



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung Masterprojekt 1

Jens Ellenberg

Vorarbeiten für den Wecker 2.0

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Vision des Wecker2.0	1
2	Teilprojekte	2
2.1	Beschreibung des Kontextes	2
2.1.1	Begriffsdefinition „Kontext“ im iFlat	2
2.1.2	Darstellungsformen für Kontext	3
2.1.3	Darstellung von Kontext im iFlat	6
2.2	Erzeugen einer einheitlichen Begriffswelt	7
2.2.1	Warum ist eine einheitliche Begriffswelt notwendig?	7
2.2.2	Welche Möglichkeiten gibt es?	8
2.2.3	Ontologie für das iFlat	9
3	Zusammenfassung und Ausblick	10
3.1	Zusammenfassung	10
3.2	Ausblick	10
	Literaturverzeichnis	11

1 Einleitung

1.1 Motivation

Nach der Vision von Marc Weiser [11] sind im 21. Jahrhundert Computer überall und sie unterstützen die Menschen bei ihren Aktivitäten. Ein grundlegendes Konzept bei dieser Vision, ist die Fähigkeit der Computer, auf ihre Umgebung zu reagieren. Dafür sind verschiedene Sensoren nötig, die die relevanten Informationen erfassen. Die erfassten Informationen müssen dann interpretiert werden, damit sie anschließend von den verschiedenen Programmen genutzt werden können. Dieser Teilbereich der Programmierung nennt sich in der aktuellen Forschung „Context Aware Computing“. Um die Grundlagen für so ein Programm zu schaffen, wurde der „Wecker 2.0“ ausgewählt. Der Wecker steht hier stellvertretend für alle Programme aus dem „iFlat“ die „Context Aware“ sind. In diesem Projekt sollen die Voraussetzungen geschaffen werden, die notwendig sind, um ein Programm zu entwickeln, das sich dem Kontext bewusst ist. Dabei wird sich am „Wecker 2.0“ und den Voraussetzungen orientiert, die der Wecker benötigt.

1.2 Vision des Wecker2.0

Der Wecker 2.0 ist ein Programm, welches sich seiner Umgebung bewusst sein soll. Mit diesem Bewusstsein, soll dieser Wecker nicht nur statisch an einem festgelegten Zeitpunkt den Bewohner einer Wohnung wecken, sondern er soll die Weck- und Informationszeitpunkte dynamisch den Anforderungen und der Umgebung anpassen. Mit Wissen über die Termine und der Situation des Bewohners, soll der Weckzeitpunkt automatisch generiert werden. Dieser Zeitpunkt soll außerdem von dem Programm automatisch geändert werden, wenn sich die Situation in der Wohnung ändert. Beispielsweise soll auf ein verfrühtes Aufwachen des Bewohners reagiert werden oder wenn ein Stau zu erwarten ist, sollte der Wecker diesen in seine Berechnungen mit einfließen lassen. Der Wecker soll außerdem den Bewohner informieren, wenn er noch nicht gewaschen ist oder noch nicht gefrühstückt hat, ab wann er damit anfangen sollte, bevor er die Wohnung verlassen muss. Weitere Details werden in [6] beschrieben.

2 Teilprojekte

Für den „Wecker 2.0“ müssen verschiedene Voraussetzungen geschaffen werden, damit er Wirklichkeit werden kann. Wie in [6] beschrieben braucht der Wecker:

- Wissen über die Zeitpunkte
- Wissen über den Bewohner

In diesem Projekt liegt der Fokus auf dem Wissen über den Bewohner. Hierbei war zu klären,

1. Wie wird der erfasste Kontext dargestellt?
2. Wie wird eine einheitliche Begriffswelt sichergestellt?

Diese Fragen werden im Folgenden beantwortet.

2.1 Beschreibung des Kontextes

Wie in der Vision des „Weckers 2.0“ (Kapitel: 1.2) beschrieben, nutzt der Wecker den aktuellen Kontext. Um mit dem Kontext zu arbeiten muss der Begriff klar definiert werden. Diese Definition ist nicht allein auf den Wecker beschränkt, da sich der Wecker in der Umgebung des Livingplace Hamburg integriert. Die Kontext Erzeugung und auch andere Programme innerhalb des „iFlat“ [10] müssen mit der gleichen Auffassung von Kontext funktionieren, damit beispielsweise die Kontext-Erzeugung und der „Wecker 2.0“ mit dem gleichen Verständnis von „Kontext“ arbeiten.

