Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung - MS - WS2010/11

Benedikt Johannsen

Generische Modelbasierte Kommunikationsinfrastruktur -thesis outline-

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Inhaltsverzeichnis

1 Einführung **3**
1.1 Rückblick 3

2 Zielsetzung **4**
2.1 Problemstellung 4
2.2 Ziele der Arbeit 5
2.3 Abgrenzung 6

3 Lösungsansatz **7**
3.1 Überblick 7
3.2 Eingesetzte Technologien 8
3.2.1 CIM - Common Information Modell 8
3.2.2 Java API 10
3.2.3 Anbindung ActiveMQ 10
3.3 Architekturübersicht 10
3.4 Geplantes Vorgehen 11

4 Herausforderungen **12**
4.1 Modellierung 12
4.2 Konsistenz 12
4.3 Skalierbarkeit 12

5 Zusammenfassung **13**

Literaturverzeichnis **14**

1 Einführung

Im Rahmen des Aufbaus des „Living Place Hamburg“ an der HAW wird eine intelligente Wohnung entwickelt und aufgebaut. Hier werden eine große Anzahl von Sensoren, Aktoren und Kontrollern eingesetzt. Damit diese miteinander interagieren können wird eine Kommunikationsinfrastruktur benötigt. Da das Living Place keine statische Einrichtung ist sondern laufend erweitert oder modifiziert ist, muss diese Infrastruktur flexibel und modular ausgelegt sein um den dortigen Anforderungen gewachsen zu sein. (vgl. von Luck u. a. [2010])

Abbildung 1.1: Logo des Living Place Hamburg

1.1 Rückblick

Im vorherigen Semestern wurden die Grundlagen für dieses Themengebiet erarbeitet. So wurde unter anderem eine Anforderungsanalyse an Bussysteme in Ambient Intelligence Anwendungen beschrieben, sowie vergleichbare Projekte aus dem Ambient Intelligence Umfeld analysiert.(vgl. Johannsen [2009] & Johannsen [2010])
Parallel hierzu wurden unterschiedliche Bussysteme näher auf ihre Eignung untersucht und exemplarisch ein proprietärer Bus in die bisherige Kommunikationsinfrastruktur des Living Place integriert. (Pautz [2009] & Pautz [2010])
Somit konnten eine Vielzahl von Anforderungen ermittelt werden die diese Umgebung an eine Kommunikationsinfrastruktur stellt. Auf diesen Erkenntnissen und den in AW2 durch die Analyse von vergleichbaren Projekten gewonnenen Ideen (vgl. Johannsen [2010]) basiert die Zielsetzung für dieses Thesis Outline.

2 Zielsetzung

Um die Zielsetzung der Arbeit zu erläutern werden zuerst die Schwächen der bisherigen Kommunikationslösung im Living Place aufgezeigt. Darauf aufbauend werden die Ziele definiert.

2.1 Problemstellung

Die bisherige Lösung für die Kommunikation im Living Place Hamburg sieht einen ActiveMQ Server als Message-Broker vor. Zudem ist das Format der Nachrichten auf JSON festgelegt worden. Diese Spezifikationen betreffen folglich nur die Übertragungen von Nachrichten. Wie die Kommunikation zu standen kommt beziehungsweise wie eine Anwendung ihre Zielgeräte findet oder woher die Anwendung die nötigen Informationen zum Ansteuern des entsprechenden Geräts hat ist bisher nicht festgelegt.

Wenn beispielweise eine Anwendung einen Aktor ansteuern oder von einem Sensor Daten abfragen will, so brauch die Anwendung beziehungsweise ihr Entwickler unter anderem folgende Informationen:

ActiveMQ-Topic

Unter welchem ActiveMQ-Topic wird das Zielgerät erreicht.

Protokoll-Informationen

Welches Protokoll erwartet das Zielgerät

Kontext-Informationen

Warum ist das geeignete Gerät das richtige für die Aufgabe. Zum Beispiel Ort oder Funktionalität.

