



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Seminararbeit

Konrad Glugla

Das iPhone im intelligenten Haus

Konrad Glugla
Das iPhone im intelligenten Haus

Seminararbeit eingereicht im Rahmen der Veranstaltung Anwendungen I
im Studiengang Informatik (Master of Science)
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Kai von Luck

Abgegeben am 25. Juli 2008

Inhaltsverzeichnis

- Abbildungsverzeichnis** **4**

- 1 Einführung** **5**
 - 1.1 Vision 5
 - 1.2 Motivation 6
 - 1.3 Aufbau der Arbeit 6

- 2 Personal Information Environments** **7**
 - 2.1 PIE Architektur 7

- 3 Apple iPhone** **9**
 - 3.1 Motivation 9
 - 3.2 iPhone SDK 9
 - 3.3 Potenzial und Ideen 10

- 4 Das iFlat der HAW** **12**
 - 4.1 Event Manager 13

- 5 Ausblick** **14**

- 6 Zusammenfassung** **15**

- Literaturverzeichnis** **16**

Abbildungsverzeichnis

2.1	PIE Infrastruktur (Quelle: [IBM08])	8
3.1	iPhone OS Technologie Ebenen (Quelle: [APP08a])	10
4.1	Konzept der Kommunikations-Infrastruktur im iFlat (Quelle: [Hol08])	13

1 Einführung

Die bereits von mehreren Masterstudenten der HAW formulierte Vision vom intelligenten Haus¹ soll in dieser Arbeit weiter behandelt werden. Dabei liegt der Fokus dieser Ausarbeitung auf dem Einsatz eines mobilen Geräts als Teil des intelligenten Hauses. Der Einsatz kann als Steuereinheit in Form einer Fernbedienung oder auch als zusätzliches Endgerät, z.B. zur Synchronisation mit Outlook oder Musik-Playlisten, konzipiert werden. Die Einsatzmöglichkeiten sowie die Kommunikationsinfrastruktur in welcher das mobile Gerät eingesetzt werden soll, werden in dieser Arbeit besprochen.

1.1 Vision

Die Vision vom intelligenten Haus knüpfen an die Ideen der Heimautomatisierung an. Heimautomatisierungen werden auch in Deutschland mittlerweile von mehreren Firmen angeboten. Die Idee dabei ist, in einer Zeit in welcher der Alltag immer hektischer wird, den Bewohnern bei alltäglichen Sachen durch technische Hilfsmittel zu unterstützen. Zusammengefasst ergeben sich dabei folgende Ziele:

- Lebensqualität verbessern
- Komfort erhöhen
- Kosten sparen (z.B. Energiekosten)
- Sicherheit erhöhen

Beispielhaft sei folgend ein Szenario der Heimautomatisierung aufgezeigt, welches bereits angeboten wird:

«Durch Aktivierung des „Nachtmodus“ über eine Bedieneinheit werden alle Türen verschlossen. Heizungen werden heruntergefahren. Lichter gehen aus. Bewegt man sich während des „Nachtmodus“ im Gebäude, z.B. um ins Badezimmer zu gelangen, schaltet sich an den Aufenthaltsorten automatisch für das Auge angenehmes gedimmtes Licht an.»

Die Heimautomatisierung ist immer noch ein Luxusprodukt und verteilt sich auf viele Einzellösungen. Jedes Unternehmen bietet seine eigenen Standards an die sich nicht kombinieren lassen. Vor allem die Geräte die jeder schon besitzt, wie z.B. der handelsübliche TV-Bildschirm oder das Mobiltelefon, lassen sich nicht in die Heimautomatisierung integrieren.

Die Idee vom intelligenten Haus verfolgt dieselben Ziele, verwendet aber einen etwas anderen Ansatz. In einem intelligenten Haus kennen alle Geräte ihre Umgebung und kommunizieren

¹Markus Dreyer, Dennis Hollatz, Sven Vollmer u.a.

miteinander. Neue Geräte lassen sich leicht integrieren und eine Personalisierung der Dienste ist möglich. Zusätzlich könnte sich das System individuell dem Lebensstil des Bewohners anpassen.

