



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Anwendung II

Sven Vollmer
Dienstsuche im IntelliHome

Sven Vollmer
Dienstsuche im IntelliHome

Inhaltsverzeichnis

1	Einleitung	4
1.1	Vision	4
1.2	Motivation	5
2	Anforderungen	6
2.1	Netzwerkschicht	6
2.2	Transportschicht	6
2.3	Architekturschicht	7
2.4	Semantikebene	8
2.5	Ontologie	9
3	Dienstbeschreibung	10
3.0.1	Beispiele: Dienstbeschreibung in XML	10
4	Semantic Web	12
4.1	Ontologien	13
4.2	Web Ontology Language	13
4.2.1	Semantic Markup for Web Services	14
5	FIPA	15
5.1	Agent Communication Language	15
5.1.1	Beispiel: RDF-basiertes FIPA-ACL	16
6	Zusammenfassung	18
	Literaturverzeichnis	19

1 Einleitung

Die Dienstsuche in einem intelligenten Haus ist Thema dieser Ausarbeitung. In der Ausarbeitung zur Ringvorlesung siehe [Vollmer (2008a)] wurde eine mögliche Masterarbeit erörtert. Dieser Bericht beleuchtet einleitend die Anforderungen an eine Dienstsuche im Kapitel 2. Die Probleme insbesondere die Einigung auf eine einheitliche Dienstbeschreibung wird in dem folgenden Kapitel 3 behandelt. Anschließend wird im Kapitel 4 eine mögliche Lösungsstrategie für die Dienstbeschreibung mit Hilfe des Semantic Webs gegeben. Drauffolgend wird im Kapitel 5 FIPA für Agentensysteme vorgestellt und deren mögliche Nutzung für die Dienstbeschreibung. Abschliessend gibt der Autor im Kapitel 6 eine Zusammenfassung wieder.

1.1 Vision

Heutzutage befinden sich in einer Wohnung sehr viele multimedia Geräte, die eine Vielzahl von Funktionen verfügen. Durch diese Vielseitigkeit sind entsprechende Steuerungseinheiten nötig, die sogenannten Fernbedienungen. Da die Steuerung der einzelnen Geräte sehr unterschiedlich sind, werden durch die vielen Geräte eine Menge Fernbedienungen notwendig. Dieser Umstand wird in Zukunft weiter zunehmen, da immer mehr Elektronik im Haushalt einzug nehmen wird. Dies kann zu entnervten Benutzer führen, da man nicht nur die Geräte mit einzelnen Fernbedienungen steuern, sondern auch die jeweilige Steuerung erlernen muss. Eine mögliche Lösung wäre die Nutzung einer zentralen Fernbedienung, die alle Geräte in der Wohnung steuern kann. Ein entsprechender Ansatz kommt von der Firma Logitech siehe [Logitech (2007)]. Solch ein Ansatz geht in die richtige Richtung, erfordern aber je nach Geräten umfangreiche Konfigurationen. In dieser Ausarbeitung möchte der Autor Ansätze aufzeigen, um Dienste wie Audio- und Videoausgabe in einer intelligentern Wohnung zu suchen und zu nutzen.

1.2 Motivation

Der Autor hat im Projekt „Pervasive Gaming“ Erfahrungen bei der Entwicklung mobiler Anwendungen erlangt. Diese möchte er nutzen, um eine zentrale Fernbedienung für ein mobiles Endgerät zu entwickeln. Mit dessen Hilfe kann man Dienste in einer intelligenten Wohnung suchen, nutzen und verwalten. Zu einem späteren Zeitpunkt soll es möglich sein Geräte, die Dienste anbieten, einfach in das bestehende System der Wohnung einbinden zu können. Man kauft sich zum Beispiel einen neuen Fernseher aus dem Elektronikmarkt seines Vertrauens. Dieser wird dann in der Wohnung aufgestellt und kann im besten Fall ohne Administrationsaufwand in das Haussystem einzubinden. Eine Möglichkeit hierfür wäre das Bekanntmachen des Fernsehdienstes durch Knopfdruck am Fernseher. Danach ist der Fernseher in der intelligenten Wohnung als Dienst nutzbar. Für dieses Szenario sind einige Anforderungen notwendig, welche in der folgenden Ausarbeitung erörtert werden.