2.1.1 Begriffsdefinition „Kontext“ im iFlat

Nach [7] gibt es drei wichtige Theorien, was Kontext bedeutet. Die „Situationsbedingte Aktion“ geht davon aus, dass die ausgeführten Aktionen durch die vorherrschende Aktion bestimmt wird. In der „Aktivitätstheorie“ definieren die Aktionen selbst den Kontext. Es werden auch die Wünsche und Ziele der beteiligten Personen betrachtet. In „soziale Netzwerke“ wird der Kontext durch das Soziale Netzwerk definiert, in dem sich eine Person gerade befindet.

Im iFlat steht der Bewohner im Mittelpunkt, der sich grundsätzlich in der Wohnung befindet. Daher ist nicht das „Soziale Netzwerk“ der kontextbestimmende Faktor. Vielmehr kommt es darauf an, was der Bewohner gerade macht und nicht in welcher Situation sich der Bewohner gerade befindet. Im „iFlat“ definiert daher die Aktion des Bewohners den Kontext. Aus diesem Grund liegt für das „iFlat“ und dem „Wecker 2.0“ die „Aktivitätstheorie“ zugrunde um den Kontext zu beschreiben.

2.1.2 Darstellungsformen für Kontext

Der Wecker muss in der Lage sein, seine Umgebung zu verstehen oder die Informationen über die Umgebung von einer Zwischenschicht abzurufen. In beiden Fällen ist eine interne Darstellung der aktuellen Situation des Bewohners (des Kontextes) notwendig. Hierbei gibt es verschiedene Möglichkeiten zur Darstellung. Diese werden im Folgenden erläutert.

Im Living Place Hamburg soll das Wohnen durch Computer verbessert werden. Das bedeutet, dass die Funktionen der Programme den Bewohner als Fokus haben. Daher muss auch der Bewohner des „iFlat“ im Vordergrund stehen, wenn der Kontext beschrieben werden soll. Das bedeutet, dass das Kontextmodell Nutzerorientiert sein muss und der Kontext des Nutzers durch die Aktion bestimmt wird. Das 5W1H-Modell [8] zur Beschreibung von Kontext genügt diesen Anforderungen.

Das 5W1H-Modell:

Who: Beschreibt den Bewohner durch Namen, ID und sein Profil.

Where: Der Ort, an dem sich der Bewohner befindet. Dieses kann durch x,y-Koordinaten geschehen oder durch abstrakte Beschreibungen, wie zum Beispiel „Küche“ oder „an der Küchentheke“.

When: Dieses ist entweder der Zeitpunkt (z.B. 01.09.2010, 12:31 Uhr) oder ein Zeitraum (z.B. 01.09.2010 mittags).

What: Das Gerät oder Objekt, das der Bewohner bei seiner Aktion braucht, beziehungsweise das Objekt, das im Fokus seiner Aufmerksamkeit liegt.

How: Beschreibt die Aktion, die der Bewohner gerade ausführt oder wie er ein Objekt gerade benutzt.

Why: Der Grund, aus dem die Handlung passiert. Dieses ist ein interner Kontext (vergleiche [7]), der schwer zu bestimmen ist und höchstens geschätzt werden kann. Daher ist dieser Zweig bei den ersten Schritten optional.

