

# Seminarausarbeitung AW1

Sebastian Rudolf

Kontextsensitive Assistierung

# Sebastian Rudolf

## Kontextsensitive Assistierung

Semiarausarbeitung AW1 eingereicht  
im Studiengang Informatik Master  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuer: Prof. Dr. Kai von Luck

Abgegeben am 28. Februar 2011

---

## Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>4</b>
1.1. Motivation . . . . .	4
1.2. Inhalt und Ziel dieser Arbeit . . . . .	5
<b>2. Problemrahmen</b>	<b>5</b>
2.1. Idee . . . . .	5
2.2. Bestehende Arbeiten . . . . .	6
2.3. Basisszenarios - Anforderungen . . . . .	6
<b>3. Lösungsansätze</b>	<b>7</b>
3.1. Context Aware Computing . . . . .	7
3.1.1. Kontextformen . . . . .	7
3.1.2. Context Awareness - Passive und aktive Assistierung . . . . .	8
3.1.3. Modellierung des Kontextes - Situationen und Zustände . . . . .	10
3.2. Umgebungsprofile . . . . .	11
3.3. Anwendung - Funktionsaufteilung . . . . .	12
<b>4. Zusammenfassender Ausblick</b>	<b>13</b>
<b>Literatur</b>	<b>14</b>
<b>Glossar</b>	<b>16</b>

# 1. Einleitung

## 1.1. Motivation

Im stetigen bestreben sein Leben und Alltag zu einfacher, angenehmer oder komfortabler zu gestalten ist der Mensch seit jeher auf der Suche nach Hilfsmitteln. Seit etwa 20 Jahren wird die Problematik von intelligenten, mit Rechenleistung ausgestattet und aufgabenorientiert funktionierenden, Umgebungen durch die Wissenschaft genauer betrachtet, Ubiquitous Computing[Weiser (1991)] und Ambient Intelligence umrahmen diese Problematik in teilen. Oft ergänzend dazu wird das Erfassen von Zustandsräumen und Situationen, eines **Kontextes**, durch Software, in dem Bereich des Context Aware Computing betrachtet. Adaptiver Software bedient sich dieser Teildisziplin wodurch zum Beispiel assistierende Systemen mit verschiedenen Ausprägungen realisierbar sind, Ortsbezogenheit und Nutzerorientierung stehen dabei oft im Vordergrund.

Dank dem Stand der Technik ist gegeben das ein Erfassen der Umgebung einfacher als jeher ist, ob durch ausgeklügelte Sensorik oder stark vernetzte Kommunikationskanäle wie dem Internet und deren Verfügbarkeit ist eine Versorgung mit Sensor- und anderen kontextrelevanten Rohdaten abgedeckbar. Diese Daten bieten eine reichhaltige Informationsbasis um kontextorientierte Anwendungen zu realisieren.

Der steigende Bedarf an solchen Anwendungen hat verschiedenen Ursprung, beispielsweise durch die stetig wachsende Zielgruppen betreuungswürdiger Mitmenschen oder der Wunsch der Überblickbarkeit der stetig wachsenden Datenmassen und zunehmend heterogener Umgebungen Herr zu werden. Dey et al. beschreibt die als das Gefühl des Kontrollverlustes des **Nutzer** an dem System.[Louise Barkhuus (2003)] Diese Gedanken sind keine neuen, jedoch existieren zumeist für solche Assitierungen produktspezifische Insellösungen was den Nutzer letztendlich mit vielen Schnittstellen konfrontiert welche er kennen und verstehen muss.

## 1.2. Inhalt und Ziel dieser Arbeit

Dieses Dokument soll eine thematische Umrahmung für ein Problem und einer möglichen Lösbarkeit darstellen. Methoden des Context Aware Computing und die Vereinbarkeit mit der Konstellation von Hard- und Software-Systemen des Living Place Hamburg sollen dabei auf einer hohen Abstraktionsebene betrachtet werden. Das Augenmerk liegt dabei auf der Verwendung bestehender Techniken sowie der Thematik um die Modellierung des Kontextes. Diese Vorarbeit soll unterstützende Grundlage für folgende Projekte und Ausarbeitungen darstellen und zur Realisierung der Masterthesis beitragen.