Diese Informationen liegen also in der entsprechenden Anwendung. Greifen mehrere Anwendungen auf das selbe Gerät zu, so sind diese Informationen zudem redundant vorhanden. (vgl. Grafik 2.1)

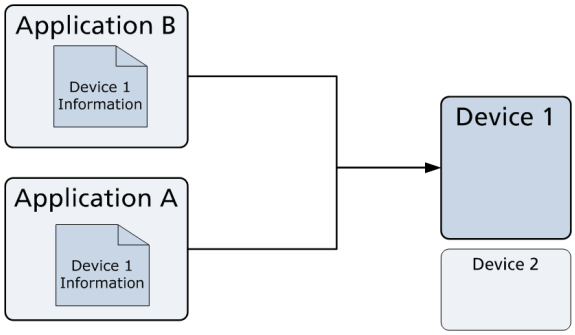

Abbildung 2.1: Bisherige Struktur

Als Konsequenz aus diesen Bedingungen sind die Anwendung sehr statisch. Gibt es Änderungen am Kommunikationsweg, am Zielgerät oder fällt dieses komplett aus und wird getauscht, so muss die Anwendung entsprechend angepasst werden. Dies bedeutet einen immensen Wartungsaufwand und schränkt die Flexibilität drastisch ein. Jeder Komponententausch bringt so einen deutlichen Mehraufwand mit sich.

2.2 Ziele der Arbeit

Diese statische Struktur soll ersetzt werden. Hierzu wird jedes Gerät durch ein Software-Modell dargestellt. Diese Modelle werden von einer Serverkomponente verwaltet die eine Schnittstelle für Abfragen über diese Modelle anbietet. Somit liegen alle gerätespezifischen Informationen zentral und vor allem unabhängig von den eigentlichen Anwendungen vor. Diese Struktur ist in Abbildung 2.2 dargestellt.

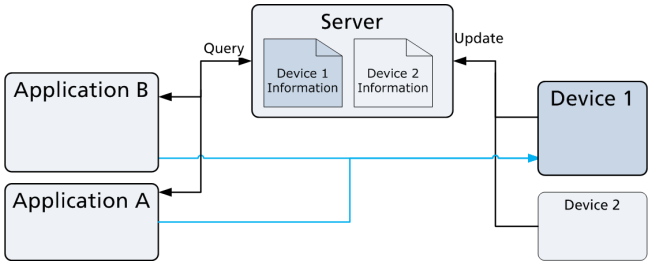

Abbildung 2.2: Lösungsansatz

Somit brauchen die Anwendung keine Informationen über konkrete Geräte und sind von diesen unabhängig.

Die dafür nötigen Aufgaben stellen die Zielsetzung der Arbeit dar:

Modellierung der Wohnung aus Funktionaler Sicht

Die Geräte müssen durch ein Modell repräsentiert werden. Diese Modellierung soll flexibel und generisch gehalten sein um alle erdenklichen Geräte abbilden zu können.

Infrastruktur zur Verwaltung des Modells und seiner Instanzen

Es müssen Abfragen über das Modell und die konkreten Instanzen möglich sein. Ebenso die Aktualisierung von Instanzen.

Entwicklung einer API für die Anwendungen

Um den Anwendungsentwicklern eine möglichst einfache Integration ihrer Anwendung in die Infrastruktur zu ermöglichen ist eine entsprechende API zu entwickeln.

2.3 Abgrenzung

Um den Rahmen für die Arbeit abzustecken werden folgende Abgrenzungen vorgenommen:

Physikalische Kommunikationswege

Es wird angenommen das eine physikalische Kommunikationswege zwischen den Endgeräten vorhanden sind. Deren Bereitstellung ist nicht Bestandteil dieser Arbeit.

Kein komplettes Modell

Das entwickelte Modell soll nicht alle vorhandenen Geräte erstellt konkret modellieren sondern nur die Möglichkeit biete dies möglichst generisch und einfach umzusetzen. Zu Testzwecken werden einige Geräte konkret modelliert um die Funktionalität zu prüfen.

Netzwerksicherheit

Der Aspekt der Netzwerksicherheit wird hier ebenfalls nicht betrachtet. Es wird nicht geprüft ob ein Client eine Anfrage stellen darf oder ob eine Antwort vom korrekten Server kommt. Dies wird als korrekt angenommen.

3 Lösungsansatz

Die in Kapitel 2 Zielsetzung gesteckten Ziele sollen mittels des im Folgenden vorgestellten Lösungsansatzes erreicht werden. Hierbei wird zuerst ein Überblick gegeben und danach werden die voraussichtlich eingesetzten Technologien vorgestellt.