1.2 Motivation

Die Vision vom intelligenten Haus ist kein neues Thema. Geforscht wird zu diesem Thema z.B. an der Stanford University ([[STA08](#)]), dem Fraunhofer Institut ([[FRA08](#)]) und der Hochschule Luzern ([[CEE08](#)]). Das meiste stützt sich dabei auf Ideen und Untersuchungen mit wenig konkreten Implementierungen. An der Stanford University wurde beispielsweise ein „iRoom“ aufgebaut, welcher einen Konferenzraum darstellt in dem technische Geräte miteinander vernetzt sind und interagieren, um das gemeinsame Arbeiten zu optimieren. Der iRoom wurde an der HAW bereits hauptsächlich vom Masterstudenten Dennis Hollatz untersucht [[Hol07](#)]. Auf die dem iRoom zugrunde liegende Architektur soll in dieser Arbeit ebenfalls kurz eingegangen werden.

An der HAW wurde bereits mit dem Aufbau eines Labors für das „Interactive Living“ begonnen. Unter dem Pseudonym iFlat soll das Leben in einem intelligenten Haus simuliert werden. Im iFlat soll es irgendwann möglich sein verschiedene Szenarien durchzuspielen, wie z.B. die automatische Rufumleitung auf die Stereoanlage mit integrierten Mikrofonen für ein freihändiges Telefonieren beim Kochen. Szenarien des „Interactive Living“ sind z.B. in [[Dre07](#)] ausführlich beschrieben.

Als mobiles Endgerät für die Entwicklung von eigenen Applikationen sind momentan das iPhone von Apple sowie Google's Android sehr attraktiv. Für Android gibt es momentan noch keine fertigen Endgeräte, so dass Implementierungen auf einem Simulator ausgeführt werden müssen. Für eine Arbeit mit dem iPhone im intelligenten Haus spricht vor allem die intuitive Bedienmöglichkeit über ein Touchscreen. Da zudem der Masterstudent Sven Vollmer sich bereits mit Android zur Nutzung als zentrale Fernbedienung beschäftigt (siehe [[Vol07](#)]), sollen in dieser Arbeit die Möglichkeiten des iPhone untersucht werden.

1.3 Aufbau der Arbeit

Zunächst wird in Kapitel 2 die Grundlage für die kombinierte Interaktion mit mehreren EDV-Geräten beschrieben. Hierfür wird die von Jeff Pierce beschriebene Idee von Personal Information Environments erläutert.

Nach dieser Grundlage wird ein Blick auf die Möglichkeiten des iPhone gelegt, und gleichzeitig deren Einsatzmöglichkeiten als mobiles Endgerät im intelligenten Haus untersucht.

Kapitel 4 beschäftigt sich mit dem iFlat der HAW und geht insbesondere auf die bereits entwickelten Ideen und Umsetzungen ein.

In einem anschließendem Ausblick werden die Einsatzmöglichkeiten des iPhone im iFlat besprochen. Es werden Ideen und Probleme angesprochen sowie Anregungen auf weiterführende Untersuchungen gegeben.

Das letzte Kapitel fasst noch einmal die Kernpunkte der vorliegenden Seminararbeit zusammen.

2 Personal Information Environments

Durch den stetigen Leistungsanstieg und dem gleichzeitigen Preisverfall der EDV-Geräte sind wir mittlerweile in einer Zeit angelangt, in der Anwender mit einer Vielzahl von heterogenen Geräten interagieren (z.B. Notebooks, PDA's, Mobiltelefone). Diese Entwicklung hat sowohl bei Arbeitsplätzen als auch in privaten Haushalten stattgefunden.

Unter dem Begriff Personal Information Environments (PIE) wird bei IBM Research in Zusammenarbeit mit der Georgia Tech untersucht, in wie weit sich Schnittstellen der heutigen Zeit verändern könnten, bzw. sollten, um den Wandel von der Single-Device Nutzung hin zu einer kombinierten Interaktion aller Geräte miteinander zu gestalten ([IBM08] und [GAT08]). Dieser Begriff wird auch in einer Seminararbeit von Dennis Hollatz [Hol07] kurz erläutert.