## 2. Problemrahmen

### 2.1. Idee

Im Living Place Hamburg soll ein kontextsensitives Assistierungssystem integriert werden, wobei ein Teil der Interaktion mit den bereits integrierten, oder noch im Integrationsprozess befindlichen, Hardware- sowie Software-Systemen des Living Place Hamburg, in Folge [Endsysteme](#) genannt, realisiert werden soll und diese gleichzeitig als primären Datenquellen dienen. Die Funktionalitäten die das Systems bereitstellen soll, sollen dabei zu der Erhöhung des Alltagskomforts beitragen. Bei der Interaktion mit der Umgebung soll der Nutzer stets die Kontrolle über die automatisierte und manuelle, kontextorientierte, Handlungsausführung, in Folge [CA-Handlung](#) genannt, besitzen und diese individualisieren können.

Eine Maßnahme welche zum bilden von Assistierungsmustern beitragen soll ist das Nutzen von [Umgebungsprofilen](#). Diese repräsentieren einen gewünschten Umgebungszustand und -verhalten. Beispielsweise könnte dies für die individuelle Lautstärkenregulierung von Geräten genutzt werden. Eine Ausführung dazu findet sich in Abschnitt [3.2](#).

## 2.2. Bestehende Arbeiten

Context Aware Computing ist keiner neuer Begriff im Kontext des Living Place Hamburg, Projekte wie Wecker 2.0 von Jens Ellenberg oder Context Learning von Wolfram Sokollek befassen sich mit der Thematik eingehend.[[Ellenberg \(2010\)](#), [Sokollek \(2010\)](#)] Eine Form dieser Arbeit ist eine bereits in Entwicklung befindliche, Living Place-umfassende, Kontextmodellierung auf Basis von Ontologien. In wie weit diese 5W1H-orientierte Modellierung den Anforderungen des Assistenzsystems genügt oder genügen kann muss eingehend untersucht werden, dafür ist jedoch eine detaillierte Feststellung der Anforderungen an das Assistierungssystem und deren Funktionalität nötig.

## 2.3. Basisszenarios - Anforderungen

Dieser Abschnitt beinhaltet Szenarien welche anhand weniger Beispiele verschiedenen angedachten Funktionalitäten des Assistierungssystems veranschaulichen soll. Im Verlauf der noch ausstehenden Arbeiten soll sich der Realisierung dieser Funktionalitäten schrittweise angenähert werden. Konkrete Szenarien ergeben sich aus der Einbindung der Endsysteme. Folgende Szenarien sind also denkbar:

### > Präsentation von Information

Der Nutzer soll in definierten (Teil-)Umgebungen die dafür als relevant eingestuft, priorisierten Informationen Zugriff erhalten, diese Umgebungen können ortsbestimmt als auch aktivitätsbestimmt sein:

- o Beim Verlassen der Wohnung wird eine sicherheitsrelevante Übersicht von aktuellen Gerätezuständen wie Fenster, Tür und Elektrogeräten bereitgestellt
- o Erreichen Geräte oder Systeme einen als kritisch definierte Zustand soll der Nutzer darüber benachrichtigt werden

### > Umgebungsinteraktion und Automatisierung

Das System soll eine Interaktionschnittstelle bieten, welche nach dem entsprechenden herrschenden Kontexten individualisiert ist und den Zugriff auf einen Teile der

Endsysteme ermöglicht, zusätzlich dazu dient es die zur automatischen Abarbeitung verfügbaren CA-Handlungen zu individualisieren und Umgebungsinformationen abzurufen:

