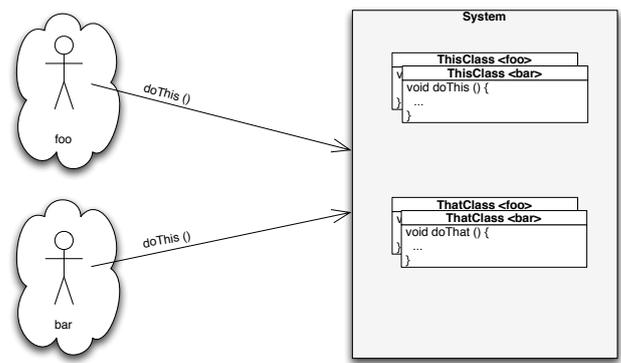


Abbildung 1: Layer-basierter Ansatz [6]

Layers eine Klasse, welche außerhalb bereits existiert, um Funktionen erweitert. Diese Funktionen werden nur dann ausgeführt, wenn der spezifische Layer aktiv ist. Somit ist es etwa möglich, Zusatzinformationen, welche von einem dafür vorgesehenen Layer stammen, nur in bestimmten Situationen anzuzeigen.

2.22 MOPET (2007)

Der *MOBile PErsonal Trainier* [4] ist eine von Buttussi und Chittaro entwickelte Anwendung, welche im Fitness-Bereich zum Einsatz kommt. Dabei werden – mit Hilfe von Sensoren – die sportlichen Aktivitäten einer Person analysiert und ausgewertet. Basierend auf dieser aktuellen Situation und persönlichen Daten (z. B. Alter) kann die Anwendung motivierende Anweisungen oder Alarme zu Sicherheit bzw. Gesundheit geben. Außerdem werden dem Anwender interaktive 3D-Animationen visualisiert, um die korrekte Ausführung der jeweiligen Übungseinheit zu zeigen.

Die kontextsensitive Adaption erfolgt hierbei durch das Vorschlagen von neuen Übungen bzw. durch alarmierende Hinweise, die Übung (z. B. laufen) zu verlangsamen – etwa wenn die Herzfrequenz zu hoch wird. Diese Entscheidungen werden mit Hilfe der aktuellen Sensorwerte sowie der Informationen über den Anwender (z. B. Geschlecht) getroffen.

2.23 Ubidoo (2007)

In ihrem Artikel beschreiben Stahl u. a. den positionsabhängigen Task-Planer *Ubidoo* [28]. Die Anwendung stellt eine kontextsensitive ToDo-Liste dar, welche – in Kombination mit einer Routenplanung – jene Tasks (z. B. „Milch kaufen“) herausfiltert, die in der näheren Umgebung abgearbeitet werden können. Außerdem nimmt der web-basierter Ansatz Rücksicht auf den verfügbaren Zeitslot bis zum nächsten Fixtermin, Öffnungszeiten von Geschäften und benötigte Zeit, um den jeweiligen Ort zu erreichen. Um dies zu ermöglichen, werden die Tasks mit Orten (z. B. Adresse) oder Kategorien (z. B. Geschäftstyp) verknüpft, wobei das Ontologie-System *UbiWorld* [27] zum Einsatz kommt. Die kontextabhängige Adaption bezieht sich auf die aktuelle Task-Liste und auszuführende Alarme. Dabei wird die aktuelle Position des Benutzers sowie die aktuelle Zeit herangezogen, um jene Tasks zu filtern, die im näheren Umkreis bzw. bis zum nächsten Fixtermin abgearbeitet werden können.

Die Adaption bzw. Selektion der Tasks erfolgt mit Hilfe eines Routenplanungs-Algorithmus. Dieser berechnet die geografischen Distanzen zwischen aktueller Position und der

jeweiligen Task-Position sowie die benötigte Zeit, um dorthin zu gelangen. Ausgehend davon werden nur jene Tasks visualisieren, für die eine Bearbeitung in der aktuellen Situation sinnvoll erscheint.

2.24 Conny (2008)

Dürr u. a. beschreiben in ihrem Artikel [11] eine Möglichkeit, mit der das für Instant-Messaging-Dienste verwendete XMPP-Protokoll (Extensible Messaging and Presence Protocols) um kontextbezogene Adressierungskonzepte erweitert werden kann. Der entwickelte Prototyp basiert auf der Nexus-Plattform [19] und ermöglicht es, Nachrichten nur an Personen zu verwenden, welche sich in einer definierten Kontextsituation befinden.