System	Space	Time	Space/Time coordinates (Relative or Absolute)	Context history	Subject (User or Application)	User profile (Role or Features based)	Variable context granularity	Valid context constraints	Type of formalism: Key-value-based	Type of formalism: Mark-up based	Type of formalism: Logic-based	Type of formalism: Graph-based	Type of formalism: Ontology-based	Formality level (High or Low)	Flexibility	Context construction (Distributed or Centralized)	Context reasoning	Context quality monitoring	Ambiguity/Incompleteness mgmt.	Automatic learning features	Multi-context model
ACTIVITY	+		A	+	U	F	+							L	+	C	+				+
CASS	+	+		+	U	U								L		D	+				
CoBrA	+	+	A		A	F					+			H	+	D	+		+		
CoDaMoS	+	+	R/A		A	F								H	+	D	+	+			
COMANTO	+	+	R/A		A	F								H	+	D	+				
Context-ADDICT	+	+	R/A		A	R	+	+	+					H	+	C	+				
Conceptual-CM	+	+	R	+	A	R								H	+	C	+		+	+	+
CSCP					A	F			+					L	+	C	+			+	+
EXPDOC		+	R	+	U	F				+				H		C	+			+	+
FAWIS				+	U	F	+		+					H		C	+		+	+	+
Graphical-CM	+	+	R		A	F								H	+	C	+	+	+	+	+
HIPS/HyperAudio	+	+	A	+	U	F		+						H		C	+		+	+	+
MAIS	+	+	A		A	F	+							H		C	+				
SCOPEs					A	A				+				H		D	+		+		+
SOCAM	+	+	R/A		A	F								H	+	D	+	+			+
U-Learn	+	+	A		U	F	+							H	+	D	+				+

Abbildung 2.1: Übersicht von Eigenschaften von Kontextmodellen

Damit wurde bisher festgelegt, wie der Begriff Kontext zu verstehen ist und was mit dem Kontext ausgedrückt werden soll. Als nächstes muss geklärt werden, wie die tatsächliche Repräsentation von Kontext erfolgen soll. In [5] werden verschiedene Systeme gegenübergestellt, mit denen Kontext modelliert wird. Diese Systeme werden in Abbildung 2.1 verglichen.

Die Kriterien in der Tabelle 2.1 teilen sich in die drei Bereiche auf:

Dimension :

- Raum (hat das Modell eine räumliche Komponente?)
- Zeit(hat das Modell eine zeitliche Komponente?)
- absolute Zeit/Raum (sind die Repräsentationen absolut oder relativ?)
- Kontextvergangenheit (ist die Vergangenheit Teil des aktuellen Kontext?)
- Subjekt (wer oder was ist Ziel der Beschreibung)
- Nutzer-Profil (besondere Merkmale des Subjekts)

Repräsentationsmerkmale :

- Art der Formalisierung (Key-Value-, Mark-up-Schema-, Logik-, Graph- oder Ontologiebasiert?).
- Stufe der Formalisierung (existiert eine formale Definition?)
- Flexibilität (kann das Modell mit verschiedenen Domänen oder Anwendungen umgehen?)
- Veränderbare Tiefe der Auflösung (gibt es verschiedene Detailtiefen?)
- Gültigkeitsmerkmale (kann man Regeln für die Gültigkeit des Kontextes einfügen?)

Verwaltung und Nutzung :

- Kontext-Erstellung (wird der Kontext zentral oder verteilt erstellt?)
- Schlussfolgerung auf Kontext (sind Schlussfolgerungen möglich?)
- Qualitätsüberwachung (um fehlerhafte Sensoren aufzudecken)
- Mehrdeutigkeit und Fehler-Verwaltung (gibt es Möglichkeiten, Fehler zu beheben (Fehler-Transparenz))
- Automatisches Lernen von Kontext
- Mehrfach Kontext (können mehrfache Kontexte verwaltet werden?)

In den Dimensionen sind die Ausprägungen Raum und Zeit wichtig für das „iFlat“, und dass das Modell mit verschiedenen Auflösungen umgehen kann, da aus den verschiedenen Sensoren die Genauigkeit variieren kann. Außerdem benötigen verschiedene Systeme unterschiedliche Genauigkeiten von Kontext. Da viele verschiedene Entwickler an dem Projekt arbeiten ist eine hohe Formalisierung notwendig um Missverständnisse zu vermeiden. Die Verwaltung und Nutzung des Kontextes spielt keine vorrangige Rolle, da der Fokus auf den Anwendungen liegt.

In [12] wird der ontologiebasierte Ansatz hervorgehoben. Die folgenden Punkte aus der Arbeit treffen auch auf dieses Projekt zu ¹:

- Bei der Nutzung der Begriffe wird klar, um welche Begriffe es sich handelt.
- Restriktionen in den Begriffen sind in der Ontologie direkt gespeichert und müssen nicht extra beschrieben werden.