3.1 Überblick

Durch einen zentralen Server werden Softwaremodelle der realen Geräte verwaltet. Um nun ein Gerät anzusprechen stellen die Anwendungen entsprechende Abfragen an den Server (1). Mit Hilfe der Antwort (2) kann die Anwendung dann direkt mit dem Gerät kommunizieren (3). Damit das Modell konsistent und aktuell bleibt muss das Gerät seine Instanz bei Zustandsänderungen aktualisieren (4).
Dieser Vorgang ist in Grafik 3.1 dargestellt. Als Beispiel versucht hier die Anwendung ein Fenster in der Küche zu öffnen:

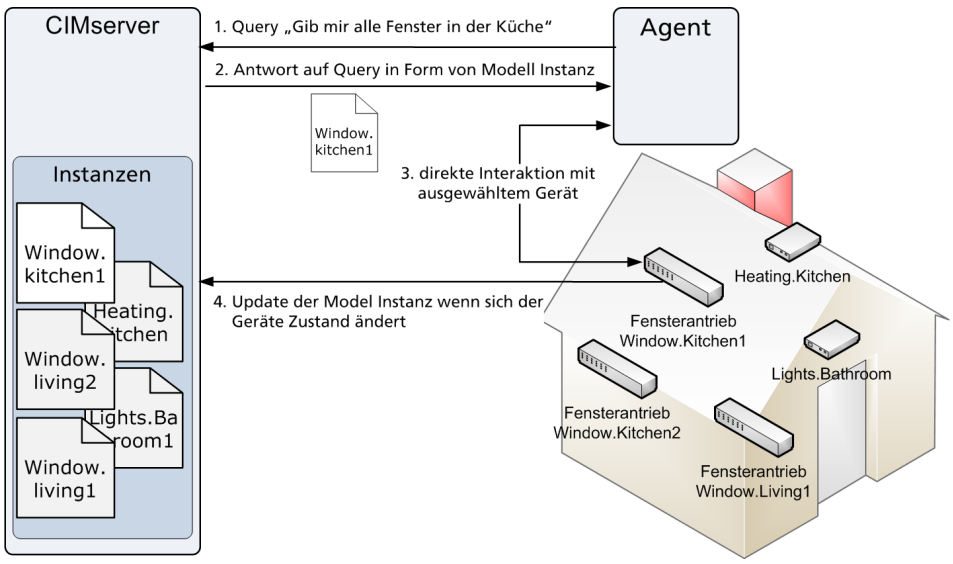

Abbildung 3.1: Schematische Darstellung der Kommunikationswege

3.2 Eingesetzte Technologien

Die Umsetzung dieser Struktur wird voraussichtlich mit Hilfe der folgenden Technologien erfolgen:

3.2.1 CIM - Common Information Modell

Das Common Information Model, kurz CIM, wurde 1997 von der Distributed Managment Task Force (DMTF) veröffentlicht. Die DMTF ist eine Organisation mehrerer Unternehmen um gemeinsame Standards zu entwickelt. Ihr gehören unter anderem Firmen wie Cisco, Compaq, Dell, HP, IBM, Intel und Microsoft an. CIM bietet ein Datenmodell mit dem Systemkomponenten, Hardware wie auch Software, beschrieben werden können. Es ist Bestandteil der „Web Based Enterprise Management“ (WBEM) Standards die dazu dienen große, verteilte Systeme wie beispielsweise Rechenzentren und Serverfarmen besser verwalten zu können. (vgl. Johannsen [2010])

Durch die Möglichkeit das Modell durch eigene Typen und Modelle zu erweitern bietet sich CIM für die Modellierung der Living Place Geräte an.

Sblim SFCB - Small Footprint CIM Broker

Sblim (gesprochen „sublime“) ist ein OpenSource Projekt das Werkzeuge und Programme für den Einsatz von WBEM Funktionalitäten unter Linux bereitstellt. Hierzu zählt auch der „Small Footprint CIM Broker“, kurz SFCB. Dies ist ein CIM Server der vor allem auf geringen Ressourcenbedarf hin konzipiert ist um unter anderem auf Embedded Plattformen eingesetzt zu werden.
Er wird kontinuierlich weiterentwickelt und üblicherweise halbjährlich aktualisiert. Zum Zeitpunkt dieser Ausarbeitung ist die Version 1.3.10 aktuell und wird voraussichtlich eingesetzt werden. (vgl. Sblim project [2011])

