

Mobile und kontextbezogene Datenbanktechnologien und Anwendungen

Motivation und Anwendungen

Simon Feil
TU-Kaiserslautern

Abstract. Diese Ausarbeitung ist der erste Teil des Informatik Seminars des Lehrgebiets Informationssysteme über mobile und kontextbezogene Datenbanktechnologien und Anwendungen. Dieser Teil schafft die notwendigen Grundlagen, in dem wichtige Begriffe und Techniken erläutert werden.

1 Einleitung

Computer sind aus dem heutigen Leben nicht mehr wegzudenken, teilweise beeinflussen sie unseren Alltag, ohne dass wir es überhaupt bemerken. Sie werden immer kleiner und integrieren sich besser in unsere Umgebung. Aktuelle Mobiltelefone bieten deutlich mehr Funktionen als die einfache Telefonie-Option. Sie sind kleine tragbare Computer, die mit aktuellster Technik wie Bluetooth, Wlan, GPS, GPRS oder UMTS ausgestattet sind. Die Verarbeitung von Mobiltelefonen ist aufgrund der fallenden Preise enorm. Mittlerweile geht in führenden Industriestaaten sogar der Trend zum Dritthandy [17]. Nur was lässt sich mit diesen Technologien alles realisieren? Derzeit werden Mobiltelefone hauptsächlich zum Versenden von Nachrichten, z.B. Email oder SMS, Planen von Terminen oder zum Surfen im WorldWideWeb genutzt. Die genutzten Anwendungen zeigen in der Regel keine Intelligenz auf und interagieren mit dem Benutzer nur auf Verlangen.

Diese Ausarbeitung soll einen Überblick geben, was mit heutiger Computertechnik möglich ist und wie Computer den Benutzer in seiner Arbeit unterstützen können.

In Kapitel 2 werden grundlegende Fachwörter wie z.B. “Ubiquitous Computing” oder “Pervasive Computing” definiert und deren Unterschiede erläutert. In diesem Zusammenhang werden weitere Begriffe wie Kontext und Kontextbewusstsein eingeführt. Kapitel 3 stellt einen Überblick über verschiedene Grundlagen dar: z.B. Wie Kontext gespeichert oder verwaltet werden kann oder welche Möglichkeiten

der Positionsbestimmung es gibt. Ein praktischer Anwendungsfall von Ubiquitous Computing wird in Kapitel 4 beschrieben. Dort wird anhand eines Beispiels in einem Krankenhaus aufgezeigt, welche Vorteile, aber auch Probleme bei der Umsetzung in der Realität auftreten können. Zum Abschluss gebe ich in Kapitel 5 eine kurze Stellungnahme ab, wie weit, meiner Meinung nach, Ubiquitous Computing unsere Zukunft bestimmen wird.

2 Definitionen

In der Literatur trifft man bei der Recherche über mobile Anwendungen, die einem den Alltag erleichtern sollen, immer wieder auf Begriffe wie Ubiquitous oder Pervasive Computing. Auch der Kontext oder das Kontextbewusstsein sind Ausdrücke, die in diesem Zusammenhang auftauchen. Was darunter zu verstehen ist und in welchem Zusammenhang sie stehen, zeigen die folgenden Absätze.

2.1 Ubiquitous computing

„The most profound technologies are those that disappear“ [1]. Mit diesem Satz beginnt Mark Weiser seinen Aufsatz „The Computer for the 21st Century“ aus dem Jahre 1991 und führt damit den Begriff des Ubiquitous Computing ein. Auch der Computer- und Medienkritiker Joseph Weizenbaum meint im Jahre 2005: „Das Bewusstsein, dass Computer überall stecken, wird allmählich verschwinden. Wir werden in zehn oder 15 Jahren nicht mehr über Computer reden - die sind dann einfach überall“ [2]. Der Personal Computer (PC) wie er in der heutigen Zeit an jedem Arbeitsplatz steht wird langfristig komplett verschwinden. Der Weg zu immer kleineren PCs, leistungsfähigen mobilen Geräten wie Handys oder PDAs ist nur eine Vorstufe zum „unsichtbaren Computer“, der komplett im Hintergrund arbeitet. Ziel ist es, den Benutzer in Alltagssituationen durch PCs zu unterstützen, ohne dass er den Dienst bewusst in Anspruch nehmen muss.

In diesem Zusammenhang tritt auch der Begriff des „Pervasive Computing“ auf. Bei der Übersetzung ins Deutsche gibt es zwischen den beiden Wörtern „pervasive“ (überall vorhanden) und „ubiquitous“ (allgegenwärtig) keinen Unterschied. In der Informatik unterscheiden sie sich jedoch in einigen Feinheiten. Die Abbildung 1 versucht die Unterschiede dieser Ausdrücke zu verdeutlichen [3].

Unter dem traditionellen Computer versteht man den handelsüblichen PC, der auf jedem Schreibtisch eines Büros steht. Er steht fest an einem bestimmten Ort und kann von dort nur schwer bewegt werden. Integriert ist der Computer in den Arbeitsplatz nicht, er steht als Kasten auf bzw. unter dem Schreibtisch und wird somit von Jedermann bewusst wahrgenommen. Die Anwendungsmöglichkeit besteht nur aktiv, indem man den PC einschaltet und Anfragen in der Regel über Tastatur oder Maus eingibt.

Unter tragbaren Computern versteht man einen Desktop PC, der auf ein handliches Format geschrumpft ist. Die Funktionen entsprechen im Wesentlichen denen eines traditionellen Computers, bei dem auf Grund der verringerten Größe, die Möglichkeit des problemlosen Transportes besteht. Beispiele hierfür sind Notebooks, PDAs oder aktuelle Handys.

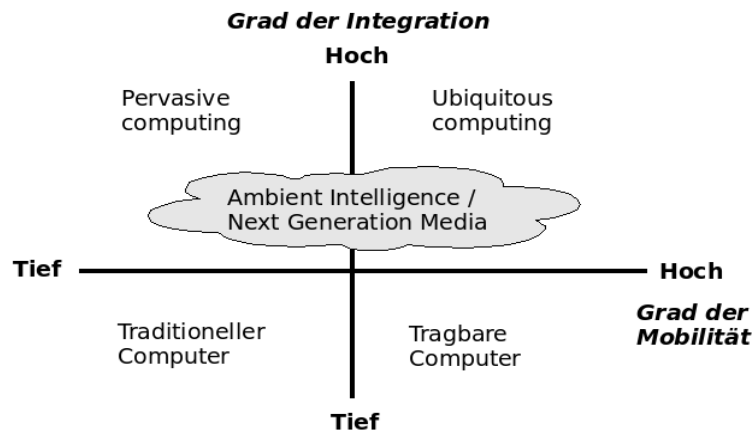


Abbildung 1: Grad der Integration bzw. Mobilität

Bei „Pervasive computing“ wird der PC unsichtbar, ist also vollkommen in seine Umgebung integriert. Dies setzt voraus, dass die Umgebung mit entsprechenden Sensoren zur Wahrnehmung ausgestattet ist. Durch diese Überwachung kann das System den Benutzer durch intelligente Hilfestellungen in seiner Arbeit automatisch unterstützen. Der Mobilitätsgrad ist jedoch weiterhin gering, der Computer ist fest an den Standort gebunden.