Dabei erfolgt also eine Adaption des Systems hinsichtlich Kommunikationspunkte – nur an diejenigen Benutzer, die sich in einer bestimmten Situation befinden (z. B. alle Personen in Hagenberg) wird die kontextsensitive Nachricht verschickt. Dabei versieht der Sender der Nachricht diese mit diversen Attributen, welche gemeinsam den gewünschten Kontext beschreiben. An einen Server übermittelt, wird die Nachricht von dort aus an jene Benutzer weitergeleitet, deren Kontext dem definierten entsprechen.

2.25 Tacticycle (2009)

Das von Poppinga u. a. entwickelte *Tacticycle* [22] ist ein mobiles, kontextsensitives System, welches Touristen auf ihrer Fahrradtour per PDA begleitet und Orientierungshilfen bietet. Außerdem gibt die Anwendung dem Benutzer über zwei vibrationsfähige Griffe taktile Auskünfte – etwa um auf nahe gelegene Sehenswürdigkeit hinzuweisen.

Tacticycle verwendet keine Regel-Mechanismen, sondern manuell definierte Bedingungen, um die kontextsensitiven Informationen an den Benutzer weiterzuleiten.

3. KONTEXT-ADAPTION: AUSWERTUNG

Die soeben vorgestellten praxisnahen Projekte bilden die Basis für die Auswertung zweier wichtiger Aspekte:

- Welche Applikationsbereiche werden aufgrund der aktuellen Kontextsituation adaptiert?
- Wie erfolgt diese Anpassung?

Tabelle 1 zeigt eine Auswertung der Projekte hinsichtlich verwendeter Adaptionarten. Dabei ist ersichtlich, dass die häufigste Art, kontextbezogene Informationen in die Anwendung zu integrieren, die der Informationsadaption ist. Dies hat den Grund, dass dies technisch relativ einfach ist – etwa können aus Textinformationen bestimmte Abschnitte entfernt werden. Die Darstellung der Informationen wird ebenfalls von den meisten Projekten hinsichtlich aktueller Situation angepasst – sei es beispielsweise durch die Visualisierung des korrekten Kartenausschnitts. Weniger oft wird das Verhalten der Applikation verändert. Denkbar ist, dass das Verhalten eher schwierig angepasst werden kann bzw. dies in den meisten der vorliegenden Projekte nicht unbedingt sinnvoll gewesen wäre.

²Informationsgehalt

³Verhalten

⁴Darstellung

	Inf. ²	Verh. ³	Darst. ⁴
ParcTab	•		
C-MAP	•		
Conference Assistant	•		
GUIDE	•		•
News Dude	•		
CybreMinder	•		•
ARREAL			•
CRUMPET	•		
HyperAudio	•		•
HIPS	•		•
PinPoint	•		•
Chisel	○	○	○
Gulliver's Genie	•		•
LOL@	•		•
mobiDENK	•		•
Sightseeing4U	•		
CASS	○	○	○
COMPASS	•		
m-ToGuide	•		•
xPOI			•
Chameleon			•
ContextL		•	
MOPET		•	
Ubidoo	•		
Conny		•	
Tacticycle	•		

Table 1: Verwendete Adaptionarten

Zwei Projekte – Chisel und CASS – werden eher als Frameworks angesehen um kontextabhängige Systeme zu entwickeln und beschreiben kein explizites Projekt. Dies bedeutet, prinzipiell ist es damit möglich, alle drei Arten der Kontext-Adaption umzusetzen.

Die Anpassungen in Darstellung, Informationsgehalt und Verhalten können unterschiedlich erfolgen. In Tabelle 2 wurde aufgelistet, welche Konzepte die vorgestellten Projekte verwenden, um die Anwendung aufgrund der aktuell herrschenden Kontextsituation zu adaptieren. Deutlich am häufigsten werden die Eigenschaften des Benutzers in Form von Profilen verwendet, um die Adaption zu verwirklichen. Dabei kommen zum Teil lernende Methoden zum Einsatz, die diese Daten automatisch erfassen – als tolle Alternative zum ausfüllbaren Formular. Zu beachten ist, dass das Benutzerprofil selbst keine kontextsensitive Adaption durchführen kann – oftmals werden Filtermechanismen verwendet, um bestimmte Inhalte passend zum Profil des Users anzupassen.

Ebenfalls oft eingesetzt werden regelbasierte Mechanismen (statisch bzw. dynamisch), um die Anwendung aufgrund der aktuellen Kontextsituation zu adaptieren. Daneben gibt es noch weniger häufig eingesetzte Methoden wie URL-Kodierung, Layer- und Ontologie-basierte Ansätze. All diese Konzepte werden in den folgenden Abschnitten nochmals kurz erläutert.