- o Temperaturregulierung über Heizung sowie Fenster, dabei zum Beispiel betrachten der Außentemperatur, Luftfeuchtigkeit, Heizungs- und Fensterstauts sowie die Wunschtemperatur aus dem Umgebungsprofil
- o Lautstärkenregelung von ausgewählten Geräte
- o Semi-Automatisierung der täglichen Rituale wie Aufstehen, Wohnung verlassen und betreten durch das aktivieren oder deaktivieren von Geräten, gegebenenfalls auch Konfigurieren

Eine Lautstärkensteuerung als Grundorientierung für weitere Arbeiten erscheint als angebracht. Es werden verschiedene Endsysteme zur Interaktion benötigt, kann bei Bedarf aber auch stark reduziert werden und deckt die sowohl die *Interaktion* als auch die *kontextsensitive Datenrepräsentation* und die *CA-Handlungen* ab.

## 3. Lösungsansätze

### 3.1. Context Aware Computing

#### 3.1.1. Kontextformen

Als Context Aware Computing wird vereinfacht Software beschrieben welche sich anhand dynamischer Umgebungen, und deren Zustandswechsel, im Verhalten umkonfigurieren[[Bill N. Schilit \(1994\)](#)] Kontext existiert auf verschiedenen Dimensionen und Ebene. Zum einen kann zwischen [internen und externen Kontext](#) unterschieden werden wobei hier die Art der [Kontext-Information](#) gedeutet wird.[[Gwizdka \(2000\)](#)]

*Interne* Kontext-Information beschreiben dabei die Zustände des Systems und Nutzers.

*Externe* Kontext-Information reflektieren Informationen welche aus der Umgebung bezogen werden können.

Zum anderen kann man in [Kontext-Ebenen](#) unterscheiden.

*Die erste Ebene* stellen Sensordaten dar welche zur weiteren Verarbeitung und Präsentation genutzt werden können. Die Quellen der Sensordaten können dabei *physische als auch virtuelle Sensoren* sein. Im Allgemeinen spricht man auch Low-Level-Kontextinformationen, diese sind zum Beispiel Zeit, Ort, Temperatur, Drehzahl, Ausrichtung und der gleichen, aber auch gebündelte Informationen wie Nutzerprofile oder andersartige Informationsansammlungen können als Low-Level-Kontextinformationen dienen, müssen aber gegebenenfalls normalisiert werden. Low-Level-Kontextinformationen bilden in Beziehung zueinander betrachtet dabei einen *Low-Level-Kontext* welcher einer Entität zugeordnet ist.

*Die zweite Ebene* ist hierbei der aus der ersten Ebene abgeleitete Kontext, man spricht hier auf oft von High-Level-Kontext, dieser hat eine semantische Bedeutung und kann *uncertain context*, also unscharfer Kontext sein.[[Claudio Bettini \(2009\)](#)] Diese Differenzierung wird von A.K. Dey zum Beispiel auch als primäre und sekundäre Kontext-Ebene bezeichnet.[[Anind K. Dey \(1999\)](#)] Sekundärer Kontext, also High-Level-Kontext, kann mehrere Abstraktionsstufen einschließen die nicht unmittelbar aufeinander aufbauen sein müssen.

Zwischen verschiedenen Kontexten können Beziehungen, [Kontext-Relationen](#), beschrieben werden, der Erschließungsprozess wird im Context Reasoning und Interpretation des Low-Level-Kontextes betrachtet, man nennt diese Kategorie der Kontexte auch [Situatio-  
nen](#).[[Claudio Bettini \(2009\)](#), [Loke \(2006\)](#)] Zunächst soll zur Realisierung der Anwendung nur der Low-Level-Kontext betrachtet werden und ein Context Learning sowie Context Reasoning außen vor bleiben. Weiterführende Arbeiten könnten die Verwendbarkeit von Umgebungsprofilen in diesem Zusammenhang untersuchen.