Die grundlegende Idee von PIE ist, die Interaktion mit mehreren EDV-Geräten für den Anwender zu vereinfachen und einen verbesserten Nutzen der einzelnen Geräte durch einen koordinierten Einsatz aller Geräte miteinander zu erzielen. Im heutigen Zustand arbeiten alle Geräte für sich alleine und wissen nichts von der Existenz anderer Geräte in ihrer Umgebung. Das Herstellen von Verbindungen zwischen Geräten, z.B. zum Datenaustausch, muss manuell eingeleitet werden. In einer PIE sollen sich die Geräte untereinander kennen und über eine gemeinsame Schnittstelle kommunizieren. Beispielhaft soll ein DVD-Rekorder wissen, dass die Videoausgabe auf allen Geräten des Typs „Display“ durchgeführt werden kann, und über eine gemeinsame Schnittstelle hat der DVD-Rekorder die Möglichkeit jedes in der Nähe befindliche „Display“ anzusteuern. Die Kommunikation kann dabei z.B. über WLAN oder Bluetooth erfolgen.

2.1 PIE Architektur

Unter der Leitung von Jeff Pierce wurde bei IBM Research eine prototypische Infrastruktur für die PIE entwickelt. Diese besteht grundlegend aus 3 Elementen (siehe Abbildung 2.1). Dem PIE Server, der die Aufgabe hat die Devices mit den Usern zu verbinden und Nachrichten zwischen den Devices zu routen, den PIE Clients, welche lokal auf jedem Device laufen und als Verbindungspunkt für die Kommunikation zwischen Services und Server dienen, und den PIE Services, welche von Entwicklern programmiert werden müssen um eine Multi-Device Funktionalität zu unterstützen.

- **PIE Server**

Der PIE Server ist in diesem Modell ein modifizierter Instant Messaging Server, der die

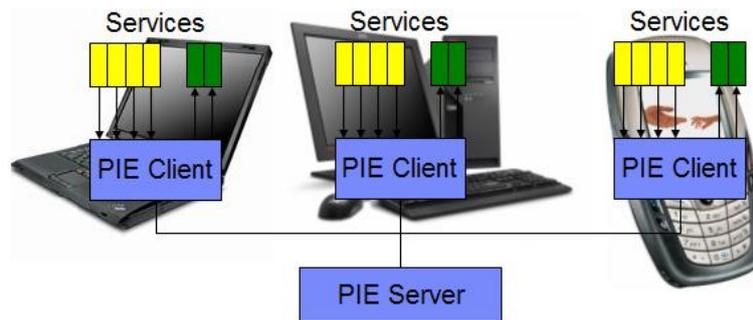


Abbildung 2.1: PIE Infrastruktur (Quelle: [IBM08])

Standard IM Funktionalitäten wie z.B. Registrierung und Authentifikation von Usern anbietet. Ausserdem bietet er die Funktionalität des Routens von Nachrichten zwischen Entitäten.

Einige Punkte die ergänzt wurden sind die Möglichkeit Alias zu definieren um die Nachrichten-Adressierung zu vereinfachen, sowie eine verbesserte Unterstützung von asynchroner Kommunikation.

- **PIE Client**

Der PIE Client stellt ein User Interface zur Verfügung über den der Status der Devices sichtbar wird. Er kann Devices hinzufügen oder entfernen und dient zur Verwaltung vorhandener Services.

Außerdem dient er als Schnittstelle für den Nachrichtenaustausch der Services. D.h. bevor eine Nachricht vom Service beim Server eintrifft wird diese Nachricht vom Client bearbeitet. Empfängt der Client eine Nachricht vom Server, wird diese ausgewertet. Beinhaltet die Nachricht die Anmeldung eines neuen Devices, so erhält jeder die Nachricht und der Client aktualisiert die Liste der verfügbaren Services für seinen Service. Bei anderen Nachrichten wird der XML namespace der Nachricht ausgewertet und die Nachricht an die Services weitergeleitet die an der Nachricht interessiert sind.