Das „Ubiquitous computing“ ist die Weiterführung des „Pervasive computing“. Dabei integriert sich der Computer vollständig in das Leben des Benutzers und ist somit als ständiger Begleiter immer dabei. Hier wird versucht durch Erkennen der Umgebung und der Situation dem Benutzer intelligente Hilfestellungen zu bieten. Der Computer ist nicht mehr nur in die Umgebung integriert, sondern in den Benutzer selbst, z.B. in seine Kleidung.

Das Sixth Framework Programme der Europäischen Union, das für die Entwicklung und Bekanntmachung neuer Technologien gegründet wurde, hat den Begriff der „Ambient Intelligence“ [4] (Deutsch: Umgebungsintelligenz) eingeführt, worunter man eine Mischung aus „Pervasive Computing“ und „Ubiquitous Computing“ versteht. Das Bundesministerium für Wirtschaft und Technologie der Bundesrepublik Deutschland hat hierfür den Begriff „Next Generation Media“ [5] ins Leben berufen.

Die eingeführten Ausdrücke der Europäischen Union und der Bundesregierung zeigen, dass die Grenzen zwischen „Pervasive Computing“ und „Ubiquitous Computing“ fließend sind. Auch in der Literatur werden Begriffe nicht immer

differenziert und als Synonym verwendet. Da die Grenzen zwischen diesen Begriffen fließend sind, wird in dieser Ausarbeitung „Ubiquitous computing“ als umfassender Oberbegriff angenommen.

Im Wesentlichen wird es durch folgende Merkmale gekennzeichnet [6]:

- *Miniaturisierung*: Die verwendete Computerhardware ist klein und dadurch leicht zu transportieren. Wie in Abbildung 1 verdeutlicht besteht ein hoher Mobilitätsgrad.
- *Einbettung*: Die Hardware wird in Gegenstände und Umgebungen integriert. Der Computer wird nicht mehr als solcher wahrgenommen. Es besteht ein hoher Integrationsgrad.
- *Vernetzung*: Zwischen den einzelnen Komponenten erfolgt ein, meist drahtloser, Datenaustausch, um Informationen untereinander auszutauschen.
- *Allgegenwart*: Die Computer sind an den Benutzer, z.B. an seine Kleidung, gebunden und somit jederzeit für den Benutzer verfügbar.
- *Kontextbewusstsein*: Die Umgebung des Benutzers wird mit Sensoren überwacht, so dass der Computer der Umgebung oder Situation angepasst agieren kann (Vgl. Kapitel 2.3).

2.2 Kontext

Wie in Kapitel 2.1 bereits erwähnt ist Kontextbewusstsein eine der Charakteristiken des Ubiquitous Computing. Bevor dieser Begriff in Kapitel 2.3 erläutert wird, soll zuerst geklärt werden, was unter dem Ausdruck Kontext an sich zu verstehen ist.

Das Wort Kontext stammt von dem lateinischen Wort contextus ab und bedeutet "Zusammenhang" oder "feste Verknüpfung". Im Bezug auf Ubiquitous Computing trifft diese Definition nur teilweise zu und wird folgendermaßen definiert:

Kontext sind alle Informationen, die benutzt werden können, um die Situation einer Entität zu charakterisieren. Eine Entität kann eine Person, ein Ort oder ein Objekt sein, das relevant für die Interaktion zwischen Benutzer und Anwendung ist – Computer und Anwendung selbst eingeschlossen [7].

Der Begriff Kontext lässt sich auf viele Bereiche mit unzähligen Ausprägungen anwenden [8]:

- geographisch (z.B. Land, Straße, Gebäude, Stockwerk, Büro)
- physisch (z.B. Lichtverhältnisse, Geräuschpegel, Temperatur, Beschleunigung, Neigung)
- organisatorisch (z.B. Institution, Abteilung, Gruppe, Projekt)
- sozial (z.B. Familie, Freunde, Mitarbeiter, Familienstand)
- emotional (z.B. Pulsschlag, Hautwiderstand)
- technologisch (z.B. Netzwerkzugang, Netzwerkbandbreite, Netzwerkverzögerung)
- Benutzer (z.B. Profile, Ort, Fähigkeiten, Rollen, Zugangsrechte)
- Aufgaben (z.B. Dokumentieren, Programmieren, Haus bauen)
- Aktionen (z.B. tippen, lesen, laufen, sitzen, erzählen)
- Zeit (z.B. Wochentag, Woche, Monat, Saison)

Die Kombination dieser Werte ermöglicht es, die Umgebung genau zu analysieren und entsprechend zu interpretieren.

2.3 Kontextbewusstsein

Kontextbewusstsein ist die Fähigkeit eines Gerätes oder eines Programmes sich an ein, oft dynamische, Umfeld anzupassen [7]. Dies geschieht durch Beobachten der Umgebung, z.B. mit Sensoren. Es gilt den notwendigen Kontext zu erfassen, um die aktuelle Situation zu erkennen und entsprechend darauf reagieren zu können. Abhängig vom Zweck eines kontextbewussten Systems spielen nur bestimmte Kontextinformationen eine Rolle. So ist z.B. für ein Heizungssystem wichtig wie warm es gerade in einem Büro ist, wer sich in dem Büro zur Zeit aufhält und welches die bevorzugte Raumtemperatur dieser Person ist. Ob diese Person verheiratet ist oder in der Lage ist, ein Java Programm zu schreiben ist ebenfalls Kontext, aber in diesem Zusammenhang irrelevant.

Kontextbewusstsein wird im Wesentlichen genutzt zur:

- Speichern von Informationen
- Präsentation von Informationen
- Ausführen von automatischen Aktionen auf Grund bestimmter Informationen.

2.4 Kontextbezogene Anwendungen

Kontextbewusstsein ist notwendig, da der Mensch sich mit dem Ausführen mehrerer Tätigkeiten gleichzeitig schwer tut und nur eine begrenzte Aufnahmefähigkeit besitzt. Deswegen soll er durch kontextbewusste Computersoftware in diesen Arbeiten unterstützt werden, indem er die Arbeit entweder abgenommen bekommt oder ihm Hilfestellungen angeboten werden. Solche Programme nennen sich kontextbezogene Anwendungen.

Im Folgenden werden zwei Möglichkeiten dargestellt, kontextbezogene Anwendungen zu untergliedern:

- die Gliederung in vier Kategorien
- die Gliederung nach Aktivitätsgrad.

2.4.1 Gliederung in vier Kategorien

Abbildung 2 zeigt die Gliederung von kontextbezogenen Anwendungen, abhängig davon, ob sie manuell oder automatisch ausgeführt werden und ob sie Informationen ausgeben oder einen Befehl ausführen [9].

	Manuell	Automatisch
Information	Benachbarte Auswahl	kontextabhängige Umgestaltung
Befehl	kontextabhängige Befehle	kontextabhängige Handlungen

Abbildung 2: Gliederung von kontextbezogenen Anwendungen in vier Kategorien

Benachbarte Auswahl

Bei der ersten Kategorie wird die Anwendung per Hand ausgeführt und es werden lediglich kontextbezogene Informationen angezeigt, die dem Benutzer hilfreich sein können. Der Anwender trifft über die Oberfläche des Programms eine Auswahl über das gewünschte Objekt. Daraufhin werden die Objekte, die am nächsten zu erreichen sind, angezeigt.