3.1 Verwendete Adaptionskonzepte

3.1.1 Benutzerprofil

Die Verwendung des Benutzerprofils stellt eine relativ ein-

	Benutzerprofil	Statische Regeln	Dynamische Regeln	URL-Kodierung	Layer	Ontologie
ParcTab	•					
C-MAP	•					
Conference Assistant	•					
GUIDE				•		
News Dude	•					
CybreMinder			•			
ARREAL		•				
CRUMPET	•					
HyperAudio			•			
HIPS			•			
PinPoint				•		
Chisel			•			
Gulliver's Genie	•					
LOL@				•		
mobiDENK	•					
Sightseeing4U	•					
CASS			•			
COMPASS	•					
m-ToGuide	•					
xPOI			•			
Chameleon	•					
ContextL					•	
MOPET	•	•				
Ubidoo						•
Conny	•					
Tacticycle		•				

Table 2: Verwendete Adaptionskonzepte

fache Möglichkeit dar, Kontextinformationen zu sammeln. Dabei gibt es neben der klassische Methodik, die Daten per Formular vom Benutzer angeben zu lassen, auch lernende Systeme, welche die Präferenzen des Anwenders automatisch (über längeren Zeitraum) erfassen. Zur einfachen Weiterverarbeitung werden oftmals definierte Werte (z. B. Interessensauswahl anhand bestehender Liste) vorgegeben.

Nach der Erfassung der Daten – egal auf welche Art und Weise – erfolgt die Adaption oftmals mit definierten Regeln oder Filter-Mechanismen. Ein Beispiel dazu wäre die Adaption von Nachrichten auf Basis des Benutzerinteresses. Dazu würden die Informationen mit den passenden vordefinierten Bereichen verknüpft. Bei der Abfrage von neuen Nachrichten könnten durch einen Filter nur jene Nachrichten selektiert werden, welche zum Benutzerprofil passen.

3.1.2 Statische Regeln

Dieses Adaptionskonzept bezieht sich auf in den Quellcode eingebettete Regeln. Diese können als simple **if**- oder **switch**-Ansätze implementiert sein und sind – aufgrund der statischen Eigenschaft – relativ schwierig anzupassen bzw. zu erweitern. Dabei muss die gesamte Anwendung neu kompiliert und ausgeliefert werden.

3.1.3 Dynamische Regeln

Anders als statische sind dynamische Regeln nicht direkt in den Quellcode eingebettet sondern meist außerhalb der Anwendung (z. B. durch externe Dateien) definiert. Ein solches Konzept beschreiben beschreiben Beer u. a. [2] – dabei

werden so genannte ECA-Regeln (Event-Condition-Action-Regeln) verwendet, um auf Änderungen der Kontextsituation einzugehen. Bei der Änderung von Objektattributen, welche Kontextinformationen enthalten, werden Events ausgelöst, welche mit Regeln in Verbindung stehen. Somit kann auf Kontextänderungen relativ einfach reagiert werden. Die Idee, auch zur Laufzeit neue Regeln in die Anwendung aufzunehmen, verleiht dem Ansatz zusätzliche Dynamik. Ein Beispiel für eine Regeldefinition beinhaltet Listing 1.

Listing 1: ECA-Regel-Beispiel

```
rules for <HeartPatients> {
  on HeartMonitor . Alarm () {
    <EmergencyDispatcher> . EmergencyCall:
      Alarm ( Patient );
  }
}
```

Dabei würde für jedes *HeartPatient*-Objekt bei einem auftretenden *Alarm* des *HeartMonitors* ein Notruf abgesetzt werden.

3.1.4 URL-Kodierung

Um Daten bereits vorselektiert von einem Dienst abzurufen, kann der Ansatz der URL-Kodierung verwendet werden. Dabei wird der Aufruf an den Server-Dienst mit Parametern (z. B. die aktuelle GPS-Position) versehen. Der Dienst selektiert aufgrund dieser zusätzlichen Angaben die gewünschten Daten (z. B. den umliegenden Kartenausschnitt) und retourniert das Ergebnis an den Aufrufer. Somit wird die Adaption

on der Informationen nicht direkt durchgeführt, sondern an einen externen Dienst verlagert.

3.1.5 Layer

Architektur-orientierte Ansätze, welche Layer zur Adaption von Anwendungen verwenden, werden weniger häufig eingesetzt. Im Rahmen dieses Artikels konnte nur ein Projekt (*ContextL*) gefunden werden, welches nach diesem Prinzip arbeitet. Dabei werden die zu behandelnden Kontextsituationen in verschiedene, klar abgetrennte Teilbereiche der Applikation (Layer) aufgeteilt. Nur der aktive Layer erhält die vom Benutzer gestellte Anfrage und verarbeitet diese bzw. gibt eine Ergebnismenge zurück.