### **3.1.2. Context Awareness - Passive und aktive Assistierung**

Kontext existiert nicht nur in verschiedenen Formen sondern kann auf verschiedene Arten Auswirkungen haben, so können Kontexte verschiedener Formen auf dieselbe Weise das Verhalten der Anwendung beeinflussen:



Further, we define the first kind of context as active context that influences the behaviors of an application, and the second kind of context as passive context that is relevant but not critical to an application.

Active context awareness: an application automatically adapts to discovered context, by changing the applications behavior. Passive context awareness: an application presents the new or updated context to an interested user or makes the context persistent for the user to retrieve later.

[[Guanling Chen \(2000\)](#)]

Differenziert muss man also die *Art der Assistierung* betrachten, welche sich grundsätzlich in passive und aktive Assistierung aufteilt lässt. Die Definition passiver und aktiver Context Awareness wird dabei von Louise Barkhuus und Anind Dey adaptiert.[[Louise Barkhuus \(2003\)](#)] Abgrenzung muss hier zur einer weiteren Anschauungssicht begangen werden, die interaktive und proaktive, welche sich in ihrer Definition leicht differenziert.[[P. J. Brown \(2001\)](#)]

Unabhängig von der Ansicht wird aber immer wieder die Balance zwischen Kontrolle und Automatisierung angebracht. Dieses Problem beschreiben Dey und Barkhuus et al. ebenfalls.[[Louise Barkhuus \(2003\)](#)] Deshalb soll der Nutzer jederzeit die Möglichkeit besitzen die Automatisierung einer Handlung zu deaktivieren und diese gegebenenfalls zu individualisieren. Die soll vor allem dahingehend dienen das Nutzer das Gefühl der Kontrolle über das System nicht verliert, was zu einer erhöhten Akzeptanz führen soll.

Die Anwendung ist aktiv ausgerichtet da zunächst die Realisierung der angedachten Umgebungsprofile im Vordergrund steht. Nichts desto trotz darf der passive Betrieb nicht vernachlässigt werden da dieser für eine Datenpräsentation und eventuelle spätere Arbeiten bezüglich des Context Reasoning die Basis bildet. Eine große Herausforderung dabei dürfte die Bewertung der Priorität bei überschneidenden Kontexten darstellen die jeweils zu einer CA-Handlung führen sollen, welche sich gegebenenfalls nicht Kooperativ zeigen. Dem Nut-

zer die Möglichkeit zu geben im Umgebungsprofil *Gewichtungen* an verknüpfte Kontexte zu verteilen erscheint daher als hilfreich und muss überprüft werden.

### 3.1.3. Modellierung des Kontextes - Situationen und Zustände

Eine Frage die sich jeher stellt ist wie Kontext modelliert werden soll das er durch ein Computersystem verarbeitet werden kann. In den letzten Jahren verstärkt sich eine Ausprägung die nach einer möglichst generischen Modellierung fragt. Claudio Bettini et al. oder auch Thomas Strang et al. zum Beispiel fragen nicht was man modelliert, also nach der gezielte und passenden Modellierung für eine Anwendungsdomäne, sondern *wie* man generisch modelliert beziehungsweise ein möglichst domänenunabhängiges Modell schafft. Dies geht zum einen auf die Beschreibungsmethoden des Kontextes zurück. A.K. Dey zum Beispiel nutzt die [Parameter](#) Location, Identity, Time und Action.[[Anind K. Dey \(1999\)](#)] Es finden sich aber auch oft vermeintlich einfachere wie die Kombination Who(Identität), Where(Ort), When(Zeit) and What(Aktivität), beziehungsweise dienen diese als Basis für weitere Modellierungsformen. Im Living Place Hamburg wird zum Beispiel ein ontologiebasierter Ansatz umgesetzt welcher ein 5W1H-Schema(Who, Where, When, What, Why, How) nutzt.