Beispielsweise kann ein Anwender durch Eingabe der Auswahl „Drucker“ und „Zimmernummer“ eine Liste bekommen, in der alle Drucker des Gebäudes aufgelistet sind, sortiert nach der Entfernung von dem angegebenen Büro. Die

sortierte Druckerliste entspricht dann der kontextbezogenen Information, die dem Benutzer die Arbeit erleichtert.

Kontextabhängige Umgestaltung

In diesem Fall gibt die Software die Informationen nicht auf explizite Anfrage des Benutzers aus, sondern sie verändert sich automatisch und bringt somit Komponenten in den Vordergrund, die für den Anwender relevant sein könnten und versteckt Objekte, die er nicht benötigt. Komponenten werden dadurch abhängig des Kontextes automatisch hinzugefügt, entfernt oder durch andere ersetzt.

Als Beispiel kann hier die Anbindung eines Mobilgerätes an das Internet genommen werden. Befindet sich der Benutzer in seinem Büro, wird das lokale Netzwerk der Firma genutzt. Verlässt er das Gebäude, wird nach verfügbaren WLANs gesucht oder eine Verbindung über das Handy hergestellt. Der Anwender muss sich nicht darum kümmern wie er ins Internet kommt, es wird automatisch die Beste – je nach Wunsch, die günstigste, schnellste, stabilste – Verbindung gewählt.

Kontextbezogene Befehle

Bei kontextbezogenen Befehlen wird auf Anweisung des Benutzers ein bestimmter Befehl ausgeführt. Je nach Kontext kann jedoch dieselbe Funktion verschiedene Ergebnisse liefern.

Hier lässt sich wieder das Druckerbeispiel aus "Benachbarte Auswahl" verwenden. Der Benutzer bekommt in diesem Fall nicht die sortierte Liste der Drucker angezeigt, sondern er betätigt einfach den Drucken-Button. Je nachdem, wo er sich gerade befindet wird der passende Drucker ausgewählt und der Druckauftrag übertragen. Somit benutzt der Anwender immer dieselbe Funktion, jedoch, abhängig des Kontextes, mit anderen Ergebnissen.

Kontextabhängige Handlungen

Bei kontextabhängigen Handlungen führt das System ohne expliziten Aufruf des Anwenders eine Handlung aus. Hierbei handelt es sich um simple IF-THEN Anweisungen, die abhängig von definierten Parametern spezielle Aufgaben ausführen. So kann eine Liste mit Aufgaben festgelegt werden, die ausgeführt werden soll, wenn ein bestimmtes Ereignis in Kraft tritt.

Beispiel: IF (Papiervorrat<50) THEN order(Papier)

Wenn der Papiervorrat eines Druckers geringer als 50 Blatt ist, soll der Drucker eine Papierbestellung an die Büromittelabteilung abschicken. Die Bestellung erfolgt automatisch und muss nicht erst vom Benutzer bestätigt werden.

2.4.2 Gliederung nach Aktivitätsgrad

Ein weiterer Ansatz kontextbezogene Anwendungen zu gliedern, ist die Aufteilung nach der Aktion [16]. Unter Aktion versteht man die Reaktion, z.B. das Ausführen eines bestimmten Befehles, auf Basis von Kontextinformationen. Abbildung 3 zeigt vier verschiedene Reaktionsarten, deren Aktionen auf einer Skala von passiv zu aktiv angeordnet sind.

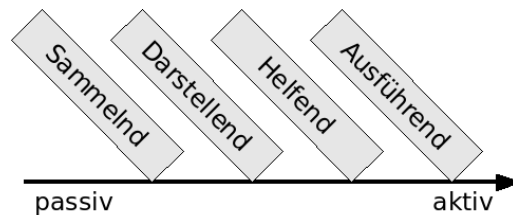


Abbildung 3: Gliederung von kontextbezogenen Anwendungen nach Aktivitätsgrad

Sammelnde Aktion: Bei der sammelnden Aktion handelt es sich um eine rein passive Handlung. Es wird nichts ausgeführt oder dem Benutzer dargestellt. Es handelt sich hierbei nur um das Speichern von Kontextdaten für eine spätere Verwendung.

Beispiel: Ein Onlineshop merkt sich welche Produkte ein Kunde sich anschaut und welche er bestellt. Daraus können später ein Kundenprofil oder andere Statistiken erstellt werden. Der Kunde bekommt davon nichts mit.

Darstellende Aktion: Aktive Aktionen finden bei der darstellenden Aktion noch nicht statt. Der Benutzer bekommt auf Grund von Kontextdaten Informationen aufbereitet und dargestellt, um Entscheidungen zu erleichtern.

Beispiel: Der Webshop schlägt dem Kunden auf Grund bereits gesammelter Informationen bestimmte Produkte vor, die für ihn interessant sein könnten. Der Benutzer hat die Wahl, sich diese Vorschläge anzusehen oder beliebig nach anderen Artikeln zu suchen.

Helfende Aktion: Bei der helfenden Aktion kommt die aktive Komponente dazu. Die Anwendung schlägt dem Benutzer bestimmte Handlungen vor, die ausgeführt werden können. Vor der Ausführung müssen sie jedoch vom Anwender bestätigt werden.

Beispiel: Der Kunde bekommt beim Besuch des Webshops automatisch Artikel in seinen Warenkorb gelegt, die für ihn interessant sein könnten. Der Warenkorb muss nur vom Kunden bestätigt werden und dann wird die Bestellung versandt.

Ausführende Aktion: In der Kategorie ausführende Aktion erfolgt die Handlung ohne Aufforderung des Benutzers. Es wird ohne Nachfragen eine selbstständige Reaktion des Systems ausgeführt, die die Anwendung auf Grund der Kontextinformationen für angemessen hält.

Beispiel: Der Webshop schickt dem Kunden monatlich ein Paket mit Produkten, die für ihn interessant sein könnten. Der Kunde hat die Bestellung nicht bestätigt oder per Hand getätigt.

Bei dieser Gliederung stehen die Konsequenzen im Vordergrund. Bei einer passiven Aktion sind die Konsequenzen überschaubar, da vom Benutzer alles kontrolliert werden kann. Je aktiver eine Handlung ist, je mehr Gefahren gehen von den Konsequenzen aus. In dem hier erwähnten Beispiel eines Webshops sind die schlimmsten Konsequenzen, dass ein Kunde Produkte geschickt bekommt, die er nicht benötigt. In einem anderen Szenario, wie z.B. das Krankenhausbeispiel in Kapitel 4, können die Folgen weitaus weitreichender sein.

3 Modelle

Dieses Kapitel stellt einige ausgewählte Modelle vor, die zeigen wie die im vorherigen Kapitel erläuterten theoretischen Grundlagen in der Praxis umgesetzt werden können.

3.1 Positionsbestimmung

Eine der wichtigsten Kontextinformationen ist der Ort. Ein System muss in der Regel wissen, wo sich der Benutzer gerade aufhält, um hilfreiche Aktionen ausführen zu können. Dazu muss die Position des Benutzers ermittelt werden. Die Positionsbestimmung kann auf drei verschiedenen Arten erfolgen [10]:

- physisch
- virtuell
- logisch.