2 Anforderungen

Das automatische Einbinden in das Haussystem und Bekanntmachen des Dienstes erfordert gewisse Anforderungen an das System. Diese werden im Verlauf dieses Kapitels in eine Art Schichtenmodell, die aufeinander aufbauen, erläutert.

2.1 Netzwerkschicht

In der Netzwerkschicht wird die Art der Vernetzung der verschiedenen Geräte untereinander festgelegt. Es gibt die Möglichkeit ohne Server zu arbeiten, also auf der P2P-Ebene oder klassisch auf der Client-Server-Ebene. Der Unterschied liegt in der Verwaltung der Dienste. Bei P2P verwaltet jeder Client seinen Dienst oder je nach Granualität auch ein Teil der Geräte in der Umgebung. Bei der klassischen Client-Server Architektur werden die Dienste beim Server registriert und können auch dort abgefragt werden.

Auf das Szenario abgebildet wird der Fernseher beim Aufstellen in der Wohnung automatisch in der TCP/IP Netz eingebunden.



Abbildung 2.1: Schichtenmodell der Anforderungen

2.2 Transportschicht

Auf der Transportschicht bietet sich zum Beispiel UPnP an, da es auf standardisierten Netzwerkprotokollen und Datenformaten basiert und im IP-basierten Netzwerken arbeitet. UPnP dient zur herstellerübergreifende Ansteuerung von Geräten wie Stereoanlage, Router Drucker usw. Außerdem bietet es Mechanismen zum Suchen und Finden von Diensten an, benötigt

keine zentrale Verwaltung und der Dienst wird mit XML beschrieben. Auf das Szenario abgebildet bedeutet es, dass sich die vernetzten Geräte untereinander Nachrichten (IP-Pakete) zusenden können. Also die Kommunikation der Dienste untereinander wird hierdurch geregelt.

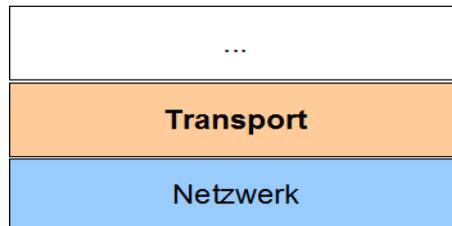


Abbildung 2.2: Schichtenmodell der Anforderungen

2.3 Architekturschicht

In der Architekturschicht geht es auf einer höheren Ebene um die Verwaltung der Dienste, die entweder zentral oder dezentral sein kann. Hier geht es um Fragestellungen wie die Dienste angeboten, verwaltet und abgefragt werden können.

Da der Autor die zentrale Lösung bevorzugt, müssen noch zwei Arten von zentraler Verwaltung erwähnt werden. Zum einen aktiv zentral und zum anderen passiv zentral. UDDI(Webservices) und JINI sind Vertreter für die passive zentrale Verwaltung. Hier werden die Dienste in den jeweiligen zentralen Instanz registriert und können vom Dienstanutzer abgefragt werden. Danach treten Dienstanbieter und Dienstanutzer direkt in Verbindung. Ein Beispiel für aktiv zentral ist der ORB. Hierbei sind Server und Client eng miteinander verkoppelt. Die Suche und die Kommunikation findet über den ORB statt.

Bei dem Szenario verwaltet der Fernseher als eine Möglichkeiten seinen Dienst selber und reagiert auf Anfragen anderer Dienste/Nutzer. Die andere Möglichkeit ist die zentrale Verwaltung. Hier meldet sich der Fernseher mit seinem Dienst an und dieser ist dort dann abfragbar.

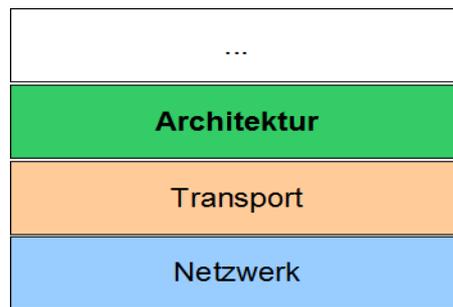


Abbildung 2.3: Schichtenmodell der Anforderungen

2.4 Semantikebene

Auf der Semantikebene werden die Themen „Semantc Web“ und die Agent Communiation Language (ACL) von FIPA interessant. Welche Möglichkeiten bietet ACL für die Dienstbeschreibung des Fernsehers. Dies wird im Kapitel 5 genauer beschrieben. Warum semantische Informationen für Dienste wichtig sein können, wird später im Kapitel 4 genauer erläutert.