Aufgrund der Komplexität wird sich sich die Anforderung an die Modellierung dennoch zunächst an der Anwendungsdomäne orientieren, was sich wiederum an den konkreten beteiligten Endsystemen und deren Funktionalität ausrichten wird. Ein Grund dafür ist auch das eine Entscheidung für ein möglichst generisches Framework schwierig ist. Verschiedene Modellierungsansätze wie ontologie-, grafik-, Key-Value- oder logik-basierend wurden in den vergangenen Jahren immer wieder untersucht(ThomasStrang2004 & Bettini), mit verschiedenen Ergebnissen, dies lässt sich vor allem daraus begründen das verschiedene Anforderungen, wie Grad der Formalisierung oder Beziehungs- und Abhängigkeitsverhalten, an den Kontext gestellt wurden. Bettini et al. macht am Ende einer seiner Arbeiten die Aussage das hybride Modelle nötig sind um das gewünschte Ziel realisieren zu können.

Die Frage ob Relationen zwischen Kontexten modelliert werden muss ist ebenfalls noch ei-

ne offen und muss geklärt werden, Umgebungsprofil übernehmen diese Funktionalität teilweise, im bisherigen Gedankengang sind Kontextrelationen jedoch vorerst von geringerer Bedeutung.

### 3.2. Umgebungsprofile

Umgebungsprofile(UP) sollen als Akkumulierungseinheit von Kontexten dienen um häufige kontextsensitive Handlungen automatisiert und konfigurierbar abarbeiten zu können um definierte Zustände herbeizuführen. Umgebungsprofile sind Beschreibungen von Umgebungszuständen, diese grenzen sich von (Umgebungs-)Situationen ab da nur der passive Zustand der Umgebung und keine Aktion betrachtet werden soll. Es handelt sich im eigentlichen um eine Art Zustandsmodellierung welche in ein Endsystem-Verhalten mündet.

So könnte die Lautstärke für Fernsehgerät, Kopfhörer oder Lautsprecher des PC, HiFi-Anlage, SmartPhone und Türklingel je nach Wunsch individuell in ihrem Verhalten konfiguriert werden. Umgebungsprofile wie Arbeit, Abwesend, Erholung, und Schlaf könnten existieren, welche je nach Bedarf aktivierbar sind. Ist es zum Beispiel das UP Arbeit aktiv werden alle Audio-Ausgabegeräte abgestellt, bis auf die Türklingel und das SmartPhone für die Rufbereitschaft. Ist hingegen das Profil Schlaf aktiv werden alle Geräte abgeschaltet und der Wecker auf die gewünschte Lautstärke reguliert. Erholung bringt alle einbezogenen Geräte auf normale Lautstärken.

Ergänzend dazu könnte gar in Betracht gezogen werden das Standard-Umgebungsprofile wie Erholung und Schlaf oder gar die Deaktivierung automatisiert, kontextgesteuert, zum Beispiel durch Uhrzeit und Wochentag, stattfinden kann. Dies führt unweigerlich zu einer anderen Art von Kontext der als Kriterium für anderen Kontext beziehungsweise Umgebungsprofile dient.

Eine große Herausforderung dabei dürfte die Erfassung der Priorität überschneidender Kontexte darstellen wenn verschiedene Umgebungsprofile und damit ausgelöste CA-

Handlungen auf die selben Endgeräte zugreifen und sich somit gegebenenfalls gegenseitig aufheben. Hier ist die Frage ob es dem Nutzer zuzumuten ist diesen Gesamtüberblick zu behalten oder ob dies wieder zu einem gefühlten Kontrollverlust über das System führen kann. Weiterführend kann untersucht werden ob Umgebungsprofile mit Hilfe von context reasoning aus einer oder mehrerer Akkumulierungen eine Konsolidierung und damit High-Level-Kontext erschließen lässt.

### 3.3. Anwendung - Funktionsaufteilung

The overall goal is to develop evolvable context-aware applications. Therefore the design of the general functions of such applications should not be intertwined with the definition and evaluation of context information, which is often subject to change.