Physische Positionsbestimmung

Bei der physischen Positionsbestimmung wird mit spezieller Hardware die aktuelle Position bestimmt. Dies ist zum Beispiel mit Hilfe des Global Position Systems (GPS) oder mit Mobilfunktelefonen möglich. Abhängig von den Anforderungen bieten sich verschiedene Techniken an. Die Anforderungen können z.B. sein:

Umgebung: GPS funktioniert hauptsächlich im Freien, da Sichtkontakt benötigt wird. Somit ist der Einsatz in Gebäuden hinter dicken Decken und Wänden kaum möglich. Der Einsatz von Bluetooth oder RFID ist in Räumen besser geeignet, lässt sich jedoch auf großen Strecken nur schwer realisieren.

Genauigkeit: Die Abweichungen können von wenigen cm (RFID) über einige Meter (GPS) bis hin zu hunderten von Metern sein (Mobilfunkzellen).

Preis: Die Herstellungskosten von RFID Chips betragen mittlerweile nur noch wenige Cents, es muss jedoch auch die entsprechende Umgebung zum Erkennen dieser Chips vorhanden sein. GPS-Empfänger und Mobiltelefone sind zwar deutlich teurer, die benötigte Infrastruktur (Satelliten/Handymasten) ist jedoch bereits flächendeckend vorhanden.

Virtuelle Positionsbestimmung

Bei der virtuellen Positionsbestimmung wird die Position nicht anhand von physischer Hardware bestimmt, sondern mit Software, die die nötigen Kontextinformationen aus anderen Quellen bezieht. Dabei gibt es folgende Möglichkeiten:

Anwendungssoftware: Durch das Durchstöbern von Terminkalendern, Reisebuchungssystemen oder Emails kann herausgefunden werden, wo der Benutzer sich gerade befindet.

Bsp.: Im Terminkalender steht 15 Uhr Treffen in Raum 434. Dann kann davon ausgegangen werden, dass sich der Benutzer um 15 Uhr auch dort befindet. Wenn der Anwender sich auf Grund diverser Umstände, wie z.B. Krankheit, nicht zu dem Treffen geht, trifft das System Entscheidungen auf Grund falscher Informationen.

Betriebssysteme: Es wird vorausgesetzt, dass ein Anwender sich vor dem Benutzen des Computers mit seinem Login anmelden muss. Somit kann man davon ausgehen, dass wenn jemand aktiv am PC arbeitet, der Benutzer sich auch vor dem PC befindet.

Bsp.: Der Benutzer ist lokal an seinem Desktop-PC im Büro angemeldet und die Maus ist aktiv. Daraus wird geschlossen, dass der Benutzer sich gerade in seinem Büro aufhält.

Netzwerk: Je nachdem wie das Netzwerk aufgebaut ist, besteht die Möglichkeit, anhand der IP herauszufinden, wo sich der Benutzer gerade befindet.

Bsp: Wenn eine Firma in Ihrem Netzwerk das Dynamic Host Configuration Protocol (DHCP) verwendet, kann eingestellt werden, dass sich der Mitarbeiter/Benutzer je nach Knoten, an dem er sich mit seinem Mobilgerät ins Netzwerk einwählt, eine andere IP bekommt. Somit lässt sich anhand der IP erkennen, in welchem Gebäudeteil, Zweigstelle, etc. der Benutzer sich gerade befindet.

Logische Positionsbestimmung

Die Ergebnisse aus physischer und virtueller Positionsbestimmung benötigen eine Logik, die die Daten anschließend auswertet. Nur die reinen Daten, wie z.B. Koordinaten oder Mobilfunkzellen, sind in der Regel nichtssagend. Erst das Auswerten der Daten und das Zuordnen einer Adresse zu den übermittelten Koordinaten, macht die Information anwendbar. Dazu werden zusätzliche Daten benötigt, die z.B. in einer Datenbank stehen können. Die Kombination aus physischer

bzw. virtueller Positionsbestimmung mit Auswertelogik und zusätzlichen Daten ergeben die logische Positionsbestimmung.

3.2 Positions-Management-Systeme

Um die in Abschnitt 3.1 angesprochene logische Positionsbestimmung durchführen zu können, müssen die Daten der einzelnen Sensoren verarbeitet werden. Dafür ist ein Positions-Management-System zuständig. Es erhält seine Informationen aus diversen Quellen und entsprechend auch in unterschiedlichen Formaten. Ziel ist es, all diese Informationen in einer einheitlichen Form aufzubereiten, so dass sie den Anwendungen zur Verfügung gestellt werden können. Dieses Management-System kann in mehrere Ebenen unterteilt werden (vgl. Abbildung 4) [10].

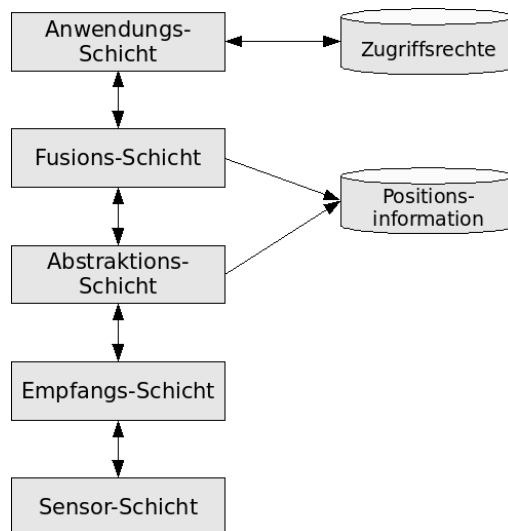


Abbildung 4: Architektur eines Positions-Management-Systems

Die unterste Ebene dieses Systems ist die Sensor-Schicht. Diese beschreibt die Software der Sensoren, die die Position bestimmen sollen, z.B. die Firmware eines GPS-Empfängers. Aufgabe der darauffolgenden Ebene, der Empfangsschicht, ist es, die relevanten Informationen der Sensordaten auszulesen und an die übergeordnete Abstraktions-Schicht weiterzureichen. Dort werden die Daten ausgewertet und auf ein einheitliches Format gebracht. Die Auswertung erfolgt mit Hilfe einer zusätzlichen Datenquelle, die es ermöglicht, den gefilterten Sensordaten einen bekannten Ort zuzuordnen. Während die ersten beiden Stufen der physischen oder virtuellen Positionsbestimmung entsprechen, wird auf dieser Ebene die logische Positionsbestimmung angewandt. In der Fusions-Schicht werden alle gesammelten Positionsinformationen zu einer gebündelten Positionsangabe vereinigt. Die Aufgabe

dieser Schicht ist es, die Daten auf Schlüssigkeit zu überprüfen und evtl. auftretende Unstimmigkeiten zu beheben. Die oberste Ebene, die Anwendungsebene, ist die Schnittstelle zu den Clients, die die aktuelle Position für weitere Aktivitäten benötigen. Um zu gewährleisten, dass nicht alle Clients alle Informationen erlangen können, gibt es eine Zugriffsverwaltung, die festlegt, wer Zugang zu welchen Daten erhalten darf.