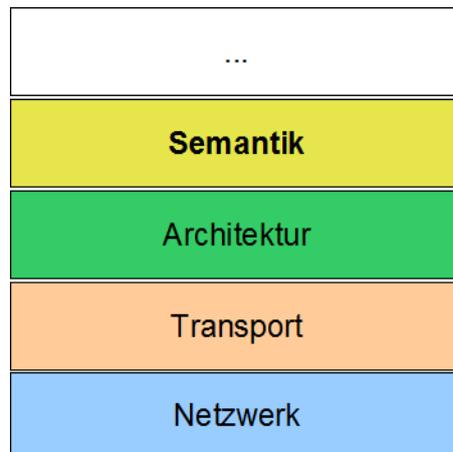


Abbildung 2.4: Schichtenmodell der Anforderungen

2.5 Ontologie

In der obersten Schicht spielt die Ontologie eine bedeutende Rolle. Der Autor wird dessen Rolle im Abschnitt [4.1](#) genauer erklären.

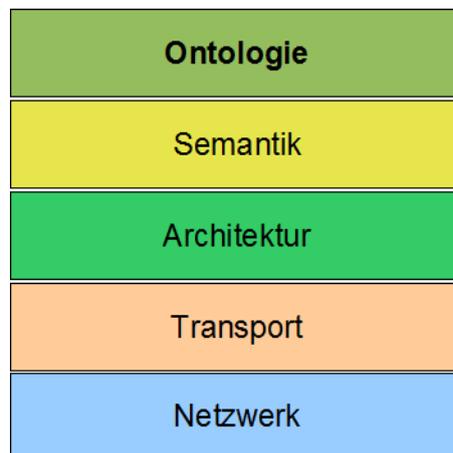


Abbildung 2.5: Schichtenmodell der Anforderungen

3 Dienstbeschreibung

Einen Dienst zu beschreiben führt zu Problemen. Jeder beschreibt seinen Dienst unterschiedlich oder gibt den jeweiligen Attributen und Bezeichnern eine andere Bedeutung. In der Industrie ist für die Dienstbeschreibung noch kein einheitlicher Standard entstanden, da hier die verschiedensten Industriezweige

- Unterhaltung
- Elektronik
- Computer

zusammenarbeiten müssten. Dieses gestaltet sich als schwierig und zeitaufwendig.

Das Problem könnte dadurch gelöst werden, dass man auf den einheitlichen Standard wartet. Der Versuch könnte aber fehlschlagen und könnte zu keinem einheitlichen Standard führen. Kurzfristig und Langfristig könnten sich einige Hersteller einigen und es würde hier zu proprietären Standards kommen. Hier kann man nur durch Adapter entgegenwirken, die zwischen den proprietären Standards übersetzen. Eine weitere Möglichkeit besteht in der individuellen Dienstbeschreibung der jeweiligen Hersteller, aber bei gleichzeitiger Lieferung einer Semantikbeschreibung. Die Hersteller müssten ihren Dienst zum Beispiel anhand von Tags in XML beschreiben. Zusätzlich müsste eine Ontologie angegeben werden, auf welcher diese Beschreibung basiert. Hierzu später im Abschnitt [4.1](#) mehr.