- **PIE Services**

Der PIE Client unterstützt 2 Arten von services: dependent (abhängige) und independent (eigenständige) services.

Dependent services (in Abbildung 2.1 in grünen Kästchen dargestellt) werden automatisch vom Client instanziiert und initialisiert. Durch diesen Vorgang laufen dependent services im selben Prozess und können so direkt auf die API's der anderen dependent services zugreifen.

Independent services (in Abbildung 2.1 in gelben Kästchen dargestellt) laufen komplett eigenständig und verbinden sich zum Client via eines local sockets. Bei independent services handelt es sich um standalone services oder plug-ins für bestehende Applikationen (wie z.B. plug-ins für Firefox). Independent services kommunizieren mit dem Client über XML-Fragmente.

Die Aspekte einer PIE beinhalten die Grundgedanken des intelligenten Hauses. Bei der Konzeption vom intelligenten Haus bietet es sich an die Erkenntnisse der PIE Architektur zu Nutzen zu ziehen.

3 Apple iPhone

3.1 Motivation

Das Apple iPhone war bereits direkt nach Markteinstieg in aller Munde. In den ersten 8 Monaten konnte das iPhone bei Smartphones in den USA einen Marktanteil von 28% erreichen. Auch Unternehmen und öffentliche Institutionen wie The Walt Disney Company, Genentech Inc. oder auch der U.S. Army nutzen mit positivem Feedback das iPhone für ihre täglichen Arbeiten [[APP08b](#)].

Das iPhone erlangte diese hohe Akzeptanz vor allem durch eine intuitive Bedienung per Multi-Touch, einem großen Display und der guten Internetunterstützung durch WLAN-Funktionalität und ausgereiftem Browser (Apple Safari).

Im März 2008 gab es eine Ankündigung von Apple bei der unter anderem Funktionserweiterungen für das iPhone vorgestellt wurden [[APP08c](#)]. Unter den Erweiterungen sind unter anderem ein Microsoft Exchange Support und die Unterstützung von Cisco's IPsec VPN. Auch bei der WWDC (Worldwide Developers Conference) im Juni 2008 wurden wieder neue Features vorgestellt die im Umfang der iPhone Version 2.0 enthalten sind, oder für spätere Versionen geplant sind. Darunter die Unterstützung von Microsoft Office Dokumentformaten, ein zusätzlicher Language-Support, vor allem für die asiatischen Sprachen, sowie ein Push Notification Service der es ermöglicht dem User eine Meldung auf sein iPhone zu „pushen“ sobald er z.B. eine neue Nachricht in seinem Instant-Messaging Dienst erhalten hat, ohne dass die Applikation gestartet sein muss.

Die stetige Erweiterung des Funktionsumfangs ist auch auf der offiziellen Homepage von Apple erkennbar, auf der in regelmäßigen Abständen neue Ankündigungen und Updates für das iPhone zu finden sind. Die Einsatzmöglichkeiten des iPhone scheinen durch die vielen Funktionen und Erweiterungsmöglichkeiten unbegrenzt.

Auch in diversen Universitäten (Stanford-, Harvard-, Yale University etc.) wird das iPhone zu Forschungszwecken eingesetzt, was vor allem durch die Möglichkeiten der Applikationsentwicklung auf dem iPhone, durch die Bereitstellung der iPhone SDK von Apple, ausgelöst wurde. Auch die Unterstützung von GPS oder Triangulation zur Ortsbestimmung über Wi-Fi Hotspots und Funkmasten machen eine Entwicklung auf dem iPhone interessant.

3.2 iPhone SDK

Im März 2008 hat Apple offiziell das iPhone SDK angekündigt und öffnete damit die nativen API's und Tools mit denen alle von Apple entwickelten iPhone Applikationen erstellt wurden. Für

freie Entwickler bedeutet dies, dass iPhone Applikationen nun mit den gleichen Voraussetzungen entwickelt werden können, mit denen auch die Entwickler bei Apple arbeiten.