MOF - Managed Object File

Um ein Modell oder seine Instanzen zu beschreiben wird eine textuelle Beschreibung verwenden. Diese „Managed Object Files“, kurz „MOF“, werden dazu verwendet eine maschinenlesbare Form der Modelle zu erhalten. (vgl. Hobbs [2004])

Listing 3.1: Beispiel für ein Managed Object File

```
Class WindowDriver:CIM_Device {
  [read, Units("cm")] uint8 max_position;
  [read, Units("cm")] uint8 position;
  uint8 close();
  uint8 gotoPosition(IN uint8 position);
};

Instance of WindowDriver {
  DeviceID = "iFlat-window03";
  Description = "Suedseite Oberlicht 3";
  max_position = 20;
};
```

CIM Provider

Soll ein Modell beziehungsweise ein MOF über den SFCB verwaltet werden können, wird ein sogenannter „Provider“ für dieses MOF benötigt. Dieser Provider wird mittels eines MOF-Compilers des SFCB erzeugt. Das Laden von Providern erfordert allerdings einen Neustart des SFCB womit ein dynamisches Nachladen im Betrieb nicht möglich ist. (vgl. Sblim project [2011])

CIM-XML

Für die Kommunikation zwischen CIM-Server und Clients wird CIMXML eingesetzt. Dieses Format ist XML-basiert und ermöglicht die Formulierung der Abfragen an den Server. (vgl. Hobbs [2004])

Listing 3.2: Beispiel für eine CIM-XML Abfrage

```
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
        <MESSAGE ID="12345" PROTOCOLVERSION="1.0">
                <IMETHODCALL NAME="EnumerateInstanceNames">
                        <IPARAMVALUE NAME="ClassName">
                                <CLASSNAME NAME="Window" />
                        </IPARAMVALUE>
                </IMETHODCALL>
        </MESSAGE>
</CIM>
```

3.2.2 Java API

Damit die Anwendungen und ihre Entwickler mit möglichst wenig Aufwand diese Lösung integrieren können wird eine entsprechende API für Java entwickelt. Aus dem sblim Projekt gibt es ein CIM Client Package for Java. Auf diesem basierend soll die API die wesentlichen Funktionalitäten für den Living Place Anwendungsentwickler kapseln um ihm so die Arbeit zu erleichtern. (vgl. Sblim project [2011])

3.2.3 Anbindung ActiveMQ

Als Kommunikationsweg wird die bereits bestehende Kommunikationslösung im Living Place, ein ActiveMQ Server, genutzt. Auf diese Weise kann in der Erprobungs- und Übergangsphase ein paralleler Betrieb gewährleisten werden. (vgl. Voskuhl und Otto [2010])
Sowohl die Modellspezifische Kommunikation wird über ActiveMQ abgehandelt sowie auch die direkte Kommunikation zwischen Gerät und Anwendung. Dementsprechend werden die ActiveMQ-Attribute eines teilnehmenden Geräts im Modell modelliert.

3.3 Architekturübersicht

Grafik 3.2 zeigt das Zusammenspiel der Komponenten.