3.3 Kontextmakler

Ein möglicher Client, der auf die in Kapitel 3.2 angesprochene Anwendungsschicht zugreift, ist der Kontextmakler. Er stellt ein Beispiel für ein kontextbewusstes System dar. Wie ein solches System aussehen kann hängt von den Anforderungen und der Umgebungsbeschaffenheit ab, wie z.B. der Ort der Sensoren (lokal oder entfernt), die Anzahl der Benutzer (wenige oder viele), die Ressourcen der Anwendergeräte (High-End-PC oder Handy) oder die Möglichkeit der Erweiterbarkeit des Systems [11].

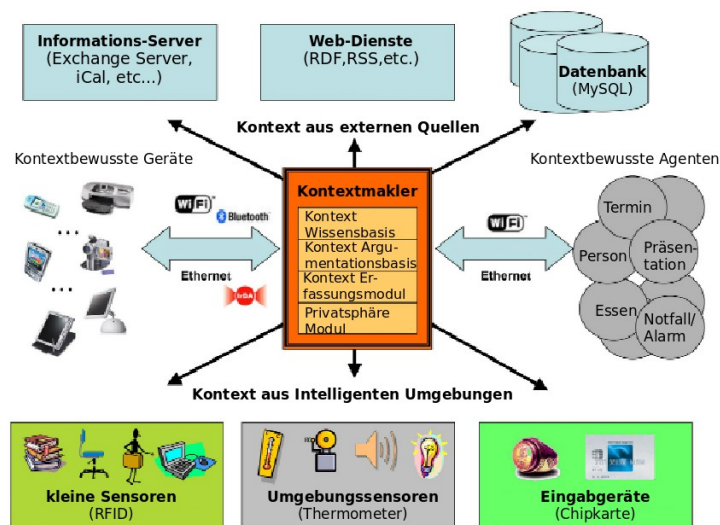


Abbildung 5: Intelligentes Kontextmakler System

Abbildung 5 zeigt ein mögliches Szenario eines kontextbewussten Systems mit Hilfe eines Kontextmaklers [12]. Der Kontextmakler läuft auf einem leistungsfähigen Server, der den gesammelten Kontext pflegt und verwaltet. Die Quellen werden hier in externe und Quellen der intelligenten Umgebung unterschieden. Die externen Quellen bestehen aus Informations-Servern, (z.B. Exchange Server oder ein iCal Kalender), Web Services oder Datenbankverwaltungssysteme. Diese können unter anderem zur virtuellen Positionsbestimmung genutzt werden (Vgl. Kapitel 3.1). Die intelligente Umgebung

versorgt den Broker über verschiedene Sensoren und Eingabegeräte mit Kontext. Dies kann zum Beispiel über kleine Sensoren wie RFID, Umgebungssensoren oder Eingabegeräte erfolgen. Der Makler sammelt und verwaltet all diese Daten, um sie dann über verschiedene Übertragungsarten (z.B. Wireless-Lan, Bluetooth, Netzwerk oder Infrarot) an kontextbewusste Endgeräte oder Agenten zu übergeben. Die Endgeräte, z.B. PDA oder Handy können die gewünschten Daten dem Benutzer über Ausgabegeräte wie Display, Boxen, etc. darbieten. Kontextbewusste Agenten hingegen werten die Daten weiter aus und lösen daraufhin ein bestimmtes Ereignis, wie z.B. Termine oder Alarm aus.

Der Makler selbst ist in vier Ebenen unterteilt:

- **Kontext-Wissensbasis:** Der Kontext, der durch die verschiedenen Quellen gesammelt wurde, wird beständig gespeichert um später darauf zurückgreifen zu können.
- **Kontext-Argumentationsbasis:** In dieser Ebene werden die Kontextdaten ausgewertet und auf Schlüssigkeit überprüft. Wenn z.B. ein Benutzer laut Kalender in einer Konferenz ist, aber ein Umgebungssensor mitteilt, dass er sich in seinem Büro aufhält, muss diese Schicht das Problem lösen.
- **Kontext-Erfassungsmodul:** Hierbei handelt es sich um eine Sammlung von Methoden, um eine Middleware-Ebene für die Kontextermittlung zu schaffen. Dieses Modul sorgt dafür, dass sich verschiedene Regeln nicht gegenseitig blockieren oder aufheben. Wenn der Makler z.B. für eine Präsentation das Licht in einem Raum ausschaltet, muss erkannt werden, dass das Licht nicht, wie üblich, angeschaltet wird sobald jemand weiteres den Raum betritt.
- **Privatsphärenmodul:** Es besteht aus einer Menge aus Folgerungsregeln, die festlegen, wer auf welche Kontextdaten Zugriff erhält. So kann z.B. festgelegt werden, dass ein in der Hierarchie untergeordneter Mitarbeiter nicht die aktuelle Position seines Chefs erfahren darf.

Das Verwenden des Kontextmaklers implementiert zwei entscheidende Dinge, die für Ubiquitous Computing wichtig sind. Es wird das Anbieten der Daten für ressourcenarme Computer und die mögliche Geheimhaltung von Daten anderen gegenüber ermöglicht. Nachteilig ist jedoch, dass bei vielen Anfragen und großen Datenmengen der Broker der Flaschenhals des Systems ist. Dies lässt sich jedoch durch das Verwenden mehrerer Broker im Verbund vermeiden.

3.4 Kontext-Modelle

Ein Kontext-Modell wird benötigt, um Kontext-Daten in maschinenverwertbarer Form zu definieren und zu speichern. Ein solches Modell sollte folgende Ansprüche erfüllen [13]:

- **Verteilter Aufbau:** Grundlage für Ubiquitous Computing sind verteilte Systeme. Der Nachteil davon ist, dass eine zentrale Instanz fehlt, die für die Erstellung, Verwaltung und Erhaltung der Kontext-Daten verantwortlich ist. Deswegen muss ein Kontext-Modell diese Aufgaben übernehmen und somit die Fähigkeit haben, mit sehr dynamischen Netzwerkstrukturen und Datenquellen umgehen zu können.
- **Fülle und Qualität der Informationen:** In der Regel werden etliche Sensoren verwendet, um Daten zu erfassen. Über die Zeit sammelt sich somit eine große Menge an Kontextdaten an. Das Kontext-Modell muss in der Lage sein, die Menge zu verwalten und qualitativ unterscheiden zu können.
- **Unvollständigkeit und Mehrdeutigkeit:** Überlicherweise sind Kontextdaten zu einem beliebigen Zeitpunkt nicht vollständig oder nicht eindeutig bestimmbar. In diesem Fall muss das Modell in der Lage sein, diese unvollständigen Daten in verwertbare Daten umzuwandeln, z.B durch Interpolation.
- **Partielle Validation:** Es sollte in der Lage sein, die unvollständigen Daten auf Gültigkeit zu überprüfen. Wegen der großen Abhängigkeiten innerhalb eines Systems, ist dies wichtig um Fehler zu verhindern.
- **Formsache:** Von großer Bedeutung ist es, dass die Daten präzise und nachvollziehbar beschrieben werden. Um Unklarheiten zu vermeiden, muss vorher exakt definiert werden, was unter einem Begriff genau verstanden wird, z.B. Definition: „in der Nähe“ bedeutet „im Umkreis von 20m“.
- **Anwendbarkeit auf bestehende Umgebungen:** Ein Kontext-Modell muss aus Entwicklersicht auf bereits vorhandene Ubiquitous Computing-Strukturen anpassbar sein, wie z.B. das Framework eines Web-Services.