Der wichtigste Teil einer SDK ist der Satz an API's, also die Plattform. Dabei baut Apple mit dem iPhone OS auf jahrelanger Erfahrung auf, da es auf Mac OS X basiert. Mac OS X gehört auf dem heutigen Markt zu einem der führenden Plattformen. Das iPhone OS besteht, wie das Mac OS X, aus 4 Ebenen, die nachfolgend beschrieben sind.

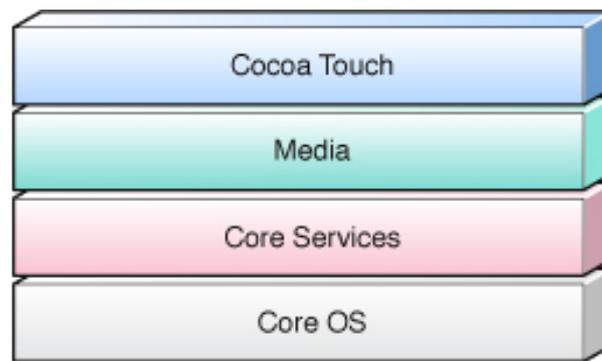


Abbildung 3.1: iPhone OS Technologie Ebenen (Quelle: [APP08a])

- **Core OS**

Das Core OS beinhaltet Kernfunktionalitäten wie das File-System, Power-Management und Security Funktionalitäten.

- **Core Services**

Die Core Services beinhalten Basisfunktionalitäten wie Netzwerk-Service, Threading, SQLite (als Datenbank API) und Core Location für die Positionsbestimmung über Wi-Fi Hotspots und Funkmasten (Triangulation), d.h. es können auch positionsabhängige Applikationen entwickelt werden.

- **Media**

Eine Ebene höher haben wir den Media Layer. Dort befinden sich die Basisfunktionalitäten für die 2D und 3D Darstellung, Audio und Video Unterstützung. Diese Ebene beinhaltet die C-basierten Technologien OpenGL ES (Embedded Version of OpenGL) für 3D Grafiken, OpenAL (API für 3D Sound) sowie Core Animation zur Entwicklung von Animationen.

- **Cocoa Touch**

Cocoa Touch ist das User Interface Application Framework für das iPhone und beinhaltet alle Funktionalitäten für den In- und Output mit dem iPhone. Darunter die Funktionalitäten für den Einsatz von Multi-Touch (Multi-Touch Events und Multi-Touch Controls), dem Accelerometer mit 3 Achsen-Sensor über welcher die Lage des iPhone erkannt wird, und viele weitere.

3.3 Potenzial und Ideen

Für die Kommunikation im intelligenten Haus sind die integrierten WLAN- und Bluetooth-Funktionalitäten ausreichend. Für den Einsatz des iPhone als Fernbedienung im intelligenten

Haus scheinen Funktionen wie die Ortsbestimmung über GPS oder Triangulation und die gute Webanbindung sehr interessant zu sein. Mit der Ortsbestimmung könnte man sich Szenarien vorstellen bei der das intelligente Haus erkennt wenn sich der Bewohner auf dem Weg nach Hause befindet und Heizungen oder Klimaanlage aktivieren.

Über die Webanbindung könnte man z.B. eine Applikation bauen mit der das Fernsehprogramm über das Internet auf das iPhone geladen wird und nach Auswahl des gewünschten Programms der am nächsten gelagerte Bildschirm automatisch mit dem Programm eingeschaltet wird, oder auch ein DVD-Rekorder zur Aufnahme programmiert wird.

Mit der neuen Funktion des Push Notification Service könnten Überwachungsfunktionen für das intelligente Haus realisiert werden. Als Beispiel könnte der Herd eine Nachricht auf das iPhone schicken bevor das Essen anbrennt, weil der Bewohner mal wieder in ein Telefongespräch vertieft ist.

Zum Einsatz im Bürozimmer eignet sich z.B. der Microsoft Exchange Support zur Synchronisation mit Kontakten, Terminen, E-Mails etc.

Die Möglichkeiten der intuitiven Bedienung sind ein großes Potenzial des iPhone auf das bei Applikationsentwicklung besonders geachtet werden sollte. Eine Fernbedienung muss leicht zu verstehen und komfortabel in der Bedienbarkeit sein. Funktionen wie Multi-Touch und der Accelerometer zur Steuerung finden bei Usern überwiegend eine gute Akzeptanz und haben zudem auch noch einen kleinen Unterhaltungswert.