3.0.1 Beispiele: Dienstbeschreibung in XML

```
<?xml version= '1.0' standalone='yes' encoding= 'UTF-8' ?>
<Fernseher>
  <hersteller>Samsung</hersteller>
  <name>LW-20 M 21 C 51cm </name>
  <farbe>schwarz</farbe>
  <jahr>2006</jahr>
  <typ>lcd</typ>
  < helligkeit >450</helligkeit>
</Fernseher>
```

```
<?xml version= '1.0' standalone= 'yes' encoding= 'UTF-8'?>
<Fernseher>
  <manufacturer>Samsung</manufacturer>
  <model>
    <name>LW-20 M 21 C 51cm</name>
    <color>schwarz</color>
    <year>2006</year>
  </model>
  <display>
    <type>LCD</type>
    <brightness>450</brightness>
  </display>
</Fernseher>
```

Bei den beiden Beispielen wird das Problem der Semantik besonders deutlich. Sie beschreiben beide denselben Dienst, aber in einer unterschiedlichen Syntax. Die Bedeutung ist hier dieselbe, aber die Syntax ist unterschiedlich.

4 Semantic Web

Der Erfinder des „World Wide Web“ Tim Berners Lee definiert das Semantic Web wie folgt:

“The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.”

- Tim Berners Lee -

Er sieht das Semantic Web als eine Erweiterung des herkömmlichen Webs an, in der Informationen mit eindeutigen Bedeutungen versehen werden, um die Arbeit zwischen Mensch und Maschine zu erleichtern. Diese Informationen werden also prinzipiell mit Metadaten versehen. Die Metadaten werden hierbei in RDF und XML als Auszeichnungssprache sowie zum Datenaustausch versehen.

Das Resource Description Framework (RDF) ist ein W3C-Standard, um beliebige Ressourcen im Web zu beschreiben. Ressourcen im Web werden Eigenschaften und Werte in einer Art RDF-Tripel zugewiesen.

Beispiel: Subjekt - Prädikat - Objekt

http://www.haw-hamburg.de ist *Homepage* der *HAW Hamburg*

http://www.haw-hamburg.de ist hierbei das Subjekt, *Homepage* das Prädikat und *HAW Hamburg* das Objekt.