¹vergleiche [12] Seite: 9

- Kontextdaten können vielfältig oder komplex sein, daher muss eine angemessene Strukturierbarkeit gegeben sein.
- Das Modell sollte in der Lage sein, die Relationen zwischen den Elementen der realen Welt möglichst präzise nachbilden zu können. Diese kann dann als Basis für ein gemeinsames Verständnis dienen.
- Kontextuelles Wissen sollte in Bezug auf Struktur und Instanzdaten gegen das Modell validiert werden können, um so zur Konsistenz der Daten beizutragen.
- Die maschinelle Verarbeitbarkeit der Kontextdaten ist notwendig um das Modell in den Programmen zu verwenden.
- Durch Weiterentwicklungen im Bereich der Sensortechnik oder durch veränderte Zielstellungen in einem kontextsensitiven System, müssen möglicherweise neue Daten erfasst werden oder die Ziele haben sich geändert. Daher muss das Modell entsprechend erweiterbar sein.
- Die zur Kontexterfassung verwendeten Quellen (z. B. physikalische Sensoren) haben meist unterschiedliche Genauigkeiten. Das Modell sollte daher das Zuordnen von Qualitätsinformationen zu den erfassten Daten unterstützen.
- Da Kontextdaten selten statisch sind, sollte die Zeitabhängigkeit mit in das Modell aufgenommen werden können.

Aus diesen Gründen wurde eine Darstellung des Kontextes nach dem Prinzip des Projekts Context ADDICT [4] gewählt. Der Kontext wird in einer Baumstruktur dargestellt. Mit dieser Darstellung sind verschiedene Auflösungen möglich. Jede Dimension, die gewünscht ist, kann implementiert werden. Außerdem wird dieser Baum als Ontologie abgelegt. Nur durch die Darstellung als Ontologie können die oben genannten Punkte sichergestellt werden.

2.1.3 Darstellung von Kontext im iFlat

Die oben genannten Argumente zusammengefasst, bedeutet das für das Living Place Hamburg, dass der Kontext in den sechs Dimensionen des 5W1H-Schemas (Zeit, Ort, Person, Gegenstand, Aktion, Intention) beschrieben werden soll. Diese Dimensionen werden in einer Baumstruktur als Ontologie dargestellt. Die Ontologie ist zentral im Netzwerk des iFlat verfügbar. Die Abbildung 2.2 zeigt die Klassen der Ontologie.

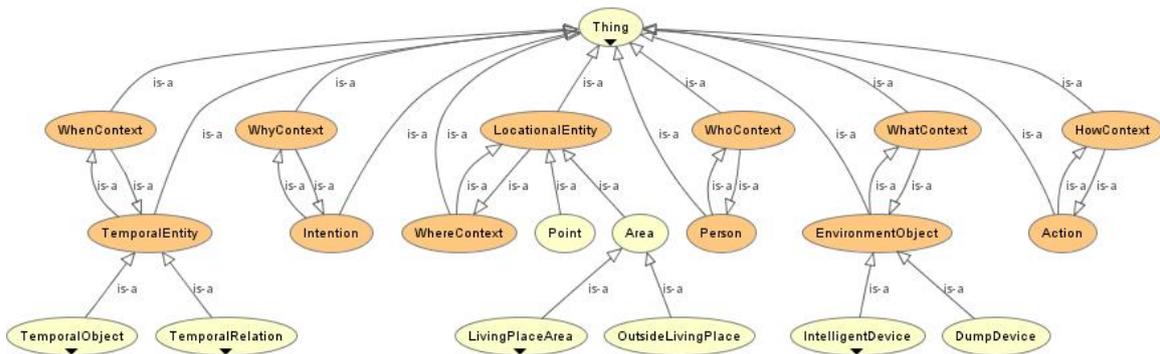


Abbildung 2.2: Kontext im Living Place Hamburg

2.2 Erzeugen einer einheitlichen Begriffswelt

Der „Wecker 2.0“ ist ein Teil eines heterogenen Systems. Es werden von verschiedenen Entwicklern Teile des Systems fertiggestellt. Diese Systemteile müssen sich dann über Nachrichten verständigen. In dem Projekt zur Entwicklung einer Architektur für das Livingplace Hamburg [9] wurde eine Plattform für die Nachrichten entwickelt und ein erstes Format für die Nachrichten definiert. Das Projekt macht aber keine Angaben über die Inhalte der Nachrichten. Dieses wird im Folgenden festgelegt.