Strang und Linnhoff-Popien [13] unterteilen die Kontext-Modell-Ansätze in folgende Klassen:

- Schlüsselwortbasierte-Modelle
- Beschreibungsschema-Modelle
- Graphische-Modelle
- Objektorientierte-Modelle
- Logische-Modelle
- Ontologiebasierte-Modelle.

Schlüsselwortbasierte-Modelle

Das wohl einfachste Modell eines Kontext-Modell-Ansatzes sind schlüsselwortbasierte Modelle. Hierbei werden Schlüsselwörter und Ereignisse in einer Tabelle gespeichert. Falls ein Algorithmus passende Schlüsselwörter findet, wird das Ereignis ausgelöst:

Bezeichner	Ort	Ereignis-Typ	Aktivität
Kalender	Büro	ankommen	starte Outlook
PC aus	Haustür	verlassen	PC Herunterfahren

Abbildung 6: Beispiel eines Stichwortbasierten-Modells

Abbildung 7 zeigt ein Beispiel eines solchen Modells. Es enthält zwei Einträge, “Kalender” und “PC aus”. Sobald das Büro betreten wird, also der Ereignis-Typ “ankommen” ausgelöst wird, wird das Programm Outlook gestartet. Ebenso wird beim Verlassen des Gebäudes durch die Haustür der PC heruntergefahren.

3.5.2 Beschreibungsschema Modelle

Bei Beschreibungsschema Modellen werden die Daten in einer hierarchischen Struktur sortiert. Sie besteht aus Marken, die aus Attributen und Inhalt bestehen. Typische Beispiele hierfür sind Profile, die in der Regel auf SGML bzw. XML basieren, wie z.B. Composite Capabilities / Preferred Profile (CC/PP) [14] oder User Agent Profile (UAProf) [15]

Beispiel: CC/PP

Abbildung 6 zeigt ein mögliches Szenario eines Kontextmodells. Es geht darum die Position eines Professors zu speichern. Dies geschieht in drei Kategorien:

- Der physische Ort: Land, Stadt und Ort
- Der logische Ort: IP, Telefonnummer, Webseite
- der geographische Ort: Längen- und Breitengrade.

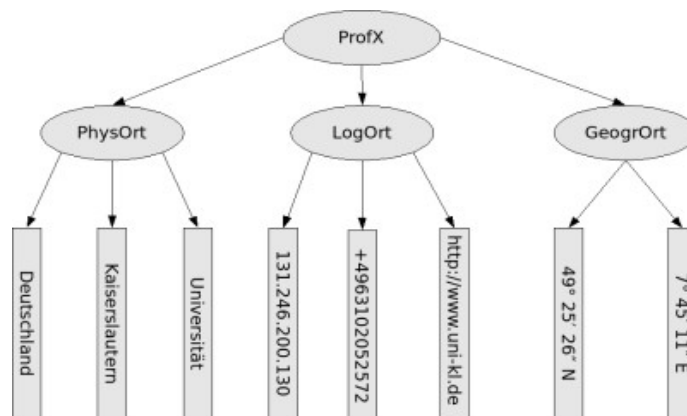


Abbildung 7: Beispiel CC/PP

Eine verkürzte Umsetzung dieses Schema in CC/PP würde folgendermaßen aussehen.

```

<? xml version="1.0" ?>
<rdf:RDF
  xmlns:loc="http://www.x.org/ort-schemat#">
  <rdf:Description about="http://www.x.com/person#ProfX">
    <ccpp:component>
      <rdf:Description about="ProfXPhysOrt">
        <rdf:type rdf:resource="http://www.x.com/ort-
schemat#PhysOrt "/>
        <loc:land>Deutschland</loc:land>
        <loc:stadt>Kaiserslautern</loc:stadt>
        <loc:ort>Universität</loc:ort>
      </rdf:Description>
    </ccpp:component>
    ...
  </rdf:Description>
</rdf:RDF>

```

3.5.3 Graphische Modelle

Aus graphischen Modellen ist es problemlos möglich Entity-Relationship-Diagramme zu erstellen, aus denen man relationale Datenbankschemas herleiten kann. Die bekannteste Methode Kontext graphisch darzustellen ist die Modellierungssprache UML (Unified Modeling Language). Eine weitere Darstellungsmöglichkeit ist Object Role Model (ORM).

3.5.4 Objektorientierte Modelle

Bei objektorientierten Kontext-Modelle sollen es die erweiterten Möglichkeiten, wie z.B. Kapselung, Vererbung oder Wiederverwendbarkeit erleichtern, die Dynamik des Kontextes besser im Modell erfassen zu können.

3.5.5 Logische Modelle

Dieses Modell besteht aus regulären Ausdrücken der Aussagenlogik wie Fakten, Regeln und Ausdrücken. Damit werden Ausdrücke erstellt, mit denen es möglich ist, Daten hinzuzufügen, zu ändern oder zu löschen. Durch Deduktion ist es dann

möglich, weitere Fakten, Regeln oder weitere Ausdrücke abzuleiten. Die Kontext-Daten werden als Fakten gesehen.

3.5.6 Ontologiebasierte Modelle

Dieses Modell ist eine erweiterte Kombination aus objektorientierten und logischen Modellen. Es besteht aus Einheiten, Instanzen, Regeln, Abhängigkeiten und Axiomen. Damit ist es möglich, komplexe Sachverhalte in ein Modell abzubilden.

4. Anwendungsbeispiel: Ein Kontextbewusstes System in einem Krankenhaus

Im Folgenden sollen die Möglichkeiten, Chancen, Risiken und Probleme von Ubiquitous Computing in einem praktischen Anwendungsfall aufgezeigt werden. Als Beispiel hierfür wurde ein Projekt an einem Dänischen Krankenhaus ausgewählt [16]. Das Krankenhaus besitzt ca. 130 Mitarbeiter, von denen 30-50 an einem normalen Arbeitstag im Dienst sind. Ubiquitous Computing soll hauptsächlich im OP verwendet werden und den Belegungsplan vereinfachen. Drei der zehn Operationssäle wurden mit dem neuen System ausgestattet.

4.1 Aufbau

Das in diesem Fall verwendete System besteht im Wesentlichen aus vier Komponenten:

- **Kontextbewusste Infrastruktur:** Zuständig für Beschaffung, Verteilung und Verwaltung der Kontextdaten.
- **Positionsbestimmungssystem:** Ermöglicht die aktuelle Positionsbestimmung von Ärzten, Mitarbeitern und Patienten.
- **AwareMedia Software:** Eine kontextbewusste Software, die die Terminplanung im OP überwacht und auf großen, interaktiven Displays darstellt.
- **AwarePhone:** Eine Software für Handys, die den Datenaustausch der Ärzte und Mitarbeiter untereinander realisiert.

Die Infrastruktur basiert auf dem Java-Context-Awareness-Framework (JCAF) und ist in drei Ebenen untergliedert:

Die erste Ebene stellt die Position der einzelnen Mitarbeiter und den Terminplaner dar. In der zweiten Ebene werden die Kontextdaten gesammelt,

verarbeitet und ausgewertet. Die dritte Ebene versorgt die Endgeräte, also die AwareMedia Software und das AwarePhone, mit relevanten Informationen.