4 Das iFlat der HAW

An der HAW Hamburg haben bereits erste Untersuchungen für das intelligente Haus begonnen. Unter dem Projektnamen „iFlat“ entsteht ein Labor in dem verschiedene Szenarien des Wohnens im intelligenten Haus simuliert werden sollen. Für den Anfang sollen folgende Komponenten in das iFlat integriert werden:

- Lautsprecher
repräsentieren eine Stereo-Anlage
- Bildschirm
- RFID Regal
Ein Regal das über RFID-Tags erkennt, welche Gegenstände im Regal liegen. Diese Komponente wurde bereits erfolgreich umgesetzt und ist unter anderem bei [Uri08] dokumentiert.
- Mobile Geräte
z.B. das Mobiltelefon bzw. das Smartphone

Die Geräte (bis auf die mobilen Geräte) kommunizieren dabei über angeschlossene Mac mini's. Für die Koordination bietet sich eine Client/Server Architektur an, bei der ein Server die Übersicht über alle Services im Haus hat und die Nachrichten zwischen diesen routen kann. Bei einem Peer-to-Peer System wäre die Koordination bei einer großen Anzahl von Clients aufwändiger. Es ist zudem schwieriger zu erkennen ob ein Teilnehmer noch in Reichweite ist, was vor allem die Arbeit mit mobilen Geräten erschwert.

Die Idee im iFlat ist, die Kommunikation über eine zentrale Kommunikationskomponente laufen zu lassen. Als zentrale Kommunikationskomponente eignet sich die Middleware iROS, welche von Dennis Hollatz in [Hol07] bereits verifiziert wurde und bereits in dem Masterprojekt „smart:shelf“ [HAW08] von den Studenten Stefan Meißner, Jaroslaw Urich und Dennis Hollatz eingesetzt wurde.

iROS ist ein Metabetriebssystem auf Java-Basis das an der Stanford University entwickelt wurde. Das iROS dient als Plattform zur Kommunikation zwischen verschiedenen Devices und ist aus mehreren Komponenten aufgebaut. Eine der wichtigsten Komponenten ist der „Event Heap“. Dieser dient als die zentrale Kommunikationsstelle für die Devices.

Dienste melden sich beim Event Heap für bestimmte Ereignistypen an. Beispielsweise meldet sich ein Bildschirm für Ereignisse an die ein Display erfordern. Der Event Heap funktioniert dabei so, dass Dienste (also unsere Devices) Events in den Event Heap „pushen“ und von dort an einen anderen Service weitergeleitet werden. Der Event Heap arbeitet dabei nach dem FIFO Prinzip.

4.1 Event Manager

Der Event Heap des iROS wurde im Masterprojekt „smart:shelf“ als Event Manager implementiert (siehe Abbildung 4.1).

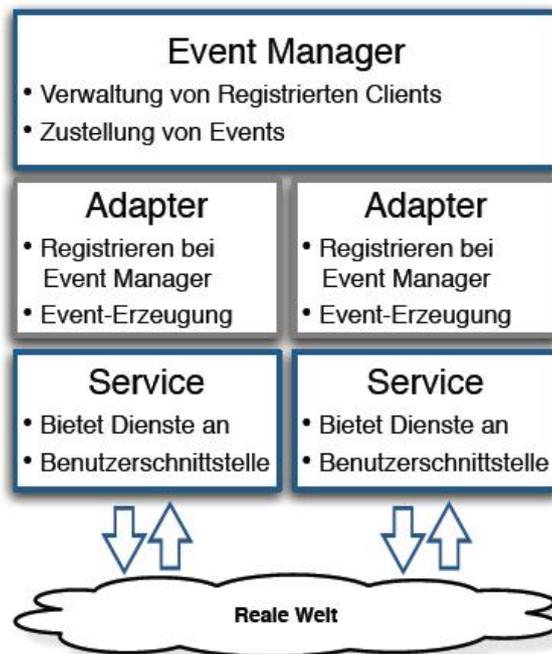


Abbildung 4.1: Konzept der Kommunikations-Infrastruktur im iFlat (Quelle: [Hol08])

Der Event Manager verwaltet die Services die sich bei ihm anmelden (z.B. der Bildschirm, die Stereo-Anlage, etc.) und regelt die korrekte Zustellung der Events.