Um das Ganze in maschinenlesbarer Form zu bekommen, wird RDF/XML verwendet. Hier das obige Beispiel in RDF/XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:haw="http://www.haw-hamburg.de/haw#">
<rdf:Description rdf:about="http://www.haw-hamburg.de">
<haw:homepage>HAW Hamburg</haw:homepage>
<haw:title>HAW Hamburg: HAW HAMBURG</haw:title>
</rdf:Description>
</rdf:RDF>
```

Das Semantic Web kann für die Dienstbeschreibung sehr nützlich sein. Mit Hilfe von von RDF/XML könnte man die Ressource, also den Dienst, mit deren Eigenschaften beschreiben.

4.1 Ontologien

Mit Hilfe der Ontologien wird versucht ein gemeinsames Vokabular von Fachleuten zu definieren. Sie beschreiben formal die Begriffe, Konzepte und Beziehungen von Wissen in einem bestimmten Fachgebiet. Mit Hilfe einer Ontologie kann eine Kommunikation zwischen Fachleuten stattfinden, obwohl sie nicht die gleichen Fachbegriffe benutzen. In dem folgenden Beispiel wird deutlich, warum eine Ontologie wichtig ist:

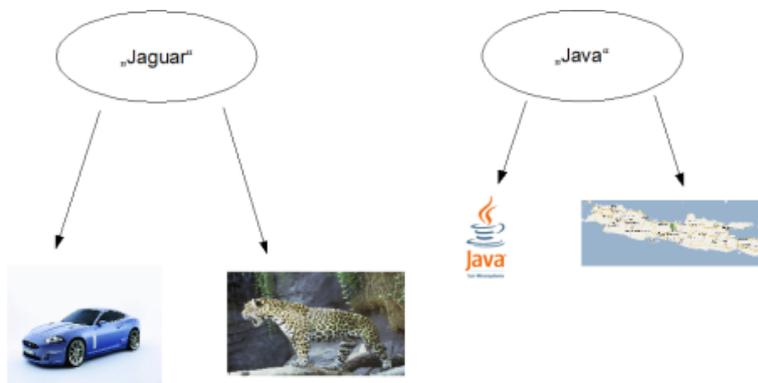


Abbildung 4.1: Beispiele für Ontologien

In dem Beispiel wird deutlich, dass man unter dem Begriff „Jaguar“ das Auto oder das Tier meinen kann. Im zweiten Fall kann mit dem Begriff „Java“ die Insel und die Programmiersprache gemeint sein. Bei Festlegung einer Ontologie tritt das Problem nicht mehr auf, da dort die Bedeutung des Begriffs festgelegt ist.

4.2 Web Ontology Language

Die Web Ontology Language (OWL) ist eine Ontologiesprache für das semantische Web und wurde vom W3C spezifiziert. OWL nutzt und erweitert RDF, um Ontologien zu definieren. Insbesondere unterscheidet OWL Klassen, Eigenschaften und Instanzen.

- **Klasse:** Person
- **Eigenschaften:** männlich oder weiblich
- **Instanz:** eine bestimmte Person

Auf die weiteren Feinheiten und unterschiedlichen Ausdrucksstärken geht der Autor nicht weiter ein, da es den Rahmen dieser Ausarbeitung sprengen würde.

4.2.1 Semantic Markup for Web Services

Die Semantic Markup for Webservices OWL-S ist eine Ontologie auf Basis von OWL. Das Ziel von OWL-S lag in der semantischen Beschreibung von Webdiensten und im weiteren auch die Dienste dort zu automatisieren. Dazu zählen *automatische Web Service Discovery*(Auffinden), *automatische Web Service Invocation*(Ausführen), *automatische Web Service Composition and Interoperation*(Zusammensetzen und Verbinden). Die Abb. 4.2 zeigt den Aufbau solch eines Dienstes in OWL-S. Im ServiceProfile wird beschrieben was der Dienst macht. Dort werden Informationen wie ServiceName, Beschreibung, Input, Output und Precondition hinterlegt. Im ServiceGrounding wird festgelegt wie man auf den Service zugreift und welches Nachrichtenformat z.B. WSDL hierbei benutzt wird. Im ServiceModel wird der Kommunikationsablauf mit dem Dienst beschrieben.