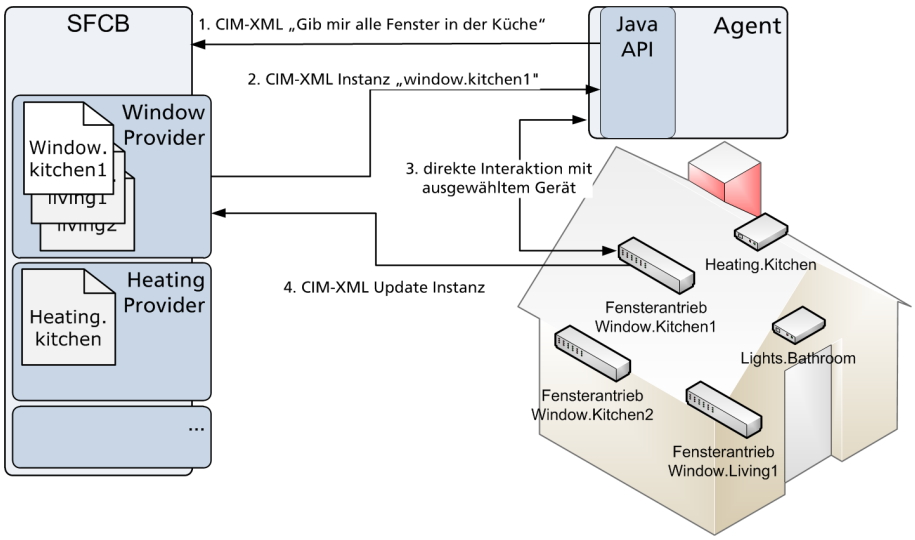

Abbildung 3.2: Zusammenspiel der Technologien

3.4 Geplantes Vorgehen

Das geplante Vorgehen für die Umsetzung des in 3 Lösungsansatz erläuterten Lösungsansatzes wird im Folgenden erläutert.

Im Rahmen von Projekt2 werden die technischen Grundlagen erarbeitet. Es wird testweise ein SFCB installiert und in Betrieb genommen und ein Beispielmodell eingerichtet um die Basis für die spätere Arbeit zu legen.
Ist dieses Ziel erreicht, so kann mit der eigentlichen Kernaufgabe, der Modellierung begonnen werden. Für die Entwicklung des endgültigen Modells wird eine umfangreiche Analyse aller bisher im Living Place eingesetzten und perspektivisch eingesetzten Geräte betrieben. Aus diesen Daten soll dann eine entsprechende Grundhierarchie abstrahiert werden.

Um die Funktionsfähigkeit zu demonstrieren wird ein einfaches Testszenario implementiert. Hierzu könnte beispielsweise der in Projekt1 entwickelte „Raumklima-Agent“ auf das modelbasierte System umgeschrieben werden.

Als letzter Schritt steht dann die Validierung der Lösung in Form von Lasttests an. Der Einsatz im Living Place wird eine hohe Anzahl von Instanzen und Providern erfordern so dass hier getestet werden muss, wie viele Instanzen bei akzeptabler Latenz bedient werden können.

4 Herausforderungen

Die Modellierung, die Modell-Konsistenz und die Skalierbarkeit werden als größte Knackpunkte erwartet. Diese werden im Folgenden näher erläutert.

4.1 Modellierung

Das entwickelte Modell für das Living Place Hamburg soll maximal flexibel und dennoch allgemeingültiges sein. Erweist sich das Modell als zu komplex zu umfangreich oder als zu konkret, so wird der Programmieraufwand für die Anwendungsentwickler zu groß und die Fehleranfälligkeit steigt.

4.2 Konsistenz

Eine Modell-Instanz muss immer den aktuellen Zustand des realen Geräts repräsentieren. Ist dies nicht der Fall ist das Modell nutzlos. Folglich muss gewährleistet werden das beide Zustände konsistent sind.

4.3 Skalierbarkeit

Die im Living Place angedachte Menge an Geräten ist hoch und wird mit der Zeit weiter steigen. Dementsprechend muss die entwickelte Lösung performant genug sein und entsprechend gut skalieren um auch in Zukunft als Kommunikationslösung eingesetzt werden zu können.

5 Zusammenfassung

Der Aufbau einer modelbasierten Kommunikationsinfrastruktur bietet für das Living Place Hamburg die Möglichkeit Anwendungen unabhängiger von der konkreten Geräte-Konfiguration zu gestalten. Die Komplexität der Geräteansteuerung würde weitestgehend vom Anwendungsentwickler fern gehalten so dass er sich mehr auf seine eigentlichen Aufgaben im Bereich kontextbasierte Ambient Intelligence fokussieren kann. So würden der Auf- und Ausbau des Living Place Hamburgs deutlich erleichtert.
Im Kern geht es hier nicht um die Neuentwicklung einer Technologie sondern um die Adaption einer Technik aus dem WBEM für eine neue Domäne, der Ambient Intelligence.

Als möglicher Knackpunkt bei der Entwicklung dieser Lösung könnte sich die Modellierung der Wohnung aus funktionaler Sicht erweisen. Ist das entwickelte Modell beziehungsweise die Klassenstruktur nicht übersichtlich und zugleich vielseitig, so muss sich der Anwendungsentwickler wieder intensiv damit beschäftigen. Somit wäre das eigentliche Ziel nicht erreicht.
Dementsprechend liegt hier der Schwerpunkt der Arbeit um diese Herausforderung erfolgreich zu bewältigen.