3.1.6 Ontologie

Ebenfalls nur eines der vorgestellten Projekte (*Ubidoo*) nutzt Ontologien zur Kontext-Adaption. Diese werden verwendet, um Relationen zwischen Aufgaben und Orten bzw. Aufgaben-Kategorien herzustellen. In weiterer Folge ist es möglich, in bestimmten Situationen jene Aufgaben auszuwählen, welche zu diesem Zeitpunkt relevant sind.

4. RESÜMEE

Im Rahmen dieses Artikels wurde gezeigt, welche Möglichkeiten zur Adaption von Anwendungen in existierenden Projekten verwendet werden. Dabei wurde zu Beginn erläutert, was Kontext-Adaption bedeutet – die Art und Weise wie sich ein System an die aktuelle Situation des Benutzers anpasst.

Eine Vielzahl an existierenden Projekten wurde – bezogen auf die verwendeten Adaptions-Arten und -Konzepte – untersucht und präsentiert. Die Auswertung dieser Ergebnisse zeigt, dass zumeist Informationsgehalt und Darstellung – unter Verwendung von Benutzerprofilen oder Regel-Mechanismen – angepasst werden.

Die Suche nach neuen Mechanismen zur Integration von kontextsensitiven Aspekten in Anwendungen – etwa mit Hilfe von modellgetriebenen Techniken oder Statecharts – wäre der logische nächste Schritt.

5. REFERENZEN

- [1] J. Baldzer, S. Boll, P. Klante, J. Krösche, J. Meyer, N. Rump, and A. Scherp. Location-aware mobile multimedia applications on the nccimon platform. *Proceedings of the 2nd Symposium on Informationssysteme für mobile Anwendungen (IMA 2004)*, pages 318–334, 2004.
- [2] W. Beer, V. Christian, A. Ferscha, and L. Mehrmann. Modeling context-aware behavior by interpreted eca rules. In *In Proceedings of the International Conference on Parallel and Distributed Computing EUROPAR '03*, pages 26–29. Springer Verlag, 2003.
- [3] S. C. Boll. Multimedia sightseeing 4 u - what web services can do for personalized multimedia applications. In *The 7th World Multiconference on Systemics, Cybernetics and Informatics*, pages 220–225, 2008.
- [4] F. Buttussi and L. Chittaro. Mopet: A context-aware and user-adaptive wearable system for fitness training. In P. Lukowicz, editor, *Artificial Intelligence in Medicine*, volume 42, pages 153–163. Elsevier, 2008.
- [5] K. Cheverst, N. Davies, K. Mitchell, and A. Friday. The role of connectivity in supporting context-sensitive applications. In *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pages 193–207, London, UK, 1999. Springer-Verlag.
- [6] P. Costanza, R. Hirschfeld, and W. D. Meuter. Efficient layer activation for switching context-dependent behavior. In *In JMLC '06: Proceedings of the Joint Modular Languages Conference*, pages 84–103. Springer, 2006.
- [7] B. Desmet, J. Vallejos, and P. Costanza. Layered design approach for context-aware systems. In *in 1st VaMoS 07*, 2007.
- [8] A. K. Dey and G. D. Abowd. Cybreminder: A context-aware system for supporting reminders. In *HUC '00: Proceedings of the 2nd international symposium on Handheld and Ubiquitous Computing*, pages 172–186, London, UK, 2000. Springer-Verlag.
- [9] A. K. Dey, D. Salber, G. D. Abowd, and M. Futakawa. An architecture to support context-aware applications. Technical report, Georgia Institute of Technology, 1999.
- [10] A. K. Dey, D. Salber, G. D. Abowd, and M. Futakawa. The conference assistant: Combining context-awareness with wearable computing. In *ISWC '99: Proceedings of the 3rd IEEE International Symposium on Wearable Computers*, page 21, Washington, DC, USA, 1999. IEEE Computer Society.
- [11] F. Dürr, J. Palauro, L. Geiger, R. Lange, and K. Rothermel. Ein kontextbezogener Instant-Messaging-Dienst auf Basis des XMPP-Protokolls. In *5. GI/ITG KuVS Fachgespräch Ortsbezogene Anwendungen und Dienste. Nürnberg. September 2008*, volume 42 of *Sonderdruck Schriftenreihe der Georg-Simon-Ohm-Hochschule Nürnberg*, pages 23–28, Nürnberg, September 2008. Georg-Simon-Ohm-Hochschule Nürnberg.
- [12] P. Fahy and S. Clarke. CASS – A middleware for mobile context-aware applications. In *Workshop on Context Awareness, MobiSys*, 2004.
- [13] S. Fels, Y. Sumi, T. Etani, N. Simonet, K. Kobayashi, and K. Mase. Progress of c-map: a context-aware mobile assistant. In *In Proceedings of AAAI 1998 Spring Symposium on Intelligent Environments*, pages 60–67. AAAI Press, 1998.
- [14] S. Gassner. "news dude- ein persönlicher nachrichten-agent, der spricht, lernt und erklärt. In A. Henrich and K. Morgenroth, editors, *Kontextbasiertes Information Retrieval: Methoden und Anwendungen*, 2006.
- [15] J. Keeney and V. Cahill. Chisel: A policy-driven, context-aware, dynamic adaptation framework. In *POLICY '03: Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks*, page 3, Washington, DC, USA, 2003. IEEE Computer Society.
- [16] J. Krösche, J. Baldzer, and S. Boll. MobiDENK – mobile multimedia in monument conservation. *IEEE MultiMedia*, 11(2):72–77, 2004.
- [17] J. Krösche and S. Boll. The xPOI concept. In *Proc. LOCA '05*, pages 113–119. Springer Berlin /