Jeder Service besitzt einen Adapter der für die Registrierung am Event Manager und für die Event Erzeugung zuständig ist.

Die Funktionsweise wird nachfolgend anhand eines Beispiels verdeutlicht.

«Der Service „Bildschirm“ meldet sich über seinen Adapter am Event Manager an, z.B. mit den Parametern Gerätetyp „Display“ und max. Auflösung „1280x1024“.

Ein Mobiltelefon meldet sich auch über seinen Adapter am Event Manager an.

Das Mobiltelefon empfängt nun ein Bild als mail-Anhang, der Adapter verpackt das Bild als Event und übergibt das Event an den Event-Manager.

Der Event Manager weiß nun, dass es sich um ein Bild der Grösse 800x600 handelt und dafür ein möglichst grosses Display mit der Mindestauflösung von 800x600 benötigt.

Er findet den angebotenen Service vom „Bildschirm“, leitet das Event an ihn weiter und das Bild wird auf dem Bildschirm angezeigt.»

Der Event Manager sowie die Adapter wurden im „smart:shelf“ Projekt in Java realisiert.

5 Ausblick

Die Integration des iPhone in das iFlat wird mit der in Kapitel 4 beschriebenen Architektur nicht reibungslos ablaufen, da der dort beschriebene Event-Manager Java-basiert arbeitet. Die iPhone SDK dagegen ist in Objective-C geschrieben. Hier ist die Idee, Nachrichten über das einheitliche XML-Format auszutauschen, was auch von Dennis Hollatz in [Hol08] angesprochen wurde. Hier ist eine Absprache mit Dennis Hollatz notwendig, da diese Idee eventuell nicht von ihm weiter verfolgt wird.

Der Einsatz von mobilen Geräten im intelligenten Haus bringt zusätzlich einige Sicherheitsaspekte mit sich, die beachtet werden müssen. Der Bewohner sollte Zugriff auf alle seine Geräte haben während für Gäste kein, oder ein eingeschränkter Zugriff auf die Geräte möglich sein soll, z.B. um seine letzten Urlaubsfotos auf einem Bildschirm des Bewohners zu zeigen. In weiteren Untersuchungen sollten sich also Gedanken über die Möglichkeiten einer Authentifizierung am Event Manager gemacht werden. Anbieten würde sich hier ein Kerberos Authentifizierungsdienst der für den Einsatz mit dem iPhone noch verifiziert werden muss. Die Unterstützung von Kerberos scheint erst in späteren Versionen des iPhone zur Verfügung zu stehen.

Eine zusätzliche Idee wäre die Registrierung am Event Manager über das Internet, um z.B. das Haus von jedem Ort aus fernzusteuern. So könnte der Bewohner z.B. auf dem Heimweg schon mal die Klimaanlage aktivieren.

Weiterhin bleibt zu beobachten welche Erweiterungen noch von anderen Masterstudenten an Event Manager und Adaptern vorgenommen werden. Ansätze hierzu finden sich z.B. in [Hol08]. Dort bringt Dennis Hollatz für die Erweiterung der Adapter die Idee eines Multiagentensystems ins Spiel. Agenten könnten in diesem Zusammenhang entsprechend der Umgebung Events unterdrücken oder erzeugen und könnten ihren Zustand anderen Agenten mitteilen. Damit wäre das Verhalten nicht explizit vorgegeben sondern bestimmt sich aus dem Verhalten der einzelnen Agenten.

Außerdem spricht er dort die Nachrichtenverschlüsselung an. Dies sollte weiter im Auge behalten werden, da eine sichere Kommunikation im intelligenten Haus, vor allem mit privaten oder vertraulichen Daten, sehr wichtig sein kann.