2.2.1 Warum ist eine einheitliche Begriffswelt notwendig?

Das Konzept im Living Place Hamburg sieht ein System von Programmen vor, die über eine Blackboard-Architektur miteinander in Verbindung stehen. Dieses System entspricht einer Multi-Agenten-Architektur, da verschiedene unabhängige Programme eine verteilte Aufgabe erfüllen. Von der FIPA² wurde in der „FIPA ACL Message Structure Specification“ [2] eine Kommunikationsstruktur standardisiert. Auch in dieser Struktur ist eine einheitliche Begriffswelt empfohlen. Die wichtigsten Vorteile für eine einheitliche Begriffswelt sind:

- Vermeidung von Missverständnissen in den Verwendungen von Ausdrücken
- Sicherstellen, dass der Begriff bekannt ist, der übermittelt wird
- Strukturierung der Informationen

²Foundation for Intelligent Physical Agents

Eine einheitliche Begriffswelt hat auch einen gewichtigen Nachteil. Der Aufwand ist größer, wenn man Formalien einhalten muss und alle genutzten Begriffe abgestimmt werden müssen. Des Weiteren ist der Entwickler in seiner Freiheit eingeschränkt, da er sich den globalen Vereinbarungen beugen muss. Daher muss für jedes Projekt abgewogen werden, wieviel Aufwand sinnvoll ist, um Fehler in der Kommunikation zu vermeiden.

2.2.2 Welche Möglichkeiten gibt es?

Die Möglichkeiten für eine Festlegung auf bestimmte Begriffe sind weitreichend. Generell können drei Systeme unterschieden werden:

- mündliche Absprachen (zwischen den Entwicklern)
- informale schriftliche Absprachen (nicht durch Programme nutzbar)
- formale Festlegungen (durch Programme nutzbar)

Jedes dieser Systeme hat seine Vor- und Nachteile. Mündliche Absprachen sind die einfachste Form. Hierbei ist kein größerer Aufwand erforderlich und die Programme brauchen keine weitere Modifikation. Bei einem Wechsel der Entwickler oder Änderungen in den Absprachen könnte bei diesem System die Anpassung der Programme sehr aufwendig werden, da nichts festgehalten wurde. Diese könnte in einer Neuentwicklung aller Programme enden. Da bei dem Projekt des Living Place Hamburg von ständig wechselnden Entwicklern auszugehen ist, hat diese Lösung mehr Nachteile als Vorteile.

Informale, schriftliche Absprachen benötigen ein System auf denen diese Absprachen hinterlegt werden. Hierfür ist eine Plattform notwendig, die von jedem Entwickler erreichbar ist und auch gepflegt und beachtet wird. Bei diesem System existiert zumindest ein gewisser Grad von Persistenz. Der Erfolg dieser Art der Absprache hängt von der Disziplin der Entwickler ab, dieses System zu nutzen. Es ist ein zusätzlicher Zeitaufwand notwendig, die Absprachen in das System einzutragen und vor jeder Änderung müssen alle beteiligten Systeme überprüft werden, um Fehler zu vermeiden. Genauso muss darauf vertraut werden, dass sich jeder Entwickler an die Absprachen hält. Eine weitere Gefahr besteht in den informalen Beschreibungen. Sollen diese zu locker gefasst werden, können wieder Missverständnisse entstehen.