Das Kontext-Modell besteht im Wesentlichen aus den Einheiten Personal, Patient und OP-Saal, wobei Räume und Personen im Modell als gleich angesehen werden. Jede Einheit enthält die Informationen Ort, Status und Aktivität. Ort und Status werden vom System selbstständig erfasst. Die Aktivität muss von dem Mitarbeiter jedoch manuell eingestellt werden, wie z.B. „operieren“.

Das Positionsbestimmungssystem ermöglicht es festzustellen, wo sich ein Mitarbeiter zu dem speziellen Moment aufhält. Hier hat man sich auf Grund der hohen Verfügbarkeit, z.B. von PDAs oder Mobiltelefonen und den niedrigen Kosten für Bluetooth entschieden. Als Empfänger werden handelsübliche, leicht modifizierte Bluetooth-USB-Dongles verwendet, die in der Reichweite reduziert wurden. Als Sender agieren bluetoothfähige Handys. Da nicht genügend Mobiltelefone für alle Mitarbeiter zur Verfügung stehen, werden die übrigen Mitarbeiter mit kleinen Bluetooth Chips, sogenannten Tags, ausgestattet. Insgesamt werden zehn Bluetooth Tags und 15 Handys verwendet.

Die AwareMedia-Software ermöglicht über große interaktive Bildschirme die Koordination der Mitarbeiter. Sie gibt darüber Auskunft, wo sich ein Mitarbeiter derzeit befindet und was er gerade tut. Des weiteren liefert sie Informationen darüber, welche OP-Säle belegt sind und wie weit die darin stattfindende Operationen fortgeschritten sind. So kann z.B. eine Schwester über das System sehen, dass der Patient gerade anästhesiert worden ist und somit die OP noch nicht begonnen hat.

Das AwarePhone ist ein auf Symbian OS basiertes Handy, über das man jederzeit die Position und den aktuellen Status der anderen Mitarbeiter einsehen kann. Die Personen können angerufen werden oder man kann ihnen eine Nachricht hinterlassen. Da Personen und Räume in diesem System gleich behandelt werden, besteht auch die Möglichkeit, Nachrichten und Anrufe an Räume zu senden. So kann z.B. die Schwester sehen, dass der Arzt gerade am Operieren ist und deswegen keine Anrufe entgegennehmen kann. Bei wichtigen Mitteilungen könnte sie eine Nachricht direkt in den OP-Saal schicken, in dem sich der Arzt gerade befindet.

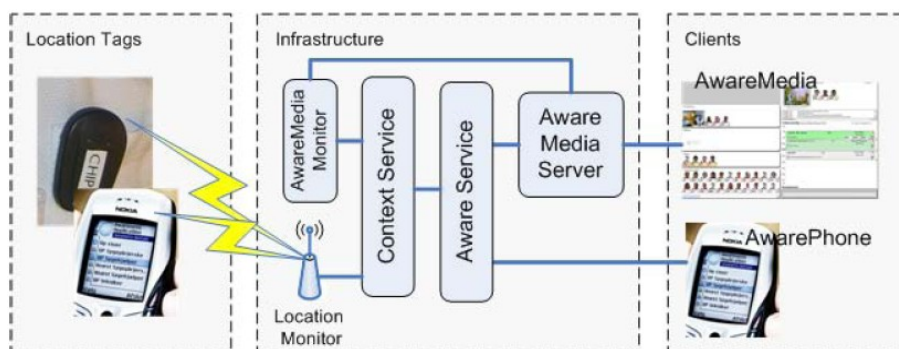


Abbildung 8: Die vier wesentlichen Komponenten des Systems

Abbildung 8 verdeutlicht das Zusammenspiel der einzelnen Komponenten miteinander. Der "Location Monitor" ermittelt über die Handys oder Location Tags, wo sich die Mitarbeiter gerade befinden. Der "Context Service" wertet diese Informationen aus und gibt sie an die AwareMedia Software weiter. Dieser Service entspricht der in Kapitel 3.2 angesprochenen Abstraktionsschicht. Die Aware-Service-Ebene, welche der Anwendungsschicht entspricht – stellt die Informationen dann den Clients und Mobiltelefonen zur Verfügung.

4.2 Auswertung

Durch Befragung der Beteiligten und Auswertung der gesammelten Daten soll vier Monate nach dem das System installiert wurde, herausgefunden werden, wie gut das System von den Mitarbeitern angenommen wurde und in wiefern es die tägliche Arbeit vereinfacht oder erschwert hat.

Die Auswertung der Daten hat ergeben, dass das System sehr gut angenommen wurde. Auch Wochen nach Einführung des Systems, als es nicht mehr neu und aufregend war, wurde es regelmäßig und häufig verwendet. Dies führt die Entwickler zu dem Schluss, dass das System in den alltäglichen Arbeitsablauf integriert worden ist. An einem Arbeitstag wurden durchschnittlich 2500 Positions-, 80 Aktivitäts- und 70 Statusabfragen ausgeführt.

Die Auswertung der Befragungen hat ergeben, dass bereits nach zwei Wochen der alte, papierbasierte OP-Plan komplett durch das neue System ersetzt wurde. Für die leitende OP-Schwester war es nach eigenen Angaben damit einfacher zu arbeiten. Sie konnte bei einem Notfall genau sehen, welcher Arzt zur Verfügung stand und welches Personal problemlos in einen anderen OP-Saal verschoben werden konnte. Auch das übrige OP-Personal steht dem neuen System positiv gegenüber. Durch die Positionsüberwachung konnte jeder sehen, wenn ein Angestellter gerade mit einer wichtigen Tätigkeit beschäftigt war. So konnte z.B. eine Schwester sehen, dass ein Arzt gerade am Operieren war und kontaktierte deshalb einen anderen Arzt, der frei war. Durch solch ein Verfahren nimmt die Anzahl störender Anrufe ab.

Negativ aufgefallen ist, dass nicht jeder Mitarbeiter einen eigenen Positions-Tag besitzt. Deshalb musste vor der Benutzung jeder Bluetooth-Tag auf den Benutzer neu eingestellt werden. Dies war für viele Angestellte zu umständlich. Des weiteren wurde kritisiert, dass die Statusleiste des OPs manuell per Hand bedient werden musste und sich nicht automatisch einstellte.

Ein weiteres Problem war, dass viele Ärzte der neuen Technik nicht so aufgeschlossen gegenüberstanden und sie deshalb nicht so intensiv nutzten wie das restliche Personal.

Ein von den Entwicklern befürchtetes Problem ist jedoch nicht aufgetreten: die Angst der ständigen Überwachung. Durch das Positionssystem besteht die Möglichkeit, einen Mitarbeiter die komplette Arbeitszeit über zu überwachen und zu kontrollieren wie effektiv er arbeitet. In diesem Fallbeispiel ist den Mitarbeitern von Beginn an versichert worden, dass die Überwachung nicht vom Management dazu benutzt wird, um sie zu kontrollieren. Die Positionsermittlung soll ausschliesslich

für Koordinationszwecke verwendet werden. Auf Grund des rechtzeitigen Klarstellens dieser Tatsache hat sich auch keiner der Mitarbeiter unter ständiger Beobachtung gefühlt. Was jedoch erwähnt werden muss ist, dass nicht alle Räume überwacht wurden. Es wurde zu Beginn festgelegt, dass Aufenthaltsräume, Cafeterien und Toiletten von der Bewachung ausgenommen werden.