6 Zusammenfassung

Die vorliegende Arbeit ist eine erste Recherche zum Einsatz von mobilen Geräten im intelligenten Haus. Dabei wurden die Themen Personal Information Environments, das iPhone von Apple und das iFlat der HAW angesprochen. Damit wurden die Grundlagen geklärt und der derzeitige Entwicklungsstand des iFlat grob aufgezeigt, um somit einen Überblick über das Themengebiet zu bekommen und den Einstieg zur Integration des iPhone's im iFlat zu erleichtern.

Als Kommunikationsplattform wurden die Vorzüge einer Client/Server Architektur für den Einsatz in einem intelligenten Haus aufgezeigt, welche bereits im iFlat mit der Middleware iROS der Stanford University eingesetzt wurde.

Für den Einsatz des iPhone im iFlat wurden Ideen gesammelt und Probleme aufgezeigt. Da die iPhone SDK noch ziemlich neu ist, dient die vorliegende Arbeit eher für einen ersten Einblick in die Entwicklungsmöglichkeiten mit dem iPhone, statt konkrete Implementierungsansätze zu geben.

In einem Ausblick wurden zudem Hinweise auf folgende Untersuchungen für das iPhone im iFlat gegeben, wobei vor allem die Punkte Nachrichtenaustausch über das einheitliche XML-Format, Authentifizierung am Server sowie die Nachrichtenverschlüsselung wichtig sind.

Literaturverzeichnis

- [APP08a] *Apple Inc.: iPhone Dev Center*. Webseite. Stand: 23. Juli 2008. –
<http://developer.apple.com/iphone>
- [APP08b] *Apple Inc.: WWDC 2008 Keynote*. Präsentation. Stand: 23. Juli 2008. –
<http://events.apple.com.edgesuite.net/0806wdt546x/event/index.html>
- [APP08c] *Apple Inc.: Apple March 6 Event*. Präsentation. Stand: 24. Mai 2008. –
<http://www.apple.com/quicktime/qtv/iphoneroadmap/>
- [CEE08] *Hochschule Luzern: CEESAR Center of Excellence for Embedded Systems Applied Research*. Webseite. Stand: 24. Juli 2008. –
<http://www.ceesar.ch>
- [Dre07] DREYER, Markus: *Your Home in Your Hand: Compare current work*. Seminararbeit. 4. Dezember 2007. –
<http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08-aw/dreyer/bericht.pdf>
- [FRA08] *Fraunhofer IAO: Smart Living*. Webseite. Stand: 24. Juli 2008. –
http://www.hci.iao.fraunhofer.de/de/arbeitsgebiete/smart_living
- [GAT08] *Georgia Institute of Technology: Personal Information Environments @ Georgia Tech*. Webseite. Stand: 22. Juli 2008. –
<http://www.cc.gatech.edu/projects/PIE/>
- [HAW08] HAW: *Berichte: Master Informatik HAW Hamburg Wintersemester 2007/08*. Projektberichte. Stand: 24. Juli 2008. –
<http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08-proj/berichte.html>
- [Hol07] HOLLATZ, Dennis: *Konzepte für interaktive Räume*. Seminararbeit. 31. Juli 2007. –
<http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2007/hollatz/bericht.pdf>
- [Hol08] HOLLATZ, Dennis: *Managing Information - Personal Information Environments auf Basis von iROS*. Seminararbeit. 3. März 2008. –
<http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08/hollatz/bericht.pdf>

- [IBM08] IBM Research: *Personal Information Environments, Almaden Research Center*. Webseite. Stand: 22. Juli 2008. –
<http://www.almaden.ibm.com/cs/projects/pie>
- [STA08] Stanford University: *Stanford Interactive Workspaces Project Overview*. Webseite. Stand: 24. Juli 2008. –
<http://iwork.stanford.edu>
- [Uri08] URICH, Jaroslaw: *smart:shelf*. Seminararbeit. 27. Februar 2008. –
<http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08-proj/urich/report.pdf>
- [Vol07] VOLLMER, Sven: *Dienstsuche im IntelliHome*. Seminararbeit. 14. Dezember 2007. –
<http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08/vollmer/bericht.pdf>