OWL-S wäre eine weitere gute Möglichkeit zur Dienstbeschreibung. Es wäre mit Hilfe dieser Technologie möglich den Fernseher(Dienst) semantisch zu beschreiben. Der Einblick in das Themengebiet ist relativ oberflächlich gehalten. Bei einer späteren Verwendung in der Masterarbeit, sollte hier weiter in die Tiefe gegangen werden.

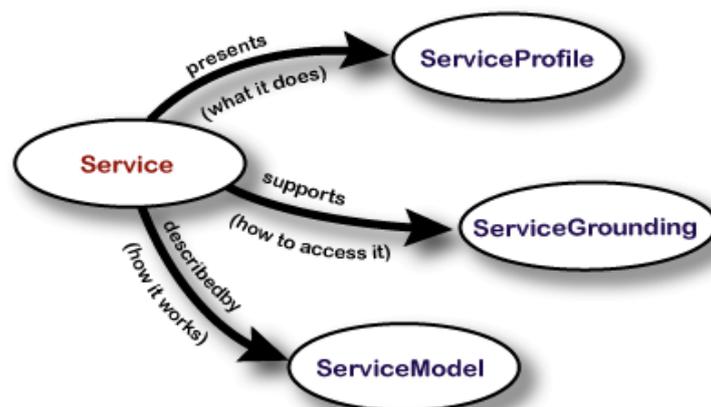


Abbildung 4.2: Dienstaufbau bei OWL-S

5 FIPA

FIPA ist eine gemeinnützige Organisation und wurde 1996 in der Schweiz gegründet. Die Organisation hat sich im Bereich der Agentensysteme um Standards bemüht. Das Ziel war es Standards für Agentenplattformen zu schaffen. Insbesondere in dem Bereich eine Anbieter-übergreifenden Kommunikation zu ermöglichen. Dadurch sollte eine weitere Verbreitung der Agententechnologie erreicht werden.

Der Autor möchte hier nicht die komplette Architektur von FIPA erklären, sondern sich auf die Agentenkommunikation konzentrieren. Da es dort Konzepte gibt, die für eine spätere Masterarbeit von Interesse sein könnten.

5.1 Agent Communication Language

Die Agent Communication Language (ACL) basiert auf der „Speech Act Theorie“. Die Theorie besagt grob, dass man mit einer sprachlichen Äußerung nicht nur Sachverhalte beschreibt, sondern darüberhinaus Handlung nachvollzieht. Das Thema ist etwas weitgreifender, sollte bei Interesse in der entsprechenden Literatur (John Langshaw Austin: How to Do Things with Words. Cambridge (Mass.) 1962) nachgelesen werden. Der Autor selber hat sich damit nicht weiter beschäftigt, da es den Rahmen in der Ausarbeitung sprengen würde. Mit ACL wurde eine gemeinsame Sprache für die Kommunikation und den Informationsaustausch zwischen Agenten geschaffen. In den folgenden Abschnitten wird der Aufbau ACL beschrieben und am Schluß wird ein Beispiel solch einer auf RDF-basierten FIPA-ACL Nachricht gegeben.

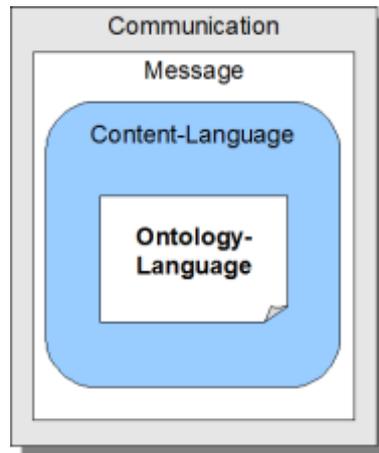


Abbildung 5.1: Aufbau einer FIPA-ACL

- **Communication**

Die höchste Ebene ist der Communication-Layer. Diese enthält Angaben über Sender, Empfänger und den Kommunikationsmodus. Außerdem wird eine eindeutige Identitätsnummer für die Nachricht festgelegt.

- **Message**

In der Ebenen darunter befindet sich der Messenger-Layer. Dort ist die Nachricht selbst aufgeführt.

- **Content-Language**

In der Content-Language Ebene wird die Sprache der Nachricht festgelegt.

- **Ontology-Language**

In der untersten Ebene befindet sich die Ontology-Language. Hier wird die Ontologie, auf der die Nachricht basiert, festgelegt.

5.1.1 Beispiel: RDF-basiertes FIPA-ACL