4.3 Erfahrungen

Der Test des Systems ging über vier Monate hinweg. Auf Grund der hohen Akzeptanz des Personals wurde er als Erfolg gewertet. Das kontextbezogene System wurde wirksam in den Arbeitsalltag integriert. Jedoch lief das System in diesem Beispiel meist im Hintergrund. Die Aufgaben, die das System übernommen hat, lassen sich im Wesentlichen den darstellenden Aktionen zuordnen (Vgl. Kapitel 2.4.2). Helfende Aktionen waren die Ausnahme und auf ausführende Aktionen wurde komplett verzichtet. Dies hängt damit zusammen, dass bei einem Krankenhaus eigenmächtige Entscheidungen des Systems Menschenleben gefährden könnten. Hier sind die Konsequenzen einer ausführenden Handlung wesentlich gefährlicher als bei einem intelligenten Büro. Das System kann auf Grund der vorhandenen Daten nie 100%ig sicher sein, dass sie stimmen und die darauffolgende Reaktion gewünscht ist. In der Regel ist es ein Raten basierend auf Erfahrungs- oder Trainingswerten. Beim Erstellen eines kontextbewussten Systems sollten deshalb immer folgende Punkte berücksichtigt werden [13]:

- Wie exakt sind meine Kontextdaten?
- Wie zuverlässig ist die drauffolgende Handlung?
- Welche Konsequenzen kann diese Handlung haben?

Besonders die Konsequenzen spielen eine entscheidende Rolle. Wenn man als Beispiel Ubiquitous Computing in einem Büro nimmt, stellt es kein großes Problem für den Anwender dar, wenn er beim Betreten des Zimmers seine Emails gezeigt bekommt, obwohl er nur schnell eine Zugverbindung nachschauen wollte. Im Krankenhaus kann eine falsche Aktion jedoch im Extremfall zum Tod eines Menschen führen.

5. Fazit und Ausblick

Die Ausarbeitung zeigt, dass Ubiquitous Computing keine Utopie mehr ist und sich viele Visionen – teilweise mit Einschränkungen - tatsächlich in die Realität umsetzen lassen. Die Kosten für die dazu benötigte Hardware sind überschaubar und das benötigte Know-How ist vorhanden. Stellt sich die Frage, wieso kontextbezogene Systeme in der Praxis so selten vorkommen. In meinen Augen fehlt bei vielen Visionen des Ubiquitous Computing die Trennung zwischen wirklichem Nutzen in

der Praxis und technischer Spielerei. Ist es wirklich wünschenswert, dass beim Betreten eines Raumes automatisch das Licht angeht oder das Email-Programm geöffnet wird? Ist es soviel Aufwand einen Lichtschalter zu bedienen oder einen Doppelklick auf ein Icon zu machen? Viele dieser Ideen sind, meiner Meinung nach, einfach überflüssig und für den nicht technikbegeisterten Benutzer uninteressant. Kapitel 4 zeigt zwar, dass es auch Bereiche gibt in denen kontextbezogene Anwendungen Arbeiten erleichtern können. Aber hier stellt sich dann die Kosten-Nutzen-Frage. Kann man durch Entlastung des Krankenhauspersonals, genug Kosten einsparen um eine solche Investition zu rechtfertigen?

Ich denke, dass kontextbezogene Anwendungen in der Zukunft sicher eine Rolle spielen werden. Meiner Meinung nach werden dies jedoch Anwendungen sein, die sich den Kategorien der darstellenden und helfenden Aktionen zuordnen lassen (Vgl. Kapitel 2.4.2). Ausführende Handlungen können, in meinen Augen, nur in Systemen eingesetzt werden, deren Auswirkungen überschaubar sind. Allerdings denke ich, dass solche Systeme eher eine technische Spielerei sind als eine wirkliche Unterstützung.

Ein kontextbewusstes System kann nur perfekt funktionieren, wenn alle Umstände bekannt sind. Dazu müsste es aber wissen, was gerade im Kopf des Benutzers vorgeht. Selbst wenn das möglich wäre, ist die Frage, ob dies überhaupt wünschenswert ist. Je mehr das System über den Benutzer und seine Umgebung weiß, desto gläserner wird der Benutzer selbst. Alle Daten müssen irgendwo gespeichert werden und somit besteht auch die Möglichkeit, dass private Daten zu anderen Zwecken missbraucht werden.

Ein System, das sich in der Praxis durchsetzen soll, muss einen wirklichen Nutzen bringen, darf also keine technische Spielerei sein und sollte möglichst wenig sensible Daten des Benutzers benötigen.

Deswegen denke ich, dass kontextbewusste Anwendungen nur mit Einschränkungen (darstellende und helfende Aktionen) in der Praxis zu finden sein werden.

Literatur

- [1] Weiser, M.: *The Computer for the 21st Century*, Scientific American, 1991
- [2] <http://www.heute.de/ZDFheute/inhalt/27/0,3672,2275067,00.html> –
Letzter Abruf: 14. 06. 2007
- [3] Lyytine, K., Yoo, Y.: *Issues and Challenges in Ubiquitous Computing*, ACM Press, 2002
- [4] http://ec.europa.eu/information_society/index_en.htm - Letzter Abruf: 14.06.2007
- [5] <http://www.nextgenerationmedia.de/> - Letzter Abruf: 14.06.2007
- [6] Hilty, L., Behrendt, S., Binswanger, M., Bruinink, A., Erdman, L., Fröhlich, J., Köhler, A., Kuster, N., Som, C., Würtenberger, F.: *Auswirkungen des Pervasive Computing auf Gesundheit und Umwelt*, TA SWISS, 2003
- [7] Orajarvi, O.: *Context aware computing*, 2006
- [8] Mayerhofer, R.-M.: *An Architecture for Context Prediction* Schriften der Johannes-Kepler-Universität Linz, Trauner Verlag, April 2005
- [9] Schilit, B., Adams, N., Want, R.: *Context-Aware Computing Applications*, IEEE Workshop on Mobile Computing Systems and Applications, 1994
- [10] Indulska, J., Sutton, P.: *Location Management in Pervasive Systems, Conferences in Research and Practice in Information Technology Series; Vol. 34*, 2003
- [11] Baldauf, M., Dustdar, S. Rosenberg, F.: *A Survey on Context-Aware Systems, International Journal of Ad Hoc and Ubiquitous Computing*, 2004
- [12] Chen, H., Finin, T., Joshi, A.: *An Ontology for Context-Aware Pervasive Computing Environments*, Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review, Mai 2004
- [13] Strang, T., Linnhoff-Popien, C.: *A Context Modeling Survey First International Workshop on Advanced Context Modelling, Reasoning And Management at UbiComp 2004*, 2004
- [14] <http://www.w3.org/Mobile/CCPP/> – Letzter Abruf: 14. 06. 2007
- [15] http://www.openmobilealliance.org/release_program/uap_v2_0.html –
Letzter Abruf: 14. 06. 2007
- [16] Bardram, J., Hansen, T., Mogensen, M., Soegaard, M.: *Experiences from Real-World Deployment of Context-Aware Technologies in a Hospital Environment*, UbiComp 2006, 2006
- [17] http://www.bitkom.org/46624_44673.aspx – Letzter Abruf: 25. 06. 2007