[[Claudio Bettini \(2009\)](#)]

Die Ausrichtung der Software soll sich also an den Anforderungen orientieren und nicht an implementierten Kontextmodellierung dahinter, was im übrigen auch auf alle weiteren, unterstützenden (Daten-)Modelle im Hintergrund bezieht. Die ersten Gedanken zur Architektur der Anwendung orientieren sich zunächst an einem gewöhnlichen Server-Client-Modell und dient der Zuordnung der angedachten Kernelemente:

#### > Kern-Komponente

Die Kern-Komponente implementiert das Kontextmodell und die Umgebungsprofile. Dabei könnte die Widget-Idee von A.K. Dey gegebenenfalls hilfreich sein um jeden Sensor in einer eigenen Instanz abzubilden und die Schnittstelle zu jenen zu stellen. [[Anind K. Dey \(1999\)](#)]

#### > Client-Komponente

Da der Nutzer an verschiedenen Stellen die Möglichkeit besitzen soll über ein beliebiges Interface Informationen abzurufen und Steuerhandlungen in Auftrag geben zu können wird es angebracht gesehen eine Komponente zu entwickeln welche in

erster Linie zur Präsentation, Aufgabendelegierung und Umgebungsprofilmanipulation bestimmt ist.

#### > **Middleware**

Als zentrales Element der Kommunikation zwischen den Systemen soll dabei die bereits eingesetzte Middleware [ActiveMQ](#) dienen. Sensordaten können, in Absprache mit den Endsystembetreibern, vorab gefiltert werden so dass nur Low-Level-Kontextinformationen am System ankommen, was den Vorteil erbringt dass eine Filterung des Informationsrauschens nicht stattfinden muss. Um Sensordaten abzufragen bietet sich die Methode des Publisher/Subscriber über Topics an. Producer/Consumer über Message-Queues kann für die Kommunikation der Kern- mit der Client-Komponente genutzt werden.

Im Vordergrund steht die Entwicklung der Kern-Komponente und das Einbinden sowie Nutzen der Middleware. Die Client-Komponente hat Bedeutung für das Evaluieren der Einflüsse von Umgebungsprofilen auf dem Komfort den der Nutzer empfindet.

## **4. Zusammenfassender Ausblick**

Im Verlauf des Dokumentes wurde veranschaulicht dass die Thematik rund um kontextsensitive Software sehr komplex ist und Problem auf verschiedensten Ebenen sowie eines Softwaresystems als auch der Kontextmodellierung existieren. Unter anderem sind dies die Kontextmodellierung und deren Anforderungen, überschneidender Kontexte, Gewichtung von kontextsensitiven Handlungen und Ableiten höherer Abstraktionsstufen aus Sensordaten. Deshalb soll zunächst ein System modelliert werden. Dafür müssen unter anderem passive Umgebungen und kritische Zustände modelliert werden, Aus- und Eingabegeräte festlegen, eine passende Softwarearchitektur gefunden werden, auch mit Hinblick auf die Erweiterbarkeit durch neue Entitäten, Datenquellen und damit möglichen neuen Kontexten. Auch disziplin fremde Probleme könnten auftreten, wie etwa Folgen für das Wohlbefinden wenn täglicher Rituale automatisiert werden, welche einen psychologischen Hintergrund besitzen.