```
(request
:sender Agent_A
:receiver Agent_B
:content (
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdfsyntax-
ns#" xmlns:fipa="http://www.fipa.org/schemas#">
<fipa:Action rdf:ID="Action1">
<fipa:actor>Agent_A</rdf:actor>
```

```
<fipa:act>open</rdf:act>  
<fipa:argument>door1</fipa:argument>  
</fipa:Action>  
</rdf:RDF>  
:language fipa-rdf0)
```

In dem Beispiel sind Sender: Agent_A und Empfänger: Agent_B festgelegt. Die Content-Language legt die Nachrichtensprache fest, die hier FIPA-rdf0 ist. In der Nachricht fragt der Agent_A mit dem Nachrichtentyp „request“ den Agent_B, ob er die Tür „door1“ öffnet.

Die wichtigste Erkenntnis für das angestrebte Szenario ist die Verwendung von verschiedenen Nachrichtentypen bei der Kommunikation der Agenten. In dem Konfliktszenario konnte die Türklingel(Dienst) den Fernseher anfragen. Dies würde über den „request“-Typ ablaufen. Der Fernseher(Dienst) könnte den Dienst aber auch verweigern. Das würde im FIPA-ACL der Nachrichtentyp „deny“ sein. Dieses Konzept könnte man sich für die Masterarbeit durchaus übernehmen.

6 Zusammenfassung

In der Ausarbeitung wurde die Architektur für eine Dienstsuche vorgestellt, welche Technologien dort zum Einsatz kommen können. Ein wichtiger Bestandteil des Berichts ist die Dienstbeschreibung, die mit verschiedenen Technologien umsetzbar ist. In der Ausarbeitung wurden die Problem rund um die Dienstbeschreibung vorgestellt und Lösungsmöglichkeiten hierfür gegeben. Insbesondere der Standard RDF/XML zeigte sich für die Dienstbeschreibung als würdiger Kandidat. Die Probleme der unterschiedlichen Bedeutungen (Semantik) bei den Dienstbeschreibungen können durch Einsatz von Ontologien gelöst werden. Eine wichtige Erkenntnis zeigte sich in der Untersuchung der FIPA-ACL bei den Agentensystemen. Bei dem Ansatz können Agenten Nachrichten annehmen oder verweigern. Dieser Umstand könnte für bei der Dienstsuche genutzt werden, da auch hier Dienste Nachrichten annehmen oder verweigern können. Die interessanten Ansätze und Konzepte, die in der Ausarbeitung durchleuchtet wurden, könnten für die zukünftige Masterarbeit genutzt werden.

Literaturverzeichnis

- [Amigo 2008] : *Ambiente Intelligence for the networked home - Integriertes Projekt des 6. EU-Rahmenprogramms*. 2008. – URL <http://www.amigo-project.org>
- [OSGi 2008] : *Open Services Gateway Initiative*. 2008. – URL [GatewayInitiative](#)
- [ubicomp 2008] : *Projekt im Masterstudiengang*. Ausarbeitung und Projekte im Rahmen des Masterstudiengangs, Fakultät Technik und Informatik. 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projects.html>
- [Dreyer 2007] DREYER, Markus: *Your Home in your Hand*. Ausarbeitung in Anwendung II im Rahmen des Masterstudiengangs, Fakultät Technik und Informatik. 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08-aw/dreyer/bericht.pdf>
- [Falk 2004] FALK, Heiko: *Dienstfindung mit höherwertigen Diensten*. Diplomarbeit an der Universität Freiburg. 2004. – URL <http://www.eioa.de/material/diplomarbeit.pdf>
- [Feldbusch u. a. 2003] FELDBUSCH, Fridtjof ; PAAR, Alexander ; ODENDAHL, Manuel ; IVANOV, Ivan: *The BTRC Bluetooth remote control system*. ACM. 2003. – URL <http://portal.acm.org/citation.cfm?id=950495.950499>
- [Hentschel 2004] HENTSCHEL, Prof. Dr. C.: *Ambient Intelligence Vision und technische Lösungsansätze*. BTU Cottbus Lehrstuhl Medientechnik. 2004. – URL <http://www-rnks.informatik.tu-cottbus.de/content/unrestricted/teachings/2004/SS/ringVL/Ringvorlesung%20Hentschel%20200404.pdf>
- [Hollatz 2007] HOLLATZ, Dennis: *Managing Information - Personal Information Environments based on iROS*. Ausarbeitung in Anwendung I im Rahmen des Masterstudiengangs, Fakultät Technik und Informatik. 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08/hollatz/bericht.pdf>

- [Klenner u. a.] KLENNER, Wolfgang ; JENISCH, Markus ; RUPP, Stephan: *Managing Information - Personal Information Environments based on iROS*. – URL www.srupp.de/intelligent_inhouse_ambient.pdf
- [Logitech 2007] LOGITECH: *Logitech Harmony 1000 Advanced Universal Remote*. 2007. – URL <http://www.logitech.com>
- [Olmos 2007] OLMOS, Juan Vicente L.: *SERVICE AND RESOURCE DISCOVERY IN SMART SPACES*. Universitat Politècnica de València. 2007. – URL www.mediateam.oulu.fi/publications/pdf/1062.pdf
- [Roth 2005] ROTH, Jörg: *Mobile Computing: Grundlagen, Technik, Konzepte*. dpunkt.verlag, 2005. – ISBN 3-89864-366-2
- [Vollmer 2007] VOLLMER, Sven: *Location-based Services*. Ausarbeitung in Anwendung I im Rahmen des Masterstudiengangs, Fakultät Technik und Informatik. 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2007/vollmer/bericht.pdf>
- [Vollmer 2008a] VOLLMER, Sven: *Dietsuche im IntelliHome*. Ausarbeitung in der Ringvorlesung im Rahmen des Masterstudiengangs, Fakultät Technik und Informatik. 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08/vollmer/bericht.pdf>
- [Vollmer 2008b] VOLLMER, Sven: *Dietsuche im IntelliHome*. Ausarbeitung in Anwendung II im Rahmen des Masterstudiengangs, Fakultät Technik und Informatik. 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08-aw/vollmer/bericht.pdf>