Formale Festlegungen, die durch Programme lesbar sind, bieten einen hohen Schutz gegenüber Missverständnissen. Man kann auch innerhalb des Programms überprüfen, ob Begriffe bekannt sind und durch den Formalismus auch Inkonsistenzen in den Absprachen finden. Ontologien aus dem Bereich des Semantik Web sind für solche formalen Festlegungen verbreitete Möglichkeiten. Ontologien werden auch von der FIPA für die Kommunikation von Agenten empfohlen [1]. Bei diesem System ist der zusätzliche Aufwand am höchsten. Bei

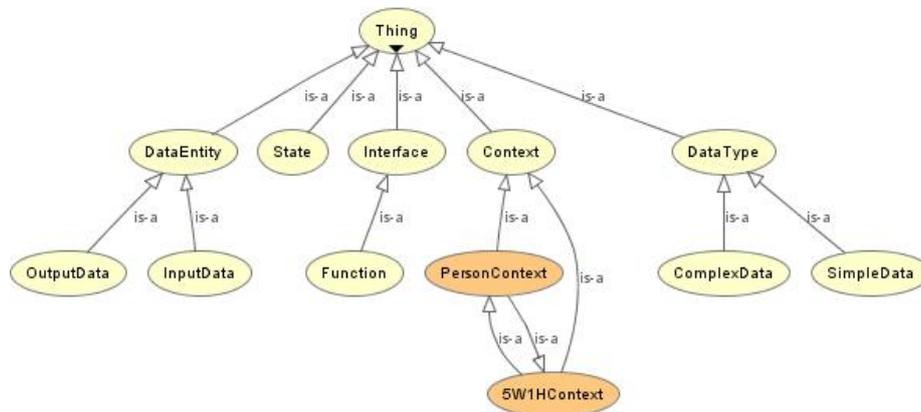


Abbildung 2.3: Ontologie des Living Place Hamburg

der Kommunikation müssen die Formalien berücksichtigt werden und die Ontologie muss zentral verfügbar sein und ständig gepflegt werden. Außerdem muss sich jeder Entwickler mit den Mechanismen von Ontologien beschäftigen und diese in seinem Programm berücksichtigen.

2.2.3 Ontologie für das iFlat

Innerhalb des Living Place Hamburg arbeiten ständig über 20 Entwickler an verschiedenen Projekten. Manche Projekte hängen miteinander zusammen andere wiederum nicht. Daher müssen Absprachen auf jeden Fall festgehalten werden, da auch in einem Projekt die Entwickler jederzeit wechseln können. Für das Projekt besteht ein Wiki³. Dieses wird von den Studenten aber nur sehr unregelmäßig gepflegt, so dass man nicht immer davon ausgehen kann, dass die Daten aktuell sind oder dass die eigenen Einträge beachtet werden. Daher sollte für die Kommunikation der Programm im Living Place Hamburg der Aufwand betrieben werden, eine Ontologie zu nutzen. Um den Aufwand so gering wie möglich zu halten, wird die Ontologie auf das Living Place Hamburg beschränkt. In der Ontologie werden nur jene Begriffe verwaltet, die auch zur Kommunikation zwischen den Programmen gebraucht werden. In Abbildung 2.3 sind die Klassen der initialen Ontologie abgebildet. Diese Ontologie soll als Ausgangspunkt für die weitere Entwicklung dienen. In dieser Ontologie sind auch schon die Klassen enthalten, die für die Darstellung des Kontextes benötigt werden.

³offenes Content-Management-System

3 Zusammenfassung und Ausblick

3.1 Zusammenfassung

In diesem Projekt wurden allgemeine Grundlagen für das iFlat und den „Wecker 2.0“ des Living Place Hamburg entwickelt. Der Begriff „Kontext“ wurde durch die „Aktivitätstheorie“ genauer definiert. Weiter wurde eine Baumstruktur dargestellt, mit der Kontext in seinen verschiedenen Dimensionen beschrieben werden kann. Der Vorteil dieser Struktur ist neben den verschiedenen Hierarchieebenen auch die Möglichkeit der unterschiedlichen Granularität der Kontextinformationen. Für die Beschreibungssprache des Kontextbaums wurde eine Ontologie vorgeschlagen. Eine Ontologie eignet sich hier besonders, da unter anderem von Software verarbeitet werden kann und das System durch die Ontologie validierbar wird. Neben der Ontologie, mit der der Kontext dargestellt werden soll, wird auch eine Ontologie für die generelle Kommunikation zwischen den Programmen des iFlat vorgeschlagen. Da es sich bei der Architektur des iFlat im wesentlichen um eine Multi-Agenten-Struktur handelt, ist eine einheitliche Kommunikation notwendig. Hierbei wurde auch auf die Richtlinien der FIPA Bezug genommen.