- Heidelberg, 2005.
- [18] I. Mohomed, J. C. Cai, and E. de Lara. Urica: Usage-aware interactive content adaptation for mobile devices. *SIGOPS Oper. Syst. Rev.*, 40(4):345–358, 2006.
- [19] D. Nicklas, M. Großmann, T. Schwarz, S. Volz, and B. Mitschang. A model-based, open architecture for mobile, spatially aware applications. In *SSTD '01: Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases*, pages 117–135, London, UK, 2001. Springer-Verlag.
- [20] G. M. P. O'Hare and M. J. O'Grady. Gulliver's genie: A multi-agent system for ubiquitous and intelligent content delivery. *Computer Communications*, 26:1177–1187, 2003.
- [21] D. Petrelli, E. Not, M. Zancanaro, C. Strapparava, and O. Stock. Modelling and adapting to context. *Personal Ubiquitous Comput.*, 5(1):20–24, 2001.
- [22] B. Poppinga, M. Pielot, and S. Boll. Tacticycle: a tactile display for supporting tourists on a bicycle trip. In *MobileHCI '09: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 1–4, New York, NY, USA, 2009. ACM.
- [23] S. Poslad, H. Laamanen, R. Malaka, A. Nick, P. Buckle, and A. Zipl. Crumpet: creation of user-friendly mobile services personalised for tourism. *IEE Conference Publications*, 2001(CP477):28–32, 2001.
- [24] J. Roth. Context-aware web applications using the pinpoint infrastructure. In *IADIS Intern. Conference WWW/Internet 2002*, pages 13–15. IADIS press, 2002.
- [25] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *In Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 85–90. IEEE Computer Society, 1994.
- [26] W. Schwinger, C. Grün, B. Pröll, W. Retschitzegger, and A. Schauerhuber. Context-awareness in mobile tourism guides – a comprehensive survey. Technical report, Johannes Kepler Universität Linz, Technische Universität Wien, 2005.
- [27] C. Stahl and D. Heckmann. Using semantic web technology for ubiquitous location and situation modeling. In *The Journal of Geographic Information Sciences CPGIS*. Berkeley, 2004.
- [28] C. Stahl, D. Heckmann, T. Schwartz, and O. Fickert. Here and now: A user-adaptive and location-aware task planner. In *Proceedings of the International Workshop on Ubiquitous and Decentralized User Modeling (UbiDeUM'2007)*, pages 52–63, 2007.
- [29] M. Umlauft, G. Pospischil, Günther Niklfeld, and E. Michlmayr. LOL@, a mobile tourist guide for UMTS. *Information Technology & Tourism*, 5:151–164, 2003.
- [30] M. van Setten, S. Pokraev, and J. Koolwaaij. Context-aware recommendations in the mobile tourist application compass. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 235–244, 2004.
- [31] W. Wahlster, J. Baus, C. Kray, and A. Krüger. REAL: Ein ressourcenadaptierendes mobiles Navigationssystem. *Informatik Forschung und Entwicklung*, 2001.
- [32] R. Want, B. N. Schilit, N. I. Adams, R. Gold, K. Petersen, D. Goldberg, J. R. Ellis, and M. Weiser. The parctab ubiquitous computing experiment. *IEEE PERSONAL COMMUNICATIONS*, 2:28–43, 1995.