## Literatur

- [Anind K. Dey 1999] ANIND K. DEY, Gregory D. A.: Towards a Better Understanding of Context and Context-Awareness. (1999). <ftp://ftp.cc.gatech.edu/pub/gvu/tr/1999/99-22.pdf>
- [Bill N. Schilit 1994] BILL N. SCHILIT, Roy W. Norman Adams A. Norman Adams: Context-Aware Computing Applications. (1994). <http://nano.xerox.com/want/papers/parctab-wmc-dec94.pdf>
- [Claudio Bettini 2009] CLAUDIO BETTINI, et a.: A survey of context modelling and reasoning techniques, Pervasive and Mobile Computing (2009). (2009), March. <http://dx.doi.org/doi:10.1016/j.pmcj.2009.06.002>. – DOI doi:10.1016/j.pmcj.2009.06.002
- [Ellenberg 2010] ELLENBERG, Jens: Ein Wecker in einem ubicom Haus. (2010). <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw1/Ellenberg/bericht.pdf>
- [Guanling Chen 2000] GUANLING CHEN, David K.: A Survey of Context-Aware Mobile Computing Research. (2000)
- [Gwizdka 2000] GWIZDKA, Jacek: What's in the Context? An Extended Position paper for CHI 2000 Workshop 11. The What, Who, Where, When, Why and How of Context-Awareness. (2000). <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.132.4110&rep=rep1&type=pdf>
- [Loke 2006] LOKE, Seng W.: On representing situations for context-aware pervasive computing: six ways to tell if you are in a meeting. (2006). <http://www.idi.ntnu.no/grupper/su/bibliography/pdf/2006/Loke2006percomw.pdf>
- [Louise Barkhuus 2003] LOUISE BARKHUUS, Anind D.: Is Context-Aware Computing Taking Control Away from the User? Three Levels of Interactivity Examined. (2003). [http://www.itu.dk/~barkhuus/barkhuus\\_ubicomp.pdf](http://www.itu.dk/~barkhuus/barkhuus_ubicomp.pdf)

- 
- [P. J. Brown 2001] P. J. BROWN, G. J. F. J.: *Context-aware retrieval: Exploring a new environment for information retrieval and information filtering. Personal and Ubiquitous Computing*. 2001
- [Sokollek 2010] SOKOLLEK, Wolfram: Context-Awareness im Living Place Hamburg. (2010). <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw1/sokollek/bericht.pdf>
- [Weiser 1991] WEISER, Mark: The Computer for the 21st Century. (1991). <http://sandbox.xerox.com/want/papers/ubi-sciam-sep91.pdf>

## Glossar

### **ActiveMQ**

ActiveMQ ein Message Broker, eine Alternative zu iROS im Rahmen der iFlat-Idee

### **CA-Handlung**

Durch das System automatisch ausgeführte Handlung welche durch eine kontextsensitive Steuerung ausgelöst wird

### **Context Awareness**

Kontextsensitivität welche in verschieden differenziert werden kann, hier in passiv und aktiv, nicht zu verwechseln mit interaktiv und proaktiv

### **Endsysteme**

Hardware- und Softwaresysteme die zu der Annahme von CA-Handlungen dienen und als Datenquelle dienen, kann auch das Assistierungssystem sein

### **Interner und externer Kontext**

Beschreibt die Art des Kontextes, nicht zu verwechseln mit local und remote context welcher oft in Literatur im Bereich des mobile computing zu finden ist

### **Kontext**

Ein definierter Zusammenhang aus Parametern, Parametern und Kontext(en) oder mehreren Kontexten, hat Abstraktionsstufen

### **Kontext-Ebene**

Abstraktionsstufe eines oder mehrere konsolidierter Kontexte, Unterscheidung in Low-Level- und High-Level-Kontext, auch als primärer und sekundärer Kontext beschrieben

### **Kontext-Information**

Sensordaten die zum bilden eines Kontextes beitragen können und in Parametern gefasst werden können

### **Kontext-Relation**

Abbildung von Beziehungen zwischen Kontexten, zumeist auf einer Abstraktionsebene



**Nutzer**

Beschreibt in erster Linie den Bewohner der Umgebung in welcher Assistenten werden soll, repräsentiert wird dieser durch ein Gerät dessen Position ortbar ist

**Parameter**

Dient zur Bildung von Low-Level-Kontext, Repräsentant einer Kontext-Information

**Situationen**

Kann eine passive oder aktive Handlungen in der Umgebung beziehungsweise den Umgebungszustand beschreiben

**Umgebungsprofil**

Beschränkt sich im Gegensatz zu Situationen nur auf passive Zustände der Umgebung