3.2 Ausblick

Nachdem diese Voraussetzungen zur Kommunikation geschaffen wurden, kann der Kontext für den Wecker definiert werden. Dieser Kontext muss nun aus den Sensoren des iFlat erkannt werden. Dieses könnte von einer Middleware geschehen, wie es in [9] vorgeschlagen wurde. Als weitere Schritte kann nun der „Wecker 2.0“ entwickelt werden, der mit der Middleware interagiert und selbst Informationen und Daten an andere Programme des iFlat sendet. Beispielsweise sind Schnittstellen zur „intelligenten Küchentheke“ [3] geplant.

Literaturverzeichnis

- [1] FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS (Hrsg.): *FIPA Ontology Service Specification*. FIPA. <http://www.fipa.org/specs/fipa00086/XC00086C.html>. Version:08 2001
- [2] *FIPA ACL Message Structure Specification*. FIPA. <http://www.fipa.org/specs/fipa00061/SC00061G.html>. Version:12 2002
- [3] BARNKOW, L. : Ausarbeitung zur Veranstaltung Anwendungen 1im Masterstudiengang Eine Multitouch-fähige Küchentheke: Im Kontext des Living Place Hamburg / HAW Hamburg. Version:2010. <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw1/barnkow/bericht.pdf>. – Forschungsbericht
- [4] BOLCHINI, C. ; CURINO, C. ; SCHREIBER, F. A. ; TANCA, L. : Context Integration for Mobile Data Tailoring. In: *MDM '06: Proceedings of the 7th International Conference on Mobile Data Management*. Washington, DC, USA : IEEE Computer Society, 2006. – ISBN 0-7695-2526-1, S. 5
- [5] BOLCHINI, C. ; CURINO, C. A. ; QUINTARELLI, E. ; SCHREIBER, F. A. ; TANCA, L. : A Data-oriented Survey of Context Models. In: *ACM SIGMOD Record* 36 (2007), Dec, Nr. 4, S. 19-26. <http://dx.doi.org/10.1145/1361348.1361353>. – DOI 10.1145/1361348.1361353
- [6] ELLENBERG, J. : Ein Wecker in einem ubicom Haus / HAW Hamburg. Version:2010. <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw1/Ellenberg/bericht.pdf>. HAW Hamburg. – Forschungsbericht
- [7] GREENBERG, S. : Context as a dynamic construct. In: *Hum.-Comput. Interact.* 16 (2001), Nr. 2, S. 257-268. http://dx.doi.org/10.1207/S15327051HCI16234_09. – DOI 10.1207/S15327051HCI16234_09. – ISSN 0737-0024
- [8] JANG, S. ; WOO, W. : 5W1H: Unified user-centric context. In: *Proceedings of the 7th International Conference on Ubiquitous Computing* Citeseer, 2005

- [9] OTTO, K. ; VOSKUHL, S. : Entwicklung einer Architektur für das Livingplace Hamburg / HAW Hamburg. Version: 2010. <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2010-proj/otto-voskuhl.pdf>. HAW Hamburg. – Forschungsbericht
- [10] STEGELMEIER, S. : iFlat - Eine dienstorientierte Architektur für intelligente Räume / HAW Hamburg. Version: 2009. <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/papers/aal2009.pdf>. VDE2. – Forschungsbericht
- [11] WEISER, M. : The computer for the 21st century. In: *SIGMOBILE Mob. Comput. Commun. Rev.* 3 (1999), Nr. 3, S. 3–11. <http://dx.doi.org/10.1145/329124.329126>. – DOI 10.1145/329124.329126. – ISSN 1559–1662
- [12] WINKLER, R. : Entwicklung eines ontologiebasierten Kontextmodells für kollaborative Web-Anwendungen / Technische Universität Dresden. 2007. – Forschungsbericht

Bildquellen

Die Quellen der Bilder sind:

- Abbildung 2.1 aus dem Paper [5]
- Abbildung 2.2 mit dem Tool Protege Version 4.1 alpha erstellt.
- Abbildung 2.3 mit dem Tool Protege Version 4.1 alpha erstellt.