Literaturverzeichnis

[Amigo Consortium 2005] Amigo Consortium: *D2.1 Specification of the Amigo Abstract Middleware Architecture*, 2005. – URL http://www.hitech-projects.com/euprojects/amigo/deliverables/Amigo_WP2_D2.1_v10%20final.pdf. – Zugriffsdatum: 24.05.2010

[Distributed Management Task Force, Inc. 2010] Distributed Management Task Force, Inc.: *Common Information Model - CIM Schema: Version 2.25.0*, 2010. – URL http://www.dmtf.org/standards/cim/cim_schema_v2250. – Zugriffsdatum: 18.05.2010

[Eisenhauer u. a. 2009] Eisenhauer, M. ; Rosengren, P. ; Antolin, P.: A Development Platform for Integrating Wireless Devices and Sensors into Ambient Intelligence Systems. In: *Proc. 6th Annual IEEE Communications Society Conf. Sensor, Mesh and Ad Hoc Communications and Networks Workshops SECON Workshops '09*, 2009, S. 1–3

[Hobbs 2004] Hobbs, Chris: *A Practical Approach to WBEM/CIM Managment*. Auerbach Publications, 2004. – 325 S. – ISBN 0-8493-2306-1

[Hydra Consortium 2007] Hydra Consortium: *D3.4 Initial architectural design specification*, 2007. – URL http://www.hydramiddleware.eu/hydra_documents/D3.4_Initial_Architectural_Design_Specification.pdf. – Zugriffsdatum: 18.05.2010

[Johannsen 2009] Johannsen, Benedikt: Die Anforderungsanalyse an Feldbussystemen für Ambient Living / HAW Hamburg. HAW Hamburg, 2009. – Forschungsbericht. – URL http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw1/johannsen/bericht.pdf. – Zugriffsdatum: 09.03.2010. AW1 Ausarbeitung WS2009/10

[Johannsen 2010] Johannsen, Benedikt: Netzwerk Infrastrukturen für Ambient Living-Anwendungen / HAW Hamburg. HAW Hamburg, 2010. – Forschungsbericht. – URL http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw2/johannsen/folien.pdf. – Zugriffsdatum: 13.06.2010

[Kostelnik u. a. 2008] Kostelnik, Peter ; Sarnovsky, Martin ; Hreno, Jan ; Ahlsen, Matts ; Rosengren, Peter ; Kool, Peeter ; Axling, Mathias: Semantic Devices for Ambient Environment Middleware. In: *ACM* 1 (2008), S. 1–6

[von Luck u. a. 2010] Luck, Prof. Dr. K. von ; Klemke, Prof. Dr. G. ; Gregor, Sebastian ; Rahimi, Mohammad A. ; Vogt, Matthias: Living Place Hamburg – A place for concepts of IT based modern living / Hamburg University of Applied Sciences. URL http://livingplace.informatik.haw-hamburg.de/content/LivingPlaceHamburg_en.pdf, Mai 2010. – Forschungsbericht

[Pautz 2009] Pautz, Alexander: Analyse von Feldbussystemen in Hinblick auf Ambient Living / HAW Hamburg. HAW Hamburg, 2009. – Forschungsbericht. – URL http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw1/pautz/bericht.pdf. – Zugriffsdatum: 12.03.2010. AW1 Ausarbeitung WS2009/10

[Pautz 2010] Pautz, Alexander: Vermittlungsinfrastrukturen in komplexen Netzwerken / HAW Hamburg. HAW Hamburg, 2010. – Forschungsbericht. – URL http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw2/pautz/folien.pdf. – Zugriffsdatum: 22.06.2010

[Sarnovský u. a. 2008] Sarnovský, Martin ; Kostelník, Peter ; Hreno, Ján ; Butka, Peter: Device Description in HYDRA Middleware. In: *ACM* 1 (2008), S. 1–4

[Sblim project 2011] Sblim project: *Sblim project*. 2011. – URL http://sblim.wiki.sourceforge.net/

[Voskuhl und Otto 2010] Voskuhl, Sören ; Otto, Kjell: Entwicklung einer Architektur für den Living Place Hamburg / HAW Hamburg. HAW Hamburg, 2010. – Forschungsbericht. – URL http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2010-proj1/otto_voskuhl.pdf. – Zugriffsdatum: 09.10.2010. PJ1 Ausarbeitung